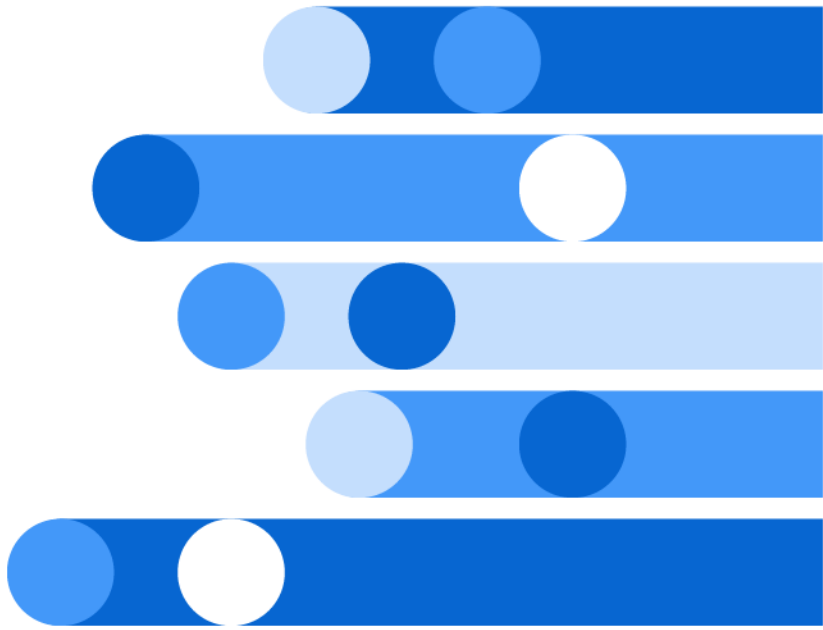
SAS® ODS Graphics: Procedures Guide

2024.04 - 2026.08*



* This document might apply to additional versions of the software. Open this document in SAS Help Center and click on the version in the banner to see all available versions.

SAS® Documentation
August 18, 2026

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2024. *SAS® ODS Graphics: Procedures Guide*. Cary, NC: SAS Institute Inc.

SAS® ODS Graphics: Procedures Guide

Copyright © 2024, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

August 2026

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

v_003-P1:grstatproc

Contents

Syntax Conventions for the SAS Language . . . *ix*
Accessibility . . . *xv*

PART 1 Getting Started 1

Chapter 1 / Get Started Using the ODS Graphics Procedures . . . **3**
About the SAS ODS Graphics Procedures . . . 3
Summary of the Procedures . . . 5
Example: Create Your First Graph . . . 8
Common Tasks Associated with Using the Procedures . . . 10
Where to Go from Here . . . 13

Chapter 2 / Gallery of Plots and Charts . . . **15**
Alphabetical List of Plots and Charts . . . 16
Basic Plots and Charts . . . 16
Fit and Confidence Plots . . . 41
Distribution Plots . . . 47
Categorization Plots and Charts . . . 52

PART 2 Overview: Using the Procedures 67

Chapter 3 / Concepts for Creating Graphs . . . **69**
Creating Single-Cell Graphs . . . 69
Creating Multi-Cell Classification Panels . . . 71
Creating Paneled Scatter Plots . . . 72
Rendering Graphs from GTL Templates . . . 73
Producing Maps Using the SGMAP Procedure . . . 74
Plotting a Cloud Analytic Services (CAS) In-Memory Table . . . 75
About ODS Destinations and Styles . . . 79

Chapter 4 / Elements of a Program . . . **81**
A Typical Program . . . 81
About the PROC Step . . . 82
About SAS Statements . . . 85
About the ODS GRAPHICS Statement Options . . . 85
About Using an Annotation Data Set . . . 86
About Using an Attribute Map Data Set . . . 86

Chapter 5 / About the Examples, Sample Data, and Additional Resources **87**
About the Examples in This Book 87
About the SASHELP Data Library 88
Examples and Resources on the Web 88

PART 3 SAS Statements Used with the Procedures 91

Chapter 6 / Global Statements **93**
About the Global Statements 93
Dictionary 94

Chapter 7 / Common Procedure Statements **141**
About the Common Procedure Statements 142
ATTRIB Statement 142
BY Statement 143
FORMAT Statement 146
LABEL Statement 150
WHERE Statement 151

PART 4 The Procedures 153

Chapter 8 / SGPANEL Procedure **155**
Overview: SGPANEL Procedure 156
Concepts: SGPANEL Procedure 158
Syntax: SGPANEL Procedure 160
Examples: SGPANEL Procedure 829

Chapter 9 / SGPIE Procedure: Preproduction **839**
Overview: SGPIE Procedure: Preproduction 840
Syntax: SGPIE Procedure: Preproduction 841
Examples: SGPIE Procedure: Preproduction 877

Chapter 10 / SGPLOT Procedure **883**
Overview: SGPLOT Procedure 884
Concepts: SGPLOT Procedure 886
Syntax: SGPLOT Procedure 888
Examples: SGPLOT Procedure 1656

Chapter 11 / SGRENDER Procedure **1677**
Overview: SGRENDER Procedure 1677
Syntax: SGRENDER Procedure 1678
Examples: SGRENDER Procedure 1691

Chapter 12 / SGSCATTER Procedure **1699**
Overview: SGSCATTER Procedure 1700
Concepts: SGSCATTER Procedure 1702
Syntax: SGSCATTER Procedure 1708

Examples: SGSCATTER Procedure 1769

Chapter 13 / Common Concepts 1777
Overview of the Common Concepts 1778
Plot Type Compatibility 1779
Plot Axes 1784
Legends 1790
Automatic Differentiation of Visual Attributes 1797
Fit Policies for Axis Tick Values, Curve Labels, and Data Labels 1802
Marker Fills and Outlines 1805
Color-Naming Schemes 1808
Fonts 1819

Chapter 14 / Commonly Used Attribute Options 1827
General Syntax for Attribute Options 1827
Line Attributes and Patterns 1828
Fill Attributes 1832
Marker Attributes and Symbols 1833
Text Attributes 1835
Units of Measurement 1836

PART 5 Controlling the Procedure Output 1837

Chapter 15 / Controlling the Appearance of Your Graphs 1839
Overview 1840
Understanding Styles 1841
Specifying Styles 1847
Using Plot Options to Control Graph Appearance 1847
Using Data Skins 1851
Subpixel Rendering 1857
Output for Grouped versus Non-Grouped Data 1858
Modifying Style Templates 1864
Using Fill Patterns to Distinguish Categories of Data 1865
Style Elements for Use with ODS Graphics 1871

Chapter 16 / Managing Your Graphics with ODS 1883
Introduction to Managing Your Graphics with ODS 1884
Specifying ODS Destinations 1884
ODS Destination Statement Options Affecting ODS Graphics 1885
Using the ODS GRAPHICS Statement 1886
Controlling the Size of Your Graph 1888
Controlling Image Resolution 1891
Adding Data Tips and Other Features 1891
Controlling the Image Name and Image Format 1892
Resetting ODS GRAPHICS Options 1893

PART 6 SG Attribute Maps 1895

Chapter 17 / Using Attribute Maps to Control Visual Attributes **1897**
Overview of Attribute Maps 1897
About Discrete Attribute Maps 1898
About Range Attribute Maps 1898

Chapter 18 / Using Discrete Attribute Maps **1901**
How Discrete Attribute Maps Work 1901
Discrete Attribute Map Data Sets 1902
Modify the Procedure to Use the Discrete Attribute Map Data Set 1911
Example: Create a Plot That Uses a Single Discrete Attribute Map 1912
Example: Combine Multiple Discrete Attribute Maps in a Graph 1914
Example: Create a Panel That Uses an Attribute Map 1916
Example: Use a Discrete Attribute Map with an Axis Table 1918
Example: Use a Discrete Attribute Map with the SGPIE Procedure 1919

Chapter 19 / Using Range Attribute Maps **1921**
Main Tasks for Range Attribute Mapping 1921
Range Attribute Map Data Sets 1922
Modify the Procedure to Use the Range Attribute Map Data Set 1928
Example: Create a Panel That Uses COLORMODELn Variables 1930

PART 7 SG Annotation 1933

Chapter 20 / Annotating ODS Graphics **1935**
Overview of SG Annotation 1935
SG Annotation Data Sets 1936
Specifying an SG Annotation Data Set in an SG Procedure Statement 1940
Controlling the Drawing Space 1941

Chapter 21 / SG Annotation Function Dictionary **1945**
Dictionary 1945
Examples 2000

Chapter 22 / SG Annotation Macro Dictionary **2007**
About the SG Annotation Macros 2007
Using the SG Annotation Macros 2008
Dictionary 2011
Examples 2070

PART 8 Appendix 2077

Appendix 1 / Units of Measurement **2079**
Units of Measurement 2079

Appendix 2 / Reserved Keywords and Unicode Values **2081**
Reserved Keywords and Unicode Values 2081

Appendix 3 / ODS Graphics Software **2085**
Overview of ODS Graphics Software 2085

Syntax Conventions for the SAS Language

Overview of Syntax Conventions for the SAS Language

SAS uses standard conventions in the documentation of syntax for SAS language elements. These conventions enable you to easily identify the components of SAS syntax. The conventions can be divided into these parts:

- syntax components
- style conventions
- special characters
- references to SAS libraries and external files

Syntax Components

The components of the syntax for most language elements include a keyword and arguments. For some language elements, only a keyword is necessary. For other language elements, the keyword is followed by an equal sign (=). The syntax for arguments has multiple forms in order to demonstrate the syntax of multiple arguments, with and without punctuation.

keyword
specifies the name of the SAS language element that you use when you write your program. Keyword is a literal that is usually the first word in the syntax. In a CALL routine, the first two words are keywords.

In these examples of SAS syntax, the keywords are bold:

CHAR (*string, position*)

CALL RANBIN (*seed, n, p, x*);

ALTER (*alter-password*)

BEST *w*.

REMOVE <*data-set-name*>

In this example, the first two words of the CALL routine are the keywords:

CALL RANBIN(*seed*, *n*, *p*, *x*)

The syntax of some SAS statements consists of a single keyword without arguments:

DO;
... *SAS code* ...
END;

Some system options require that one of two keyword values be specified:

DUPLEX | **NODUPLEX**

Some procedure statements have multiple keywords throughout the statement syntax:

CREATE <UNIQUE> **INDEX** *index-name* **ON** *table-name* (*column-1* <, *column-2*, ...>)

argument
specifies a numeric or character constant, variable, or expression. Arguments follow the keyword or an equal sign after the keyword. The arguments are used by SAS to process the language element. Arguments can be required or optional. In the syntax, optional arguments are enclosed in angle brackets (< >).

In this example, *string* and *position* follow the keyword CHAR. These arguments are required arguments for the CHAR function:

CHAR (*string, position*)

Each argument has a value. In this example of SAS code, the argument *string* has a value of 'summer', and the argument *position* has a value of 4:

```
x=char('summer', 4);
```

In this example, *string* and *substring* are required arguments, whereas *modifiers* and *startpos* are optional.

FIND(*string, substring* <, *modifiers*> <, *startpos*>

argument(s)
specifies that one argument is required and that multiple arguments are allowed. Separate arguments with a space. Punctuation, such as a comma (,) is not required between arguments.

The MISSING statement is an example of this form of multiple arguments:

MISSING *character(s)*;

<LITERAL_ARGUMENT> *argument-1* <<LITERAL_ARGUMENT> *argument-2* ... >
specifies that one argument is required and that a literal argument can be associated with the argument. You can specify multiple literals and argument

pairs. No punctuation is required between the literal and argument pairs. The ellipsis (...) indicates that additional literals and arguments are allowed.

The BY statement is an example of this argument:

BY <DESCENDING> *variable-1* <<DESCENDING> *variable-2* ...>;

argument-1 <*options*> <*argument-2* <*options*> ...>
specifies that one argument is required and that one or more options can be associated with the argument. You can specify multiple arguments and associated options. No punctuation is required between the argument and the option. The ellipsis (...) indicates that additional arguments with an associated option are allowed.

The FORMAT procedure PICTURE statement is an example of this form of multiple arguments:

PICTURE name <(*format-options*)>
<*value-range-set-1* <(*picture-1-options*)>
<*value-range-set-2* <(*picture-2-options*)> ...>>;

argument-1=value-1 <*argument-2=value-2* ...>
specifies that the argument must be assigned a value and that you can specify multiple arguments. The ellipsis (...) indicates that additional arguments are allowed. No punctuation is required between arguments.

The LABEL statement is an example of this form of multiple arguments:

LABEL *variable-1=label-1* <*variable-2=label-2* ...>;

argument-1 <, *argument-2*, ...>
specifies that one argument is required and that you can specify multiple arguments that are separated by a comma or other punctuation. The ellipsis (...) indicates a continuation of the arguments, separated by a comma. Both forms are used in the SAS documentation.

Here are examples of this form of multiple arguments:

AUTHPROVIDERDOMAIN (*provider-1:domain-1* <, *provider-2:domain-2*, ...>

INTO :*macro-variable-specification-1* <, :*macro-variable-specification-2*, ...>

Note: In most cases, example code in SAS documentation is written in lowercase with a monospace font. You can use uppercase, lowercase, or mixed case in the code that you write.

Style Conventions

The style conventions that are used in documenting SAS syntax include uppercase bold, uppercase, and italic:

UPPERCASE BOLD
identifies SAS keywords such as the names of functions or statements. In this example, the keyword ERROR is written in uppercase bold:

ERROR <*message*>;

UPPERCASE
identifies arguments that are literals.

In this example of the CMPMODEL= system option, the literals include BOTH, CATALOG, and XML:

CMPMODEL=BOTH | CATALOG | XML |

italic
identifies arguments or values that you supply. Items in italic represent user-supplied values that are either one of the following:

- nonliteral arguments. In this example of the LINK statement, the argument *label* is a user-supplied value and therefore appears in italic:

  LINK *label*;

- nonliteral values that are assigned to an argument.

  In this example of the FORMAT statement, the argument DEFAULT is assigned the variable *default-format*:

  FORMAT *variable(s)* <*format* > <DEFAULT = *default-format*>;

Special Characters

The syntax of SAS language elements can contain the following special characters:

=
an equal sign identifies a value for a literal in some language elements such as system options.

In this example of the MAPS system option, the equal sign sets the value of MAPS:

MAPS=*location-of-maps*

< >
angle brackets identify optional arguments. A required argument is not enclosed in angle brackets.

In this example of the CAT function, at least one item is required:

CAT (*item-1* <, *item-2*, ...>)

|
a vertical bar indicates that you can choose one value from a group of values. Values that are separated by the vertical bar are mutually exclusive.

In this example of the CMPMODEL= system option, you can choose only one of the arguments:

CMPMODEL=BOTH | CATALOG | XML

...
an ellipsis indicates that the argument can be repeated. If an argument and the ellipsis are enclosed in angle brackets, then the argument is optional. The repeated argument must contain punctuation if it appears before or after the argument.

In this example of the CAT function, multiple *item* arguments are allowed, and they must be separated by a comma:

CAT (*item-1* <, *item-2*, ...>)

'*value*' or "*value*"
indicates that an argument that is enclosed in single or double quotation marks must have a value that is also enclosed in single or double quotation marks.

In this example of the FOOTNOTE statement, the argument *text* is enclosed in quotation marks:

FOOTNOTE <*n*> <*ods-format-options* '*text*' | "*text*">;

;
a semicolon indicates the end of a statement or CALL routine.

In this example, each statement ends with a semicolon:

```
data namegame;
   length color name $8;
   color = 'black';
   name = 'jack';
   game = trim(color) || name;
run;
```

References to SAS Libraries and External Files

Many SAS statements and other language elements refer to SAS libraries and external files. You can choose whether to make the reference through a logical name (a libref or fileref) or use the physical filename enclosed in quotation marks.

If you use a logical name, you typically have a choice of using a SAS statement (LIBNAME or FILENAME) or the operating environment's control language to make the reference. Several methods of referring to SAS libraries and external files are available, and some of these methods depend on your operating environment.

In the examples that use external files, SAS documentation uses the italicized phrase *file-specification*. In the examples that use SAS libraries, SAS documentation uses the italicized phrase *SAS-library* enclosed in quotation marks:

```
infile file-specification obs = 100;
libname libref 'SAS-library';
```

Accessibility

Accessibility Features of ODS Graphics Procedures

Overview

The ODS Graphics Procedures have been tested against the accessibility standards for electronic information technology that were adopted by the U.S. Government under Section 508 of the U.S. Rehabilitation Act of 1973 (2008 draft proposal initiative update). The procedures are also tested against Web Content Accessibility Guidelines (WCAG) 2.0, part of the Web Accessibility Initiative (WAI) of the Worldwide Web Consortium (W3C). For detailed information about the accessibility of this product, send email to accessibility@sas.com, or call SAS Technical Support.

Documentation Format

Please contact accessibility@sas.com if you need this document in an alternative digital format.

Creation of Accessible Graphs

For information about generating accessible graphs using the procedures in conjunction with the Output Delivery System (ODS) and ODS Graphics software, see *Creating Accessible SAS Viya Platform Output Using ODS and ODS Graphics*.

PART 1

Getting Started

Chapter 1
Get Started Using the ODS Graphics Procedures 3

Chapter 2
Gallery of Plots and Charts 15

1

Get Started Using the ODS Graphics Procedures

About the SAS ODS Graphics Procedures . . . 3

Summary of the Procedures . . . 5
- Procedures That Create Graphs and Panels . . . 5
- Procedure That Creates Geographical Maps . . . 7
- Procedure That Renders Graphs That Are Defined Externally . . . 7
- Facilities Used to Customize and Annotate Procedure Output . . . 7

Example: Create Your First Graph . . . 8
- About This Example . . . 8
- Code for the Example . . . 8
- Output . . . 10

Common Tasks Associated with Using the Procedures . . . 10
- Setting the Output Environment . . . 10
- Creating Graphs . . . 11
- Enhancing Graphs . . . 12

Where to Go from Here . . . 13

About the SAS ODS Graphics Procedures

The ODS Graphics procedures, sometimes called Statistical Graphics procedures, use ODS Graphics functionality to produce plots for exploratory data analysis and for customized statistical displays. The procedures support statistical analysis and create simple or complex graphical views of your data. Though the procedures were initially designed to facilitate the production of standard statistical graphs, the procedures are also well suited for the production of non-statistical or business graphs.

The procedures provide a simple, high-level syntax that enables you to produce sophisticated graphs by using a wide array of plot types and layouts. You can create scatter plots, histograms, bar charts, box plots, classification panels, scatter plot matrices, and many other types of statistical and business graphs. Your graphs can have titles, footnotes, legends, and other graphics elements.

The ODS Graphics procedures create graphs that are based on the SAS Graph Template Language (GTL). However, you do not need to know the details of templates and the GTL in order to use the ODS Graphics procedures. With very little coding effort, you can use the procedures to create the most commonly used graphs that are supported by the GTL.

The following table lists the ODS Graphics procedures that are included with Base SAS.

Table 1.1 *ODS Graphics Procedures That Are Included With Base SAS*

Procedure	Description
SGPLOT	Creates single-cell plots with a variety of plot and chart types and overlays. The SGPLOT procedure is described in this document.
SGPIE	Creates pie and donut charts. The SGPIE procedure is described in this document.
SGMAP	Creates geographical maps. The SGMAP procedure is described in "SGMAP Procedure" in *SAS/GRAPH and Base SAS: Mapping Reference*.
SGPANEL	Creates classification panels for one or more classification variables. The SGPANEL procedure is described in this document.
SGSCATTER	Creates scatter plot panels and scatter plot matrices with optional fits and ellipses. The SGSCATTER procedure is described in this document.
SGRENDER	Renders graphs from graph templates, which are written in the Graph Template Language (GTL). The SGRENDER procedure is described in this document. GTL is described in *SAS Graph Template Language: User's Guide* and *SAS Graph Template Language: Reference*.

For more information about these procedures, see "Summary of the Procedures" on page 5.

An ODS destination must be open to create output from these procedures. By default, the ODS HTML5 destination is open. The HTML output is displayed automatically when you execute the procedure. In addition, you can use the ODS GRAPHICS statement options along with the ODS destination statements to

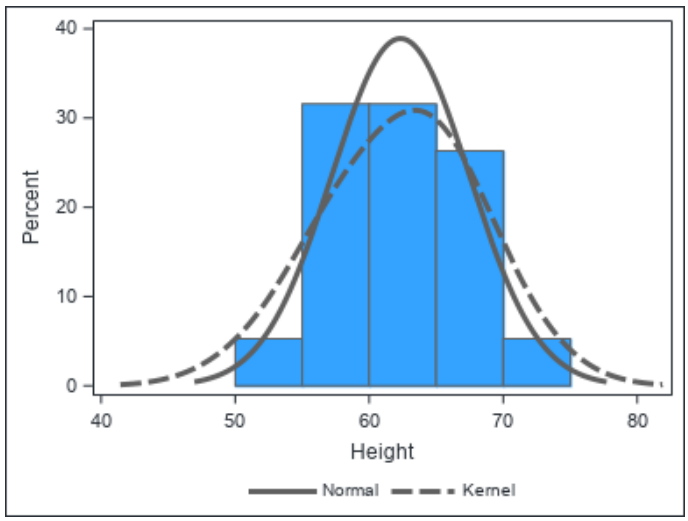
control many aspects of your graphics. For more information, see Chapter 16, "Managing Your Graphics with ODS," on page 1883.

Summary of the Procedures

Procedures That Create Graphs and Panels

Here are the ODS Graphics procedures that create graphs:

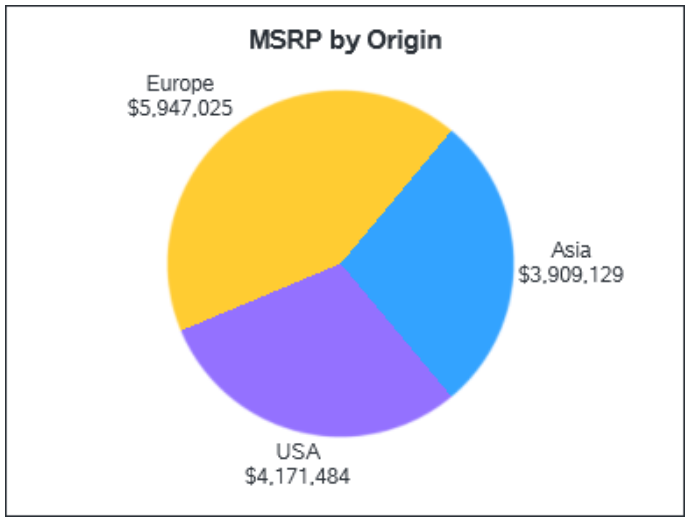
SGPLOT
: creates single-cell plots with a variety of plot and chart types and overlays.



See "Creating Single-Cell Graphs" on page 69.

SGPIE
: The SGPIE procedure includes the ability to create pie and donut charts. This procedure is considered preproduction.

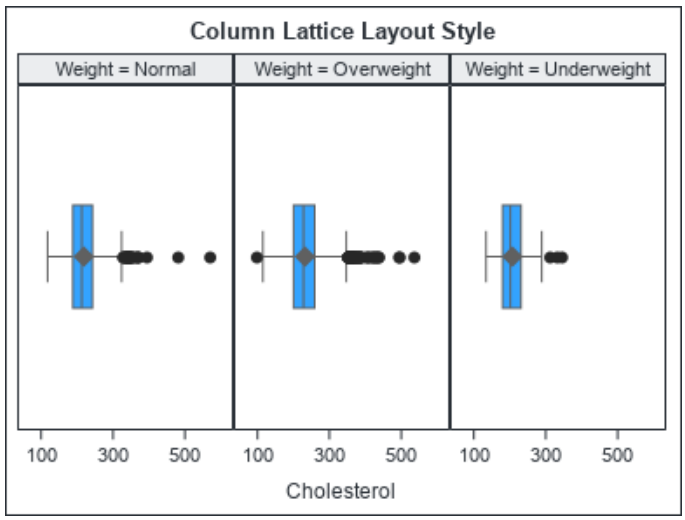
Here is a pie chart.



See Overview: SGPIE Procedure on page 840.

SGPANEL
creates classification panels for one or more classification variables. The layout contains paneled cells determined by the classification variables. Each panel can contain either a simple plot or multiple, overlaid plots. All panels are the same size. Changes that you make to plot attributes, such as line patterns and fill colors, apply to all panels in the graph.

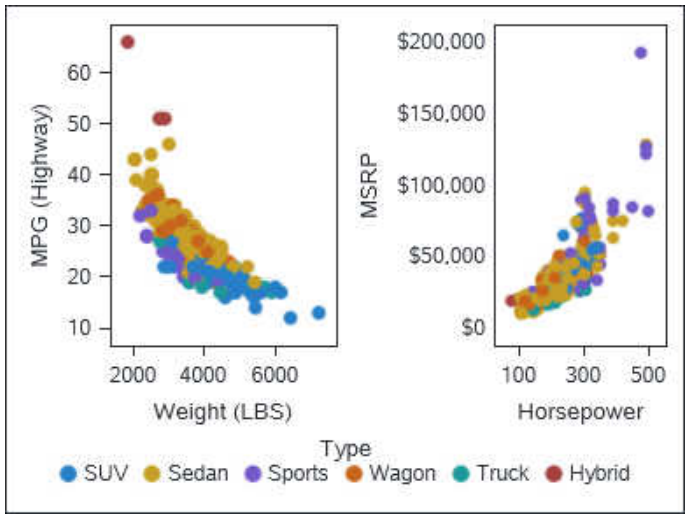
Here is a graph with three panels.



Note: The SGPLOT and SGPANEL procedures largely support the same types of plots and charts. For this reason, the two procedures have an almost identical syntax. The main distinction between the two procedures is that the SGPANEL procedure produces a panel of graphs, one for each level of a classification variable.

See "Creating Multi-Cell Classification Panels" on page 71.

SGSCATTER
creates scatter plot panels and scatter plot matrices with optional fits and ellipses. The following example shows a paneled graph of scatter plots



See "Creating Paneled Scatter Plots" on page 72.

Procedure That Creates Geographical Maps

The SGMAP procedure is included with Base SAS. This procedure provides concise syntax for creating geographical maps. You can also overlay plots such as scatter plots, bubble plots, and text plots onto the map.

Note: This document does not describe the SGMAP procedure. For complete documentation about the SGMAP procedure, see "SGMAP Procedure" in *SAS/GRAPH and Base SAS: Mapping Reference*.

Procedure That Renders Graphs That Are Defined Externally

The SGRENDER procedure renders graphs from graph templates, which are written in the Graph Template Language (GTL).

See "Rendering Graphs from GTL Templates" on page 73.

Facilities Used to Customize and Annotate Procedure Output

The procedures have two facilities that enable you to modify graph output:

- The SG annotation feature enables you to add text, shapes, images, and other annotations to graph output.
- SG attribute maps enable you to control the visual attributes that are applied to specific data values in your graphs. Attribute maps enable you to ensure that particular visual attributes are applied based on the value of the data rather than the position of the data in the data set.

Example: Create Your First Graph

About This Example

This example produces a simple bar chart that shows the mean highway miles per gallon for various types of vehicles. A bar chart summarizes the values of a category variable. In this example, the bars in the chart are grouped by country of origin. The example takes you through the basic steps that are needed to create a chart or plot using the SGPLOT procedure.

In this example, you perform the following tasks:

- use the ODS GRAPHICS statement to specify the graph size.
- use the TITLE and FOOTNOTE statements to add a title and footnote to the graph.
- use the SGPLOT procedure to identify the data and create the bar chart. You specify the mean statistic to use for the chart's response variable. You also specify the variable used to group the data.

  Finally, you specify a custom label for the Y axis.

Using what you learn from this example, you can begin exploring how to create other charts and plots with the procedures.

Code for the Example

```
ods graphics / width=4.5in;                                    /* 1
*/

title "Mean Highway MPG By Vehicle Type and Origin";           /* 2
*/
footnote "Data: SASHELP.CARS";
```

```
proc sgplot data=sashelp.cars (where=(type ne 'Hybrid'));        /* 3
*/
   vbar type / response=mpg_highway                              /* 4
*/
          stat=mean
          group=origin
          groupdisplay=cluster;
   yaxis label="Mean Highway MPG";                               /* 5
*/
run;

title;                                                           /* 6
*/
footnote;
ods graphics /
reset=width;                                                  /* 7 */
```

1 The WIDTH option of the ODS GRAPHICS statement specifies the graph width.

Note: The procedure uses a standard aspect ratio for the graph. If the width is specified but not the height, then the procedure determines the height automatically to maintain the default aspect of the graph.

2 The TITLE and FOOTNOTE statements add a title and footnote to the graph.

3 The PROC SGPLOT statement begins the procedure block that defines and creates the graph. The statement also identifies the plot data. In this example, the Sashelp.Cars data set is used. The data is subset to omit hybrid type vehicles.

4 The VBAR statement defines a vertical bar chart. The forward slash (/) and everything that follows it within the VBAR statement are optional. You specify the mean statistic to use for the chart's response variable. You also specify the variable used to group the data.

Here are the details about the bar chart options:

- The RESPONSE=*mpg_highway* option specifies that the MPG_Highway variable is used for the response axis (Y axis). If you do not specify a response variable, the procedure displays the frequency count of the category variable.
- The STAT=MEAN option specifies the calculated statistic for the response axis.
- The GROUP=ORIGIN option specifies the variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes. In this example, the data is grouped by the country of origin.
- The GROUPDISPLAY=CLUSTER option specifies how to display the grouped bars. By default, groups are stacked in bars without any clustering.

5 The LABEL= option in the YAXIS statement enables you to specify a custom label for the Y axis. The default Y-axis label for this example is MPG (Highway).

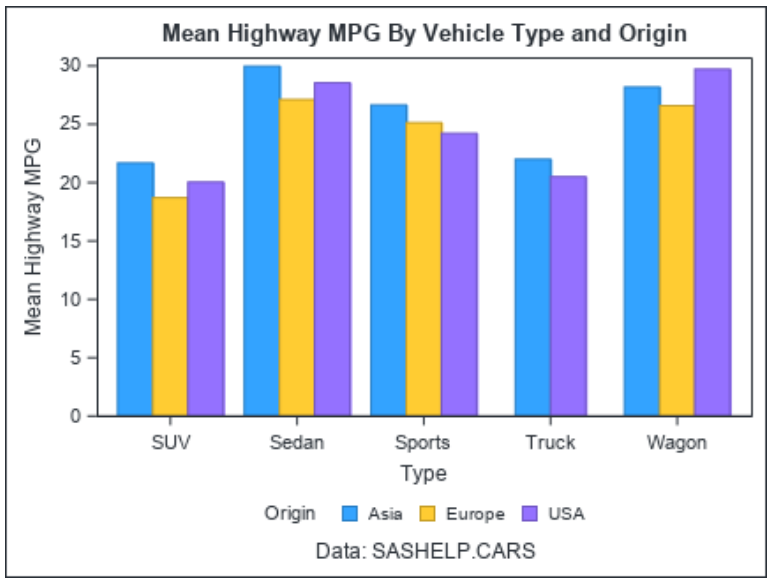
6 The TITLE and FOOTNOTE statements cancel the current title and footnote.

7 The ODS GRAPHICS statement resets the WIDTH option to its default.

Note: You can instead specify the following: `ods graphics / reset;`

This statement resets all ODS Graphics options that have been set in the session to their defaults.

Output



Common Tasks Associated with Using the Procedures

Setting the Output Environment

Table 1.2 *Tasks for Setting the Output Environment*

Task	Description
Specify the style	ODS styles determine the overall appearance of your output. A style controls the default colors, fonts, line styles, fill colors, and other presentation aspects. By default, ODS applies a style to all output.

Task	Description
	For more information and sample styles, see "Understanding Styles" on page 1841.
Learn about image size, resolution, and more	The ODS GRAPHICS statement enables ODS Graphics output and controls many aspects of your graphics. **Note:** ODS GRAPHICS is ON by default. See Chapter 16, "Managing Your Graphics with ODS," on page 1883. See also "ODS GRAPHICS Statement" on page 102.
Use ODS destinations to specify the output document format	See Chapter 16, "Managing Your Graphics with ODS," on page 1883.

Creating Graphs

Table 1.3 *Tasks for Creating Graphs*

Task	Description
Learn about the types of graphs that you can create	The plot gallery shows the many plots and charts that you can create with the procedures. See Chapter 2, "Gallery of Plots and Charts," on page 15.
Learn about the graph layouts	You can create single-cell graphs, classification panels, and scatter plot matrices. See "Summary of the Procedures" on page 5.
Overlay plots one on top of the other	You can add multiple plots and reference lines to a graph. Each plot or line is specified in its own plot statement. All overlaid plots must be compatible. See "Plot Type Compatibility" on page 1779.
Create accessible graphs	You can produce a wide range of plot types that can be read and interpreted by the SAS Graphics Accelerator. See *Creating Accessible SAS Viya Platform Output Using ODS and ODS Graphics*. See also *SAS Graphics Accelerator: Supported Types of Data Visualizations* and the SAS Graphics Accelerator product page on the SAS support site.

Task	Description
Control graph appearance	The procedures contain many options for controlling the appearance and output of graphics. For example, you can customize labels, markers, fills, lines, and so on. You can also manage the appearance of group data. See Chapter 15, "Controlling the Appearance of Your Graphs," on page 1839.
Add titles and footnotes	You can specify titles and footnotes for your graphs by using the TITLE and FOOTNOTE global statements. See "Global Statements" on page 93.
Create legends	The procedures create a legend automatically based on the plot statements and options that you specify. You can create discrete legends and color gradient legends manually. You can also customize the appearance of your legends using the legend options. For more information, see "Legends" on page 1790.
Control the axes	You can change the appearance of the axes, customize axis labels, add grid lines, and more. You can also change an axis type and customize the range of values that are displayed on the axis. See "Plot Axes" on page 1784.

Enhancing Graphs

Table 1.4 *Tasks for Enhancing Graphs*

Task	Description
Add data tips and drill-down links	You can add data tips and URL drill-down functionality to your graphs. See "Adding Data Tips and Other Features" on page 1891.
Use persistent attributes (attribute mapping)	You can use attribute mapping to assign particular graphical properties to specific values or ranges of values regardless of data order or the data range. Attribute maps enable you to ensure that particular visual attributes are applied based on the value of the data instead of the position of the data in the data set. For example, if you create a graph that plots items sold in different countries, you might specify the display attributes for the sales data of each country.

Task	Description
	See Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.
Add annotations	The annotation feature enables you to add text, shapes, images, and other annotations to graph output. See Chapter 20, "Annotating ODS Graphics," on page 1935.

Where to Go from Here

Table 1.5 *Tasks for Exploration*

Task	Topic
Explore the charts and plots that you can create with the procedures.	Chapter 2, "Gallery of Plots and Charts," (p. 15)
Learn more about the elements of a typical procedure.	Chapter 4, "Elements of a Program," (p. 81)
Learn more about the procedures and what they can do.	Chapter 3, "Concepts for Creating Graphs," (p. 69)
Learn about ODS destinations and styles.	"About ODS Destinations and Styles " (p. 79)
Learn about the ODS GRAPHICS statement.	"Using the ODS GRAPHICS Statement" (p. 1886)

2

Gallery of Plots and Charts

Alphabetical List of Plots and Charts **16**
See Also 16

Basic Plots and Charts **16**
About Basic Plots and Charts 16
About Band Plots 17
About Block Plots 18
About Bubble Plots 19
About Fringe Plots 20
About Heat Maps 21
About High-Low Plots 23
About Lines 25
About Needle Plots 29
About Scatter Plots 31
About Series Plots 33
About Spline Plots 34
About Step Plots 35
About Text Plots 36
About Text Insets 38
About Vector Plots 39

Fit and Confidence Plots **41**
About Fit and Confidence Plots 41
About Ellipse Plots 42
About Loess Plots 43
About Penalized B-Spline Plots 44
About Regression Plots 45

Distribution Plots **47**
About Distribution Plots 47
About Box Plots 47
About Density Plots 49
About Histograms 51

Categorization Plots and Charts **52**
About Categorization Plots and Charts 52
About Bar Charts 53
About Dot Plots 60
About Line Charts 61
About Pie and Donut Charts (Preproduction) 63
About Waterfall Charts 64

Alphabetical List of Plots and Charts

The following table lists all of the plots and charts in alphabetical order for easy access. Select a plot or chart to see its description.

Band plot
Bar chart
Block plot
Box plot
Bubble plot
Density plot
Donut chart
Dot plot
Ellipse plot
Fringe plot
Heat map
High-Low plot
Histogram
Line chart
Line, drop
Line, parameterized
Line, reference
Loess plot
Needle plot
Penalized B-Spline plot
Pie chart
Regression plot
Scatter plot
Series plot
Spline plot
Step plot
Text Inset
Text plot
Vector plot
Waterfall chart

See Also

"Plot Type Compatibility" on page 1779

Basic Plots and Charts

About Basic Plots and Charts

You can use the SGPLOT and SGPANEL procedures to produce basic plots and charts.

The plot and chart statements include options for controlling how the output is displayed. Many of the options are unique to the particular plot or chart. However, some general options apply to most of the basic plots and charts.

For example, options enable you to do the following:

- specify colors, line attributes, and other visual features.
- group the data by the values of a variable. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.
- use a secondary axis (X2 or Y2). This option is available only for the SGPLOT procedure.
- reference an ID variable in attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

The basic plots and charts are described in the following sections. If you run the examples, your output might differ somewhat depending on the size of your graphics. The examples here were specified to be a particular size, usually 4 inches or 4.5 inches wide. You can specify the size in the ODS GRAPHICS statement. See "WIDTH=*dimension*" on page 123.

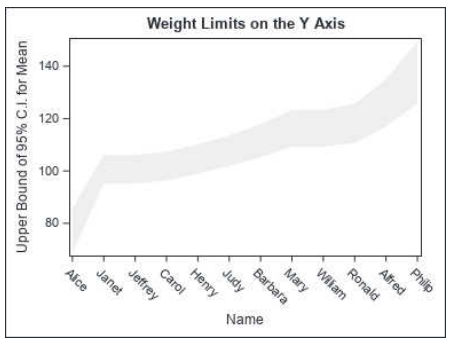
See Also

"Plot Type Compatibility" on page 1779

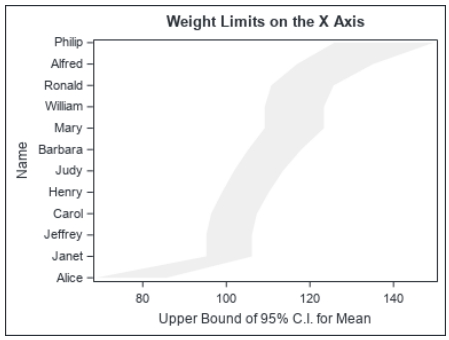
About Band Plots

A band plot creates a band that highlights part of the plot and shows upper and lower limits. The input data should be sorted by the X or Y variable.

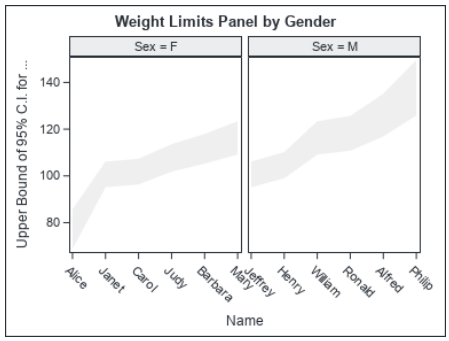
The following examples show upper and lower mean weight values for a class of students. The first two examples use the SGPLOT procedure to show the same band plotted along the X axis and the Y axis, respectively. The third example uses the SGPANEL procedure to show a matrix that is paneled by gender.




```
title "Weight Limits on the Y Axis";
proc sgplot data=sashelp.classfit;
  where age > 12;
  band x=name lower=lowermean upper=uppermean;
run;
title;
```




```
title "Weight Limits on the X Axis";
proc sgplot data=sashelp.classfit;
  where age > 12;
  band y=name lower=lowermean upper=uppermean;
run;
title;
```




```
title "Weight Limits Panel by Gender";
proc sgpanel data=sashelp.classfit;
  panelby sex / uniscale=row
                spacing=8;
  where age > 12;
  band x=name lower=lowermean upper=uppermean;
run;
title;
```

Options are available that enable you to customize the band plot and enhance its appearance. For example, you can do the following:

- add labels to the upper and lower edges of the band, specify how the labels are positioned, and set other attributes for the labels
- specify fill and outline attributes
- specify legend labels and plot transparency

Note: This list does not include all available options.

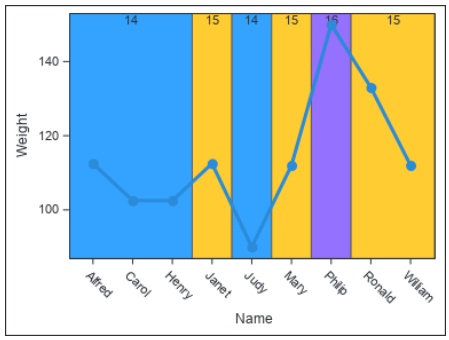
See Also

- "BAND Statement" on page 188 (SGPANEL procedure)
- "BAND Statement" on page 907 (SGPLOT procedure)

About Block Plots

Block plots show continuous ranges of data that have the same BLOCK= value. Block plots are available in the SGPLOT and SGPANEL procedures.

In the following example, blocks represent students of different ages.




```
proc sgplot data=sashelp.class;
  where age>13;
  block x=name block=age;
  series x=name y=weight / markers;
run;
```

Block plots can also be used with an interval (numeric linear or time) X variable.

Options are available that enable you to customize the block plot and enhance its appearance. For example, you can do the following:

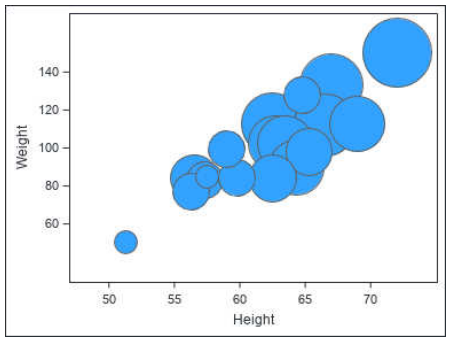
- specify attributes for the blocks, such as outlines, fill color, and transparency.
- alternate the fill color used by the blocks.
- control the display of the text within the blocks. For example, you can split block text into multiple lines in order to fit within the containing block.

Note: This list does not include all available options.

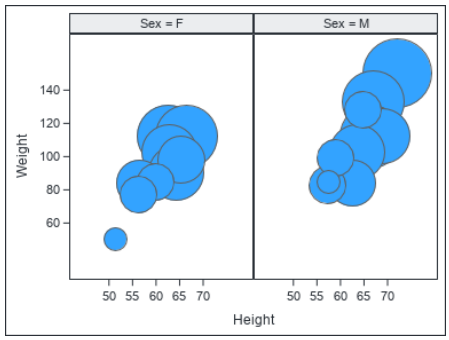
About Bubble Plots

Bubble plots show the relative magnitude of the values of a variable. The values of two variables determine the position of the bubble on the plot, and the value of a third variable determines the size of the bubble.

The following examples show the height and weight values for a class. The size of each bubble is determined by the student's age. Examples are provided for the SGPLOT and SGPANEL procedures.




```
proc sgplot data=sashelp.class;
  bubble x=height y=weight size=age;
run;
```




```
proc sgpanel data=sashelp.class;
  panelby sex;
  bubble x=height y=weight size=age;
run;
```

Options are available that enable you to customize the bubble plot and enhance its appearance. For example, you can do the following:

- control the size of the largest and the smallest bubble
- specify fill and outline attributes, and data labels and their attributes
- specify legend labels, plot transparency, and URLs for web pages to be displayed when parts of the plot are selected within an HTML page

Note: This list does not include all available options.

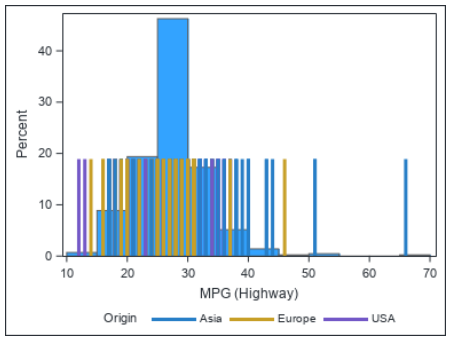
See Also

- "BUBBLE Statement" on page 207 (SGPANEL procedure)
- "BUBBLE Statement" on page 930 (SGPLOT procedure)

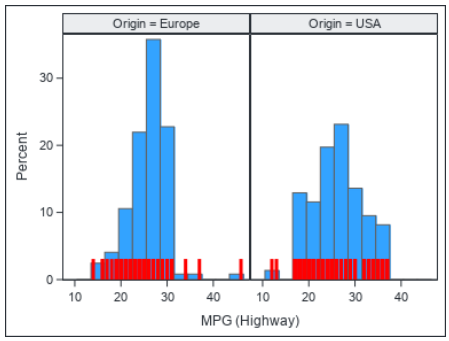
About Fringe Plots

A fringe plot displays data values as a fringe on the X or X2 axis of the plot, and often is used to enhance a histogram or a density plot.

The following examples use fringe lines to compare miles-per-gallon ratings for vehicles from different regions.




```
proc sgplot data=sashelp.cars;
histogram mpg_highway;
fringe mpg_highway / group=origin height=1in;
run;
```




```
proc sgpanel data=sashelp.cars noautolegend;
where origin in("Europe" "USA");
panelby origin;
histogram mpg_highway;
fringe mpg_highway / height=20 lineattrs=(color=red);
run;
```

Options are available that enable you to customize the fringe plot and enhance its appearance. For example, you can do the following:

- specify the height of the fringe lines, and control line attributes
- control the data tips that are displayed and their attributes
- specify legend labels and plot transparency

Note: This list does not include all available options.

See Also

- "FRINGE Statement" on page 267 (SGPANEL procedure)
- "FRINGE Statement" on page 998 (SGPLOT procedure)

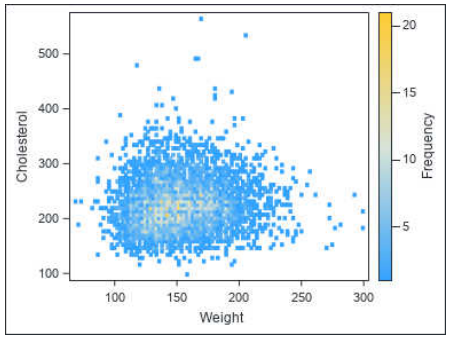
About Heat Maps

Heat maps present data values as rectangular areas with varying colors. The rectangles correspond to X and Y crossings.

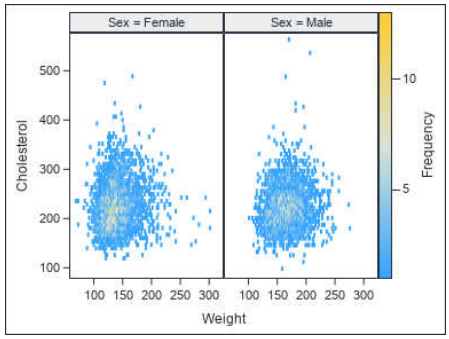
In a heat map, each variable range is subdivided into equal size bins to create a rectangular grid of bins. The frequency or some other statistic is computed for the

response variable. The grid is displayed by coloring each bin with a shade of color computed from a color gradient.

The following two examples create heat maps for participants in a heart disease study. The graphs plot weight and cholesterol. The heat map colors correspond to the frequency for the cholesterol count, as indicated by the gradient legend.

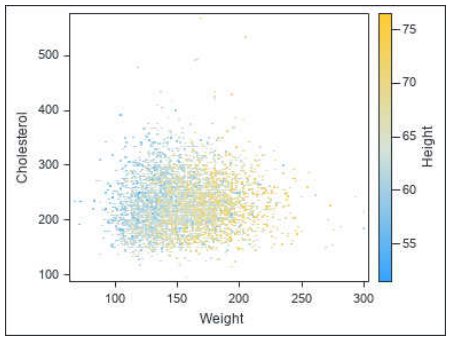



```
proc sgplot  data=sashelp.heart;
  heatmap x=weight y=cholesterol;
run;
```

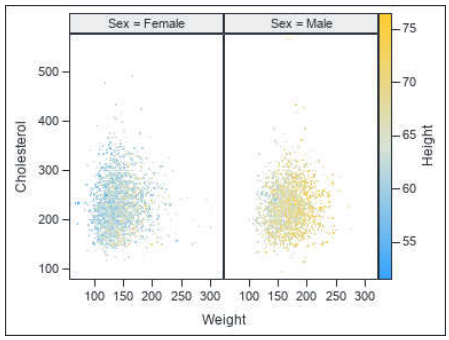



```
proc sgpanel data=sashelp.heart;
  panelby sex;
  heatmap x=weight y=cholesterol;
run;
```

Parameterized heat maps create colored rectangles for each X and Y crossing based on specified response values.




```
ods graphics / nxybinsmax=120000;
proc sgplot  data=sashelp.heart;
  heatmapparm x=weight y=cholesterol
     colorresponse=height;
run;

ods graphics / reset=nxybinsmax;
```




```
ods graphics / nxybinsmax=120000;
proc sgpanel data=sashelp.heart;
  panelby sex;
  heatmapparm x=weight y=cholesterol
     colorresponse=height;
run;

ods graphics / reset=nxybinsmax;
```

Options are available that enable you to customize the heat map and enhance its appearance. For example, you can do the following:

- (HEATMAP statement only) control aspects of the bins. For example, you can specify the sizes of the bins.
- control the color ramp that is used for the color response.
- (HEATMAP statement only) specify the statistic to use for the response variable.
- specify outlines for the colored rectangles.
- control data tips. For example, you can specify the information to display as well as the format.

Note: This list does not include all available options.

See Also

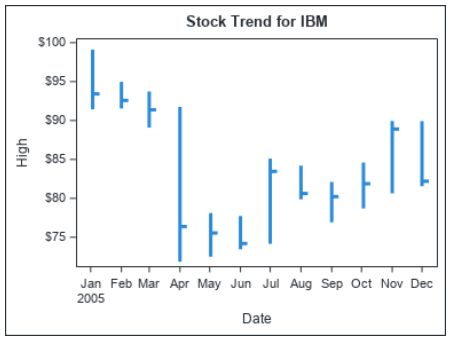
- "HEATMAP Statement" on page 363 (SGPANEL procedure)
- "HEATMAP Statement" on page 1097 (SGPLOT procedure)
- "HEATMAPPARM Statement" on page 371 (SGPANEL procedure)
- "HEATMAPPARM Statement" on page 1109 (SGPLOT procedure)

About High-Low Plots

High-low plots show how several values of one variable relate to one value of another variable. Typically, each variable value on the horizontal axis has several corresponding values on the vertical axis.

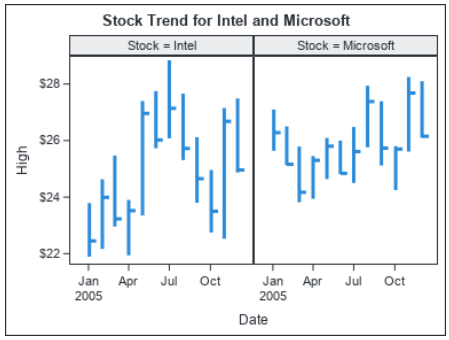
The following examples show the stock trend for IBM during a particular year. The first two examples use the SGPLOT procedure to show the same plot along the X axis and the Y axis, respectively. The third example uses the SGPANEL procedure

to show a paneled graph for Intel and Microsoft stock prices in the same year. Optional values have been specified for the closing stock prices, which are represented as tick marks on the high-low lines.




```
title "Stock Trend for IBM";
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2005"d and stock = "IBM"));
  highlow x=date high=high low=low
   / close=close;
run;
title;
```

```
title "Stock Trend for IBM";
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2005"d and stock = "IBM"));
  highlow y=date high=high low=low
   / close=close;
run;
title;
```




```
title "Stock Trend for Intel and Microsoft";
proc sgpanel data=sashelp.stocks
  (where=(date >= "01jan2005"d and
  (stock = "Intel" or stock = "Microsoft")));
  panelby stock;
  highlow x=date high=high low=low
   / close=close;
run;
title;
```

Options are available that enable you to customize the high-low plot and enhance its appearance. For example, you can do the following:

- use bars instead of lines to represent the data. If you use bars, then you can specify the fill and outline attributes for the bars.
- show tick marks for the open and closing values.
- specify labels and arrowheads for the high and low values.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the order of lines or bars within a group.
- specify legend labels, plot transparency, and URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

See Also

- "HIGHLOW Statement" on page 380 (SGPANEL procedure)
- "HIGHLOW Statement" on page 1118 (SGPLOT procedure)

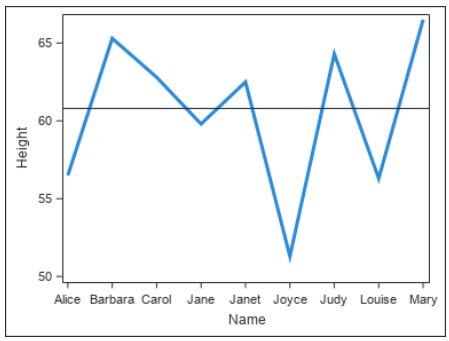
About Lines

About Reference Lines

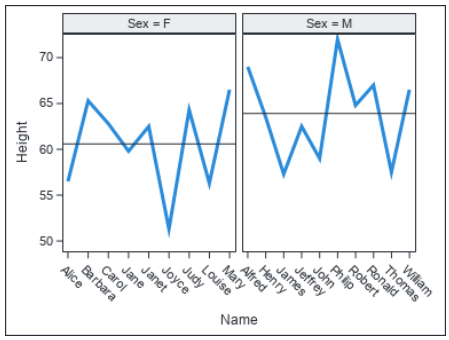
You can add horizontal or vertical reference lines to your graphics. You can draw a reference line for each value of a specified variable. Or you can specify one or more explicit values for the reference lines.

The following examples show the height values for a class of students. A horizontal reference line is overlaid on a series plot to show the average height. Examples are provided for the SGPLOT and SGPANEL procedures.

In the first example, a value of 60.8 is specified for the reference line. The second example uses the MEANS procedure to obtain the averages for males and females in the class. The SGPANEL procedure then specifies the variable that contains these averages in order to obtain the reference lines.




```
proc sgplot data=sashelp.class;
  where (sex="F");
  series x=name y=height;
  refline 60.8;
run;
```




```
proc means data=sashelp.class mean;
  var height;
  class sex;
  output out= classAverages
    mean=averageWeight;
run;

data classHeightAverage;
  set sashelp.class work.classAverages;
run;

proc sgpanel data=classHeightAverage;
  panelby sex / uniscale=row
    spacing=8;
  series x=name y=height;
  refline averageWeight;
run;
```

Options are available that enable you to customize the reference line and enhance its appearance. For example, you can do the following:

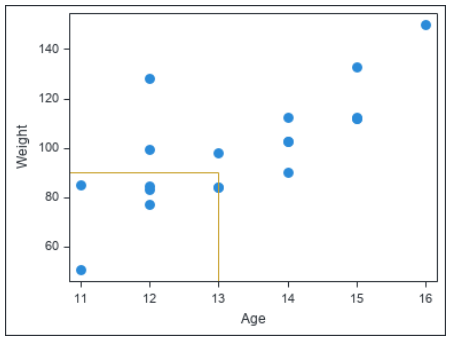
- specify a horizontal or vertical line. In the SGPLOT procedure, you can associate the line with a secondary axis.
- specify line attributes, labels, and label attributes.
- specify legend labels and line transparency.
- specify an amount to offset all lines from discrete axis values.
- extend the plot axes to contain the reference lines.

Note: This list does not include all available options.

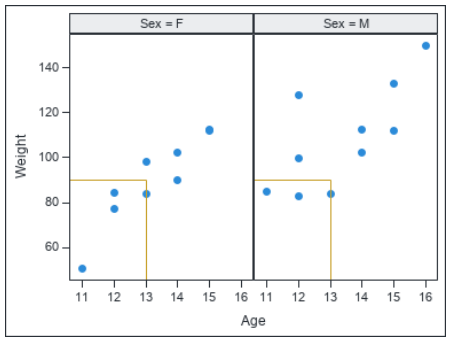
About Drop Lines

You can create one or more drop lines from data points to one or both axes. The lines can be horizontal, vertical, or both. A drop line is always drawn perpendicular from the specified point to the X, X2, Y, or Y2 axis.

The following examples show the weight values for a class of students. The example drops lines from the point (13, 90) to the respective points on both axes. The LINEATTRS= option specifies that line appearance is determined by the GraphData2 style element. The examples show drop lines in the SGPLOT and SGPANEL procedures.




```
proc sgplot data=sashelp.class;
   scatter x=age y=weight;
   dropline x=13 y=90 / dropto=both
     lineattrs=GraphData2;
run;
```




```
proc sgpanel data=sashelp.class;
   panelby sex;
   scatter x=age y=weight;
   dropline x=13 y=90 / dropto=both
     lineattrs=GraphData2;
run;
```

Options are available that enable you to customize the drop line and enhance its appearance. For example, you can do the following:

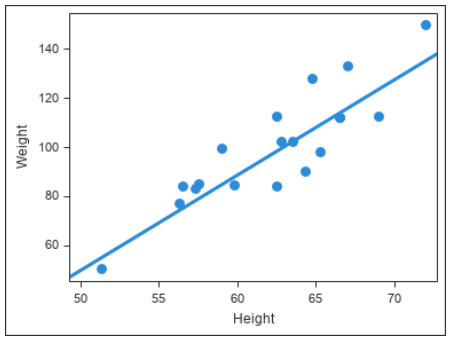
- specify a horizontal line, a vertical line, or both. In the SGPLOT procedure, you can associate the line with a secondary axis.
- draw a drop line for each value of a specified variable.
- specify line attributes, labels, and label attributes.
- specify legend labels.
- specify an amount to offset all lines from discrete axis values.
- extend the plot axes to contain the drop lines.

Note: This list does not include all available options.

About Parameterized Lines

Parameterized lines are straight lines specified by a point and a slope. The statement must be used with another plot statement that is derived from data values that provide boundaries for the axis area. For example, the LINEPARM statement can be used with a scatter plot or a histogram.

The following example shows weight with respect to height for a class of students. A single line is generated by specifying values for the point and for the slope. The line in the example approximates a line of best fit.




```
proc sgplot data=sashelp.class
    noautolegend;
  scatter x=height y=weight;
  lineparm x=50 y=50 slope=3.89;
run;
```

You can generate multiple lines by specifying a numeric variable for any or all required arguments. Examples are provided for the SGPLOT and SGPANEL procedures. The following two examples create lines of best fit for male and female participants in a heart disease study. The lines show weight with respect to height.

The examples first sort the data set by male and female participants. The sorted data is written to a data set named HEART.

```
proc sort data=sashelp.heart(keep=height weight sex)
     out=heart;
  by sex;
run;
```

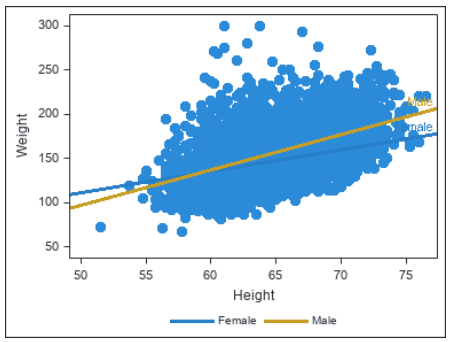
The examples then use the REG procedure and output the regression statistics to a data set named STATS. The STATS data set includes the slope and the Y-intercept for the regression.

```
proc reg data=heart
     outest=stats(rename=(height=slope));
  by sex;
  model weight=height;
run;
```

Finally, the examples merge the HEART and STATS data sets.

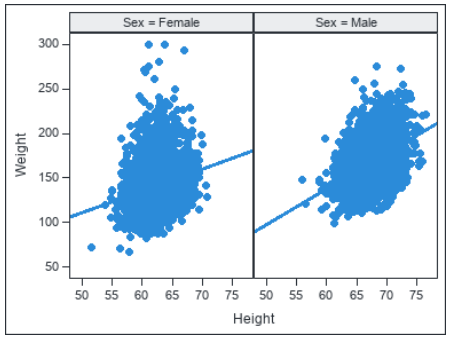
```
data heartStats;
  merge heart stats(keep=intercept slope sex);
run;
```

The first example uses the SGPLOT procedure to show lines of best fit for females and males in the study. The regression lines are labeled and have their own legend.




```
proc sgplot data=heartStats;
  scatter x=height  y=weight;
  lineparm x=0 y=intercept slope=slope /
     name="Line" group=sex
     curvelabel;
  keylegend "Line";
  xaxis min=50;
  yaxis min=50;
run;
```

The following example uses the SGPANEL procedure to create the same information, which is paneled by gender.




```
proc sgpanel data=heartStats
   noautolegend;
  panelby sex;
  scatter x=height  y=weight;
  lineparm x=0 y=intercept slope=slope;
  colaxis min=50;
  rowaxis min=50;
run;
```

Options are available that enable you to customize the line and enhance its appearance. For example, you can do the following:

- specify line attributes, labels, and label attributes
- specify legend labels and line transparency
- prevent the line from being extended beyond the axis offset

Note: This list does not include all available options.

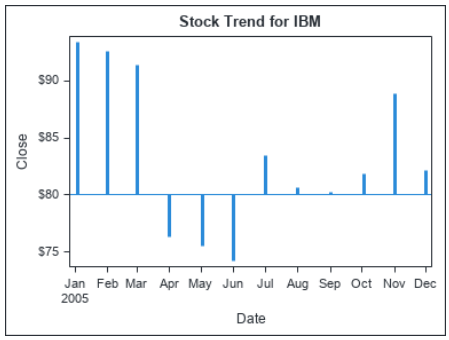
See Also

- “REFLINE Statement” on page 513 (SGPANEL procedure)
- “REFLINE Statement” on page 1260 (SGPLOT procedure)
- “LINEPARM Statement” on page 445 (SGPANEL procedure)
- “LINEPARM Statement” on page 1188 (SGPLOT procedure)

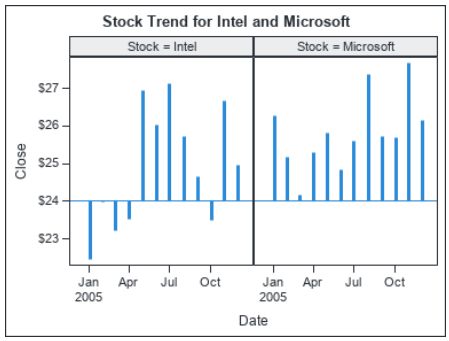
About Needle Plots

Needle plots use vertical line segments, or needles, to connect each data point to a baseline.

The following examples show the stock trend during a particular year. Examples are provided for the SGPLOT and SGPANEL procedures. Each example specifies an optional baseline value on the Y axis.




```
title "Stock Trend for IBM";
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2005"d and stock = "IBM"));
  needle x=date y=close / baseline=80;
run;
title;
```




```
title "Stock Trend for Intel and Microsoft";
proc sgpanel data=sashelp.stocks
  (where=(date >= "01jan2005"d and
  (stock = "Intel" or stock = "Microsoft")));
  panelby stock;
  needle x=date y=close / baseline=24;
run;
title;
```

Options are available that enable you to customize the needle plot and enhance its appearance. For example, you can do the following:

- specify a baseline value, as shown in the example.
- add markers to the tips of the needles and specify marker attributes.
- add data labels and specify label attributes.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the order of needles within a group.
- specify an amount to offset all needle lines from discrete X values.
- specify legend labels, plot transparency, and URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

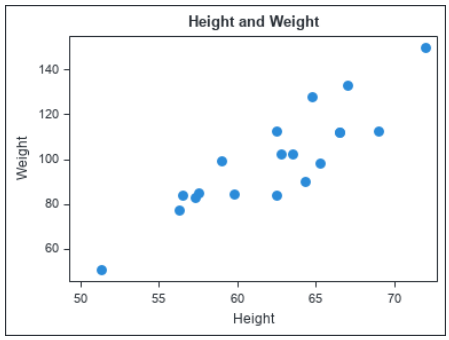
See Also

- “NEEDLE Statement” on page 465 (SGPANEL procedure)
- “NEEDLE Statement” on page 1210 (SGPLOT procedure)

About Scatter Plots

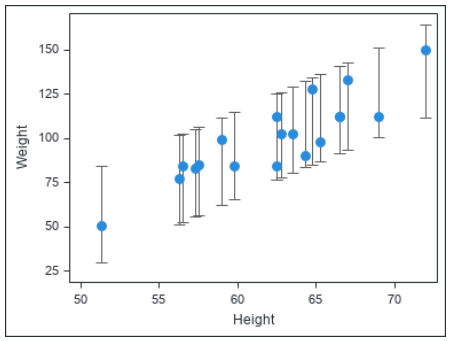
Scatter plots show the relationship of one variable to another, often revealing concentrations or trends in the data. Typically, each variable value on the horizontal axis can have any number of corresponding values on the vertical axis.

The following examples show the relationship of height to weight for a class of students. Examples are provided for the SGPLOT, SGPANEL, and SGSCATTER procedures.



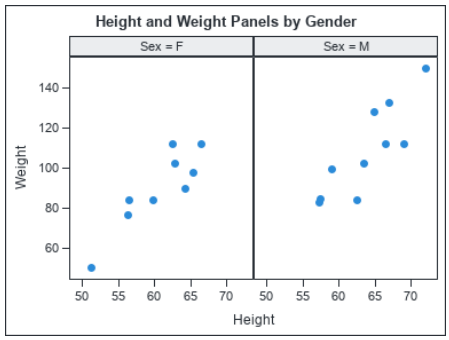
This code creates a basic scatter plot of Height and Weight.

```
title "Height and Weight";
proc sgplot data=sashelp.class;
  scatter x=height y=weight;
run;
title;
```



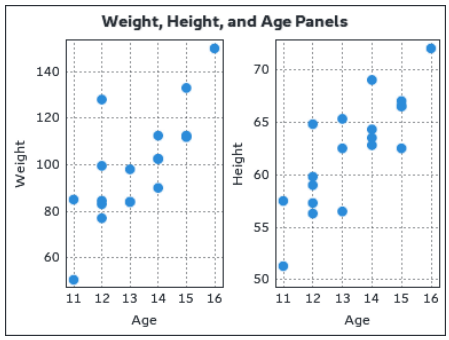
This code creates a scatter plot that includes error bars.

```
proc sgplot data=sashelp.classfit;
scatter x=height y=weight /
  yerrorlower=lower yerrorupper=upper;
run;
```



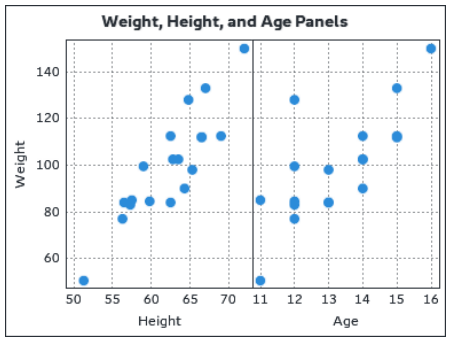
This code creates a panel of scatter plots by Sex.

```
title "Height and Weight Panels by Gender";
proc sgpanel data=sashelp.class;
  panelby sex;
  scatter x=height y=weight;
run;
title;
```



This code creates a panel of scatter plots.

```
title "Weight, Height, and Age Panels";
proc sgscatter data=sashelp.class;
   plot weight*age height*age;
run;
```



This code creates a comparative panel of scatter plots.

```
title "Weight, Height, and Age Panels";
proc sgscatter data=sashelp.class;
   compare y=weight
      x=(height age);
run;
```

Options are available that enable you to customize the scatter plot and enhance its appearance. For example, when using the SGPLOT or SGPANEL procedure, you can specify options to do the following:

- add and customize error bars. The previous example shows error bars.
- specify how many times observations are repeated for computational purposes.
- specify marker attributes, data labels, and label attributes.
- specify an amount to offset all markers from discrete axis values.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the order of markers within a group
- specify legend labels, plot transparency, and URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

For more information, refer to the procedure documentation.

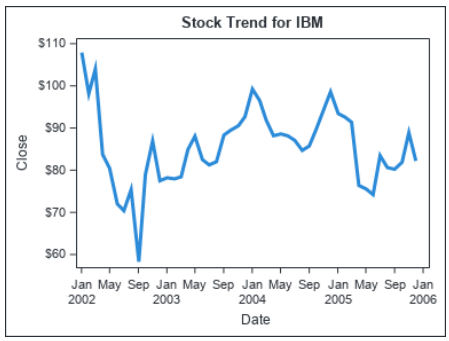
See Also

- “SCATTER Statement” on page 1283 (SGPLOT procedure)
- “SCATTER Statement” on page 535 (SGPANEL procedure)
- Chapter 12, “SGSCATTER Procedure,” on page 1699

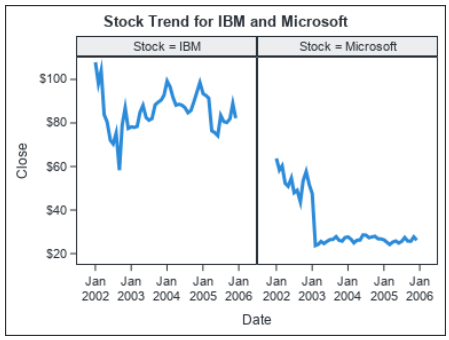
About Series Plots

Series plots display a series of line segments that connect observations of input data.

The following examples show series plots of stock trends. Examples are provided for the SGPLOT and SGPANEL procedures.




```
title "Stock Trend for IBM";
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2002"d
    and stock = "IBM"));
  series x=date y=close;
run;
title;
```




```
title "Stock Trend for IBM and Microsoft";
proc sgpanel data=sashelp.stocks
  (where=(date >= "01jan2002"d and
  (stock = "IBM" or stock = "Microsoft")));
  panelby stock;
  series x=date y=close;
run;
title;
```

Options are available that enable you to customize the series plot and enhance its appearance. For example, you can do the following:

- specify line and marker attributes, data labels, curve labels, and label attributes.
- specify an amount to offset all markers from discrete axis values.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the order of lines within a group
- specify legend labels, plot transparency, and URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

See Also

- "SERIES Statement" on page 553 (SGPANEL procedure)
- "SERIES Statement" on page 1301 (SGPLOT procedure)

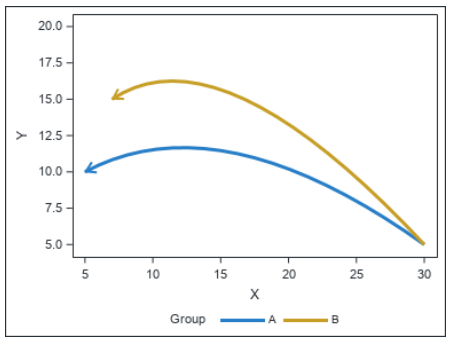
About Spline Plots

Spline plots are similar to series plots but have a quadratic Bézier spline interpolation that produces smooth curves.

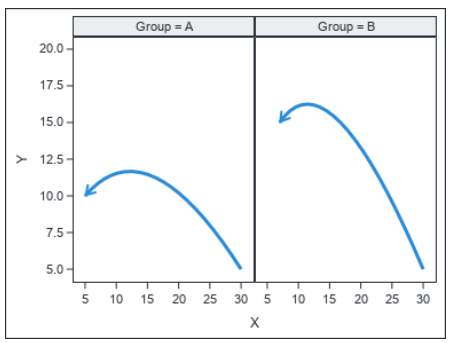
The following examples show spline plots for the SGPLOT and SGPANEL procedures. Each spline curve consists of three points. The curves are smooth and do not pass through the middle point.

Both examples use the following data:

```
data lines;
  input X Y Group $;
datalines;
30 5 A
15 15 A
5 10 A
30 5 B
15 20 B
7 15 B
;
run;
```




```
proc sgplot data=lines;
  spline y=y x=x / group=group
    arrowheadpos=end;
run;
```




```
proc sgpanel data=lines;
panelby group;
  spline y=y x=x / arrowheadpos=end;
run;
```

Options are available that enable you to customize the spline plot and enhance its appearance. For example, you can do the following:

- add arrows to the spline plot, as shown in the examples.
- specify line and arrow attributes, curve labels, and label attributes.
- specify an amount to offset all lines from discrete axis values.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the order of lines within a group.
- specify legend labels, plot transparency, and URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

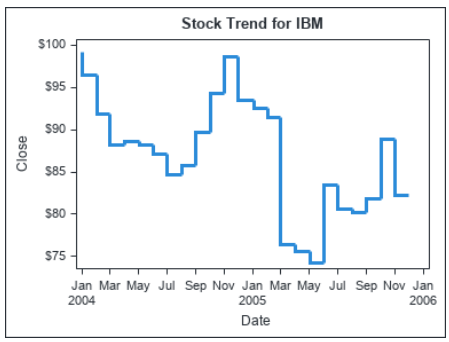
See Also

- "SPLINE Statement" on page 574 (SGPANEL procedure)
- "SPLINE Statement" on page 1327 (SGPLOT procedure)

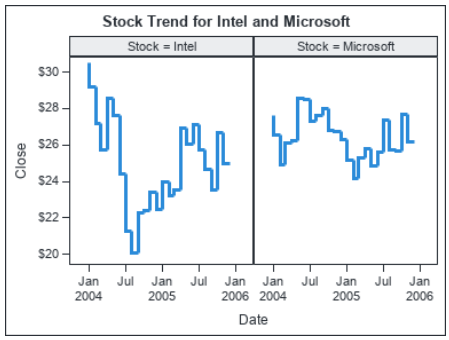
About Step Plots

Step plots display a series of horizontal and vertical line segments that connect observations of input data. The plots use a step function to connect the data points. The vertical line can change at each step.

The following examples show step plots of stock trends. Examples are provided for the SGPLOT and SGPANEL procedures.




```
title "Stock Trend for IBM";
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2004"d and stock = "IBM"));
  step x=date y=close;
run;
```




```
title "Stock Trend for Intel and Microsoft";
proc sgpanel data=sashelp.stocks
  (where=(date >= "01jan2004"d and
  (stock = "Intel" or stock = "Microsoft")));
  panelby stock;
  step x=date y=close;
run;
title;
```

Options are available that enable you to customize the step plot and enhance its appearance. For example, you can do the following:

- add and customize markers and error bars.
- specify line attributes, data labels, curve labels, and label attributes.
- specify an amount to offset all step lines from discrete axis values.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the order of lines within a group
- specify legend labels, plot transparency, and URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

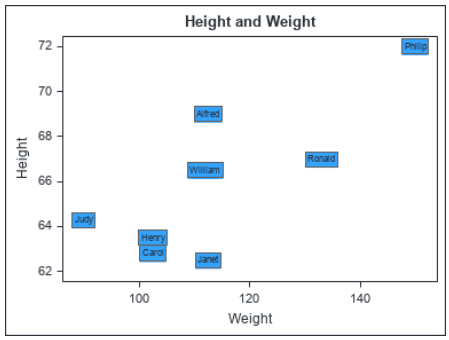
See Also

- "STEP Statement" on page 589 (SGPANEL procedure)
- "STEP Statement" on page 1343 (SGPLOT procedure)

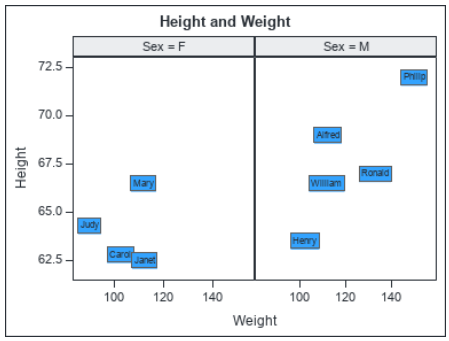
About Text Plots

The TEXT statement displays the associated text values at (X, Y) locations in the graph. The text can be numbers or characters.

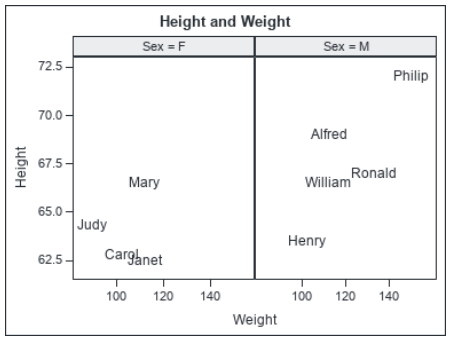
The following examples show the relationship of height to weight for a class of students. Examples are provided for the SGPLOT and SGPANEL procedures. The first two examples apply fills and outlines to the text markers.




```
title "Height and Weight";
proc sgplot data=sashelp.class;
 where (age>13);
 text x=weight y=height text=name /
  outline backfill;
run;
title;
```




```
title "Height and Weight";
proc sgpanel data=sashelp.class;
 where (age>13);
 panelby sex;
 text x=weight y=height text=name /
  outline backfill;
run;
title;
```




```
title "Height and Weight";
proc sgpanel data=sashelp.class;
 where (age>13);
 panelby sex;
 text x=weight y=height
   text=name /
   textattrs=(size=10);
run;
title;
```

Although you can achieve similar results using the SCATTER statement with the MARKERCHAR= option, the TEXT statement gives you more control over the appearance of the text. For example, you can do the following:

- add fills and outlines to the text markers, as shown in the previous examples. You can also specify a back-light effect for the text.
- rotate the text, reposition the text, and specify text attributes such as font size.
- split the text at one or more specified characters.
- control the display of grouped data. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the axis to use for clustering.
- specify legend labels, plot transparency, and URLs for web pages that are displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

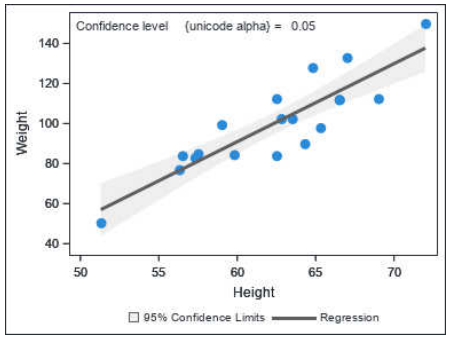
See Also

- "TEXT Statement" on page 618 (SGPANEL procedure)
- "TEXT Statement" on page 1373 (SGPLOT procedure)

About Text Insets

A text inset provides an easy way to add text to a graphic. You can insert a text string as well as a series of label-value pairs.

The following example shows a linear regression curve with a text inset in the upper left corner. This text inset specifies a text string and is available only with the SGPLOT procedure.

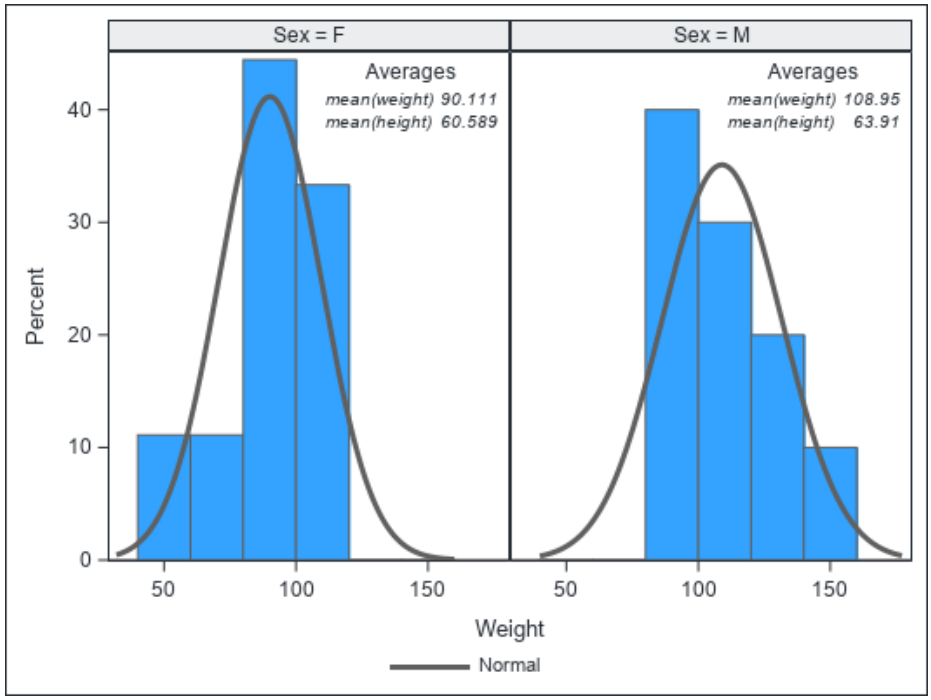



```
proc sgplot data=sashelp.class;
  reg x=height y=weight / CLM;
  inset ("Confidence level (*ESC*)
    {unicode alpha} =" ="0.05");
run;
```

Insets are also available with the SGPANEL procedure. Unlike the SGPLOT procedure, panel insets do not accept text strings as arguments. The INSET statement in the SGPANEL procedure generates data-driven text from one or more variables. Typically, the variable is a computed numeric value, such as a mean or a sum. For non-computed variables, the statement displays the value of the first observation for each classification.

The inset labels are derived from the variable labels, or variable names if the labels are not present. (You can suppress the labels using a NOLABEL option.) The inset values come from the variable data.

The following example shows a panel of histograms along with a text inset for each panel.



To see the code for this example, see "Example: Panel with Insets" on page 429.

Options are available that enable you to customize the text inset and enhance its appearance. For example, you can do the following:

- show or hide a border
- position the text box within the plot
- specify text attributes, add a title, and specify title attributes
- align the labels and values when you specify label-value pairs (SGPLOT procedure only)

Note: This list does not include all available options.

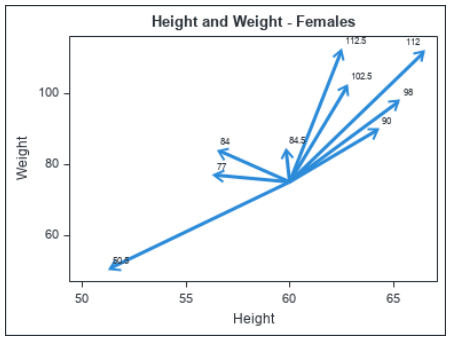
See Also

- "INSET Statement" on page 1166 (SGPLOT procedure)
- "INSET Statement" on page 426 (SGPANEL procedure)

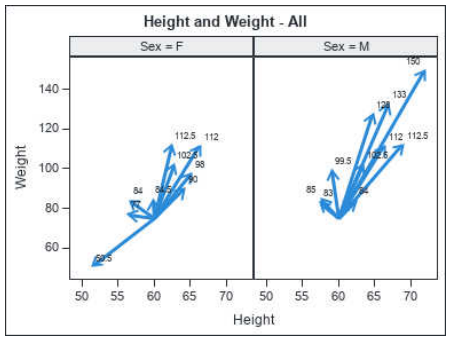
About Vector Plots

Vectors are directed line segments. A vector plot is a two-dimensional graphic that uses vectors to represent both direction and magnitude at each point.

The following examples show the relationship of height to weight for a class of students. Examples are provided for the SGPLOT and SGPANEL procedures. Both examples specify optional X and Y origins and data labels.




```
title "Height and Weight - Females";
proc sgplot data=sashelp.class;
  where(sex="F");
  vector x=height y=weight
   / yorigin=75 xorigin=60
     datalabel;
run;
title;
```




```
title "Height and Weight - All";
proc sgpanel data=sashelp.class;
  panelby sex;
  vector x=height y=weight
   / yorigin=75 xorigin=60
     datalabel;
run;
title;
```

Options are available that enable you to customize the vector plot and enhance its appearance. For example, you can do the following:

- specify the origin, as shown in the examples.
- specify line attributes, data labels, and data label attributes. You can also show or hide the arrows, change the arrowhead shape, and the change the arrow direction.
- specify legend labels and plot transparency.

Note: This list does not include all available options.

See Also

- “VECTOR Statement” on page 724 (SGPANEL procedure)
- “VECTOR Statement” on page 1482 (SGPLOT procedure)

Fit and Confidence Plots

About Fit and Confidence Plots

You can use the SGPLOT and SGPANEL procedures to produce fit plots and ellipses (the ellipses plot is available with the SGPLOT procedure only). Fit plots represent the line of best fit (trend line) with confidence limits.

The plot statements include many options for controlling how the output is displayed. The options that are available depend on the plot type. However, some general options apply to most of the fit and confidence plots. For example, options enable you to do the following:

- add confidence limits (CLM) to the plot. When you add CLM limits, you can specify the confidence level, the transparency for the confidence limits, and other visual attributes. You can add CLM limits to loess, penalized B-spline, and regression plots.
- add prediction limits (CLI) for the individual predicted values. When you add CLI limits, you can specify the text that appears for the limits and other visual attributes. You can add CLI limits to penalized B-spline and regression plots.
- control the appearance of the markers and the fit line. You can also specify a smoothing parameter.
- add and customize curve and data labels.
- specify legend labels. You can also show or hide the legend entries for the CLM limits, the CLI limits, and the fit line.
- group the data by the values of a variable. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.
- specify the value of an ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

Note: Not all of these features are available for all of the plots. Also, the list does not include all available options.

The fit and confidence plots are described in the following sections. If you run the examples, your output might differ somewhat depending on the size of your graphics. The examples here were specified to be a particular size, usually 4 inches or 4.5 inches wide. You can specify the size in the ODS GRAPHICS statement. See "WIDTH=*dimension*" on page 123.

See Also

"Plot Type Compatibility" on page 1779

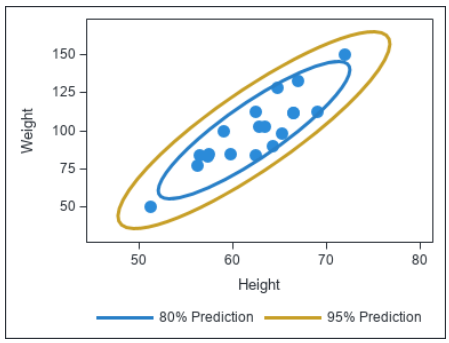
About Ellipse Plots

Ellipse plots create a confidence or prediction elliptical curve computed from input data. In order to produce useful output, the ELLIPSE statement should be used with another plot statement that uses numeric axes. When creating an ellipse with the SGPANEL procedure, the ellipse must be used with another plot.

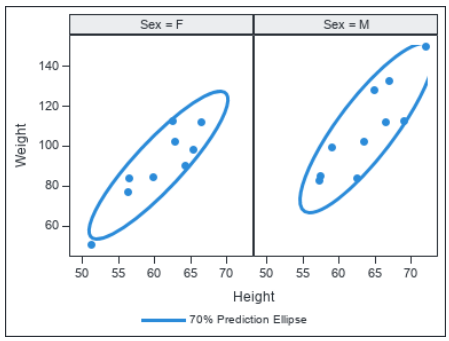
The following examples show the relationship of height to weight for a class of students.

Here are the noteworthy features of the examples:

- The ELLIPSE statements use TYPE=PREDICTED. This is the default.
- In the SGPLOT procedure, one ELLIPSE statement uses ALPHA=.2 and the other uses ALPHA=.05. The ELLIPSE statement in the SGPANEL procedure uses ALPHA=.3.
- A legend was created with entries only for the ellipses. Each ELLIPSE statement specifies a plot name. The KEYLEGEND statement uses the NAME value to determine the entries in the legend.




```
proc sgplot data=sashelp.class;
  scatter x=height y=weight;
  ellipse x=height y=weight /
     alpha=.2
     name="eighty"
     legendlabel="80% Prediction";
  ellipse x=height y=weight /
     alpha=.05
     name="ninetyfive"
     legendlabel="95% Prediction";
  keylegend "eighty" "ninetyfive";
run;
```

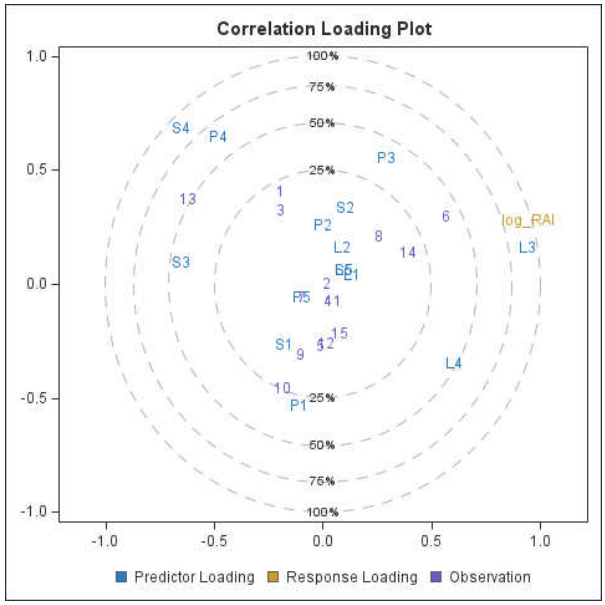



```
proc sgpanel data=sashelp.class;
  panelby sex;
  scatter x=height y=weight;
  ellipse x=height y=weight /
     alpha=.3
     name="seventy";
  keylegend "seventy";
run;
```

The ELLIPSEPARM statement plots an ellipse with specified axis lengths, a specified slope for the major axis, and the ellipse center.

The ELLIPSEPARM statement does not perform computations on input data to derive the location and shape of the ellipse. Another plot that is derived from data values that provide boundaries for the axis area must be included in the graph.

This example resembles a correlation loading plot that is produced using the PLS procedure in the SAS/STAT product. See Spectrometric Calibration in *SAS/STAT User's Guide*.



For details and code, see "Example: Simple Correlation Loading Plot" on page 996.

See Also

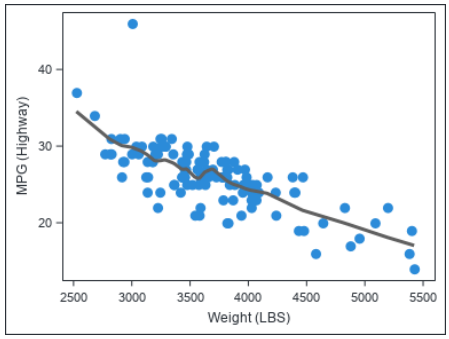
- "ELLIPSE Statement" on page 979 (SGPLOT procedure)
- "ELLIPSE Statement" on page 252 (SGPANEL procedure)
- "ELLIPSEPARM Statement" on page 988 (SGPLOT procedure)
- "ELLIPSEPARM Statement" on page 260 (SGPANEL procedure)

About Loess Plots

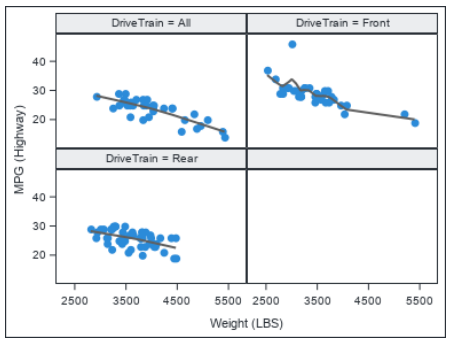
A loess plot includes a scatter plot of two numeric variables along with an overlaid nonlinear fit line that enables you to perform locally weighted polynomial regression. You can specify the degree of the local polynomials to use for each local regression. You can also change the default smoothing technique that is applied to the fit.

The following examples show the relationship of height to weight and the line of best fit for a class of students. Examples are provided for the SGPLOT and

SGPANEL procedures. In both examples, the automatically generated legend for the fit line is not needed and has been suppressed.




```
proc sgplot data=sashelp.cars
     noautolegend;
  where (origin = "Europe");
  loess x=weight y=mpg_highway;
run;
```




```
proc sgpanel data=sashelp.cars
      noautolegend;
  where (origin = "Europe");
  panelby drivetrain;
  loess x=weight y=mpg_highway;
run;
```

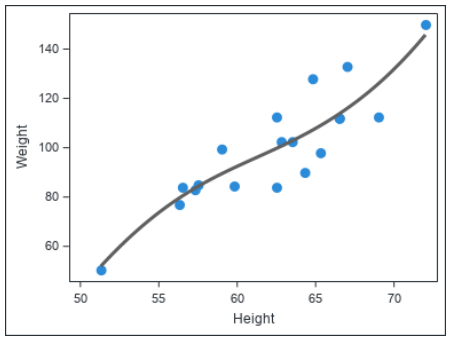
See Also

- "LOESS Statement" on page 451 (SGPANEL procedure)
- "LOESS Statement" on page 1195 (SGPLOT procedure)

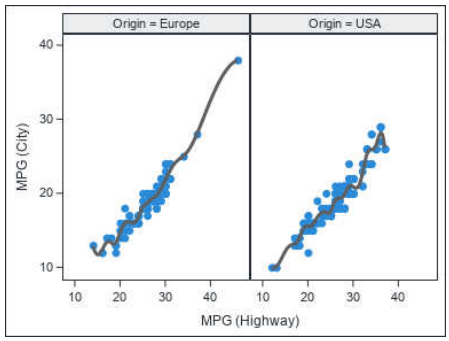
About Penalized B-Spline Plots

A penalized B-spline curve includes a scatter plot of two numeric variables along with an overlaid nonlinear fit line. You can specify the degree of the local polynomials to use for each local regression. You can also change the default smoothing technique that is applied to the fit.

The first example below shows the relationship of height to weight and the line of best fit for a class of students. The second example, which uses the SGPANEL procedure, shows the city and highway mileage for vehicles, classified by region of origin. In both examples, the automatically generated legend for the fit line is not needed and has been suppressed.




```
proc sgplot data=sashelp.class
     noautolegend;
  pbspline x=height y=weight;
run;
```




```
proc sgpanel data=sashelp.cars
       (where=(origin in('USA', 'Europe')))
       noautolegend;
  panelby origin;
  pbspline x=mpg_highway y=mpg_city;
run;
```

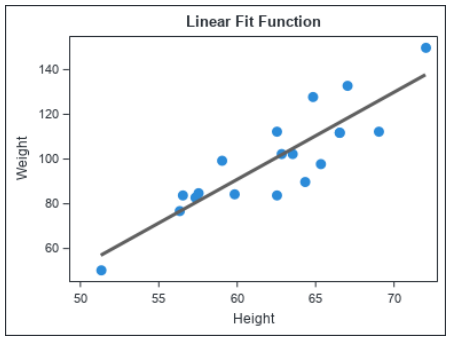
See Also

- "PBSPLINE Statement" on page 476 (SGPANEL procedure)
- "PBSPLINE Statement" on page 1221 (SGPLOT procedure)

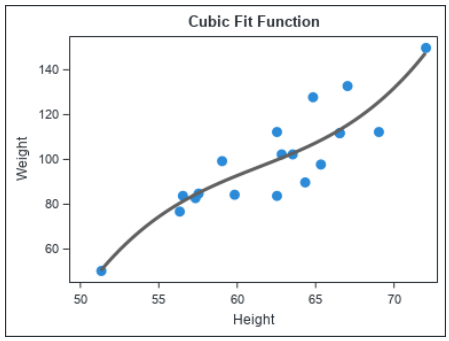
About Regression Plots

A regression plot includes a scatter plot of two numeric variables along with an overlaid linear or nonlinear fit line that enables you to perform a regression analysis. You can specify one of three types of regression equation: linear, quadratic, or cubic. You can display confidence limits for mean predicted values or individual predicted values.

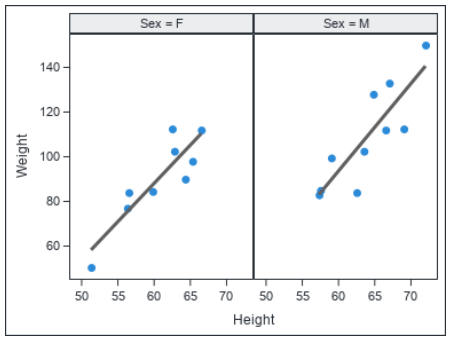
The following examples show the relationship of height to weight and the line of best fit for a class of students. Examples are provided for the SGPLOT and SGPANEL procedures. The first two examples show the same plot with a linear fit line and a cubic fit line, respectively. The third example shows a paneled graph. In all three examples, the automatically generated legend for the fit line is not needed and has been suppressed.




```
title "Linear Fit Function";
proc sgplot data=sashelp.class
    noautolegend;
  reg x=height y=weight;
run;
title;
```




```
title "Cubic Fit Function";
proc sgplot data=sashelp.class
    noautolegend;
  reg x=height y=weight / degree=3;
run;
title;
```




```
proc sgpanel data=sashelp.class
    noautolegend;
  panelby sex;
  reg x=height y=weight;
run;
```

See Also

- “REG Statement” on page 520 (SGPANEL procedure)
- “REG Statement” on page 1267 (SGPLOT procedure)

Distribution Plots

About Distribution Plots

You can use the SGPLOT and SGPANEL procedures to produce plots that characterize the frequency or the distribution of your data.

The plot statements include many options for controlling how the output is displayed. The options that are available depend on the plot type. The following sections describe each plot and the options that are available.

The distribution plots are described in the following sections. If you run the examples, your output might differ somewhat depending on the size of your graphics. The examples here were specified to be a particular size, usually 4 inches or 4.5 inches wide. You can specify the size in the ODS GRAPHICS statement. See "WIDTH=*dimension*" on page 123.

See Also

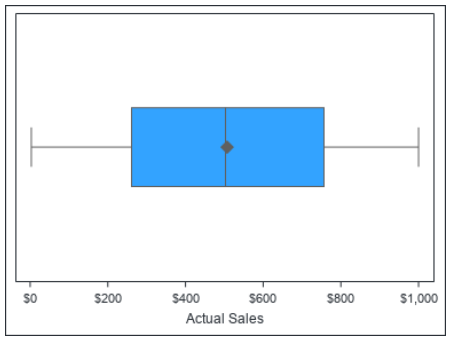
"Plot Type Compatibility" on page 1779

About Box Plots

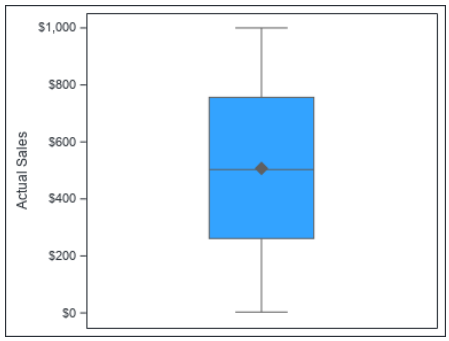
A box plot summarizes the data and indicates the median, upper and lower quartiles, and minimum and maximum values. The plot provides a quick visual summary that easily shows center, spread, range, and any outliers. The SGPLOT and SGPANEL procedures have separate statements for creating horizontal and vertical box plots.

The following examples show product sales summaries. Examples are provided for the SGPLOT and SGPANEL procedures.

The following two examples use the SGPLOT procedure to create a horizontal box plot and a vertical plot, respectively.

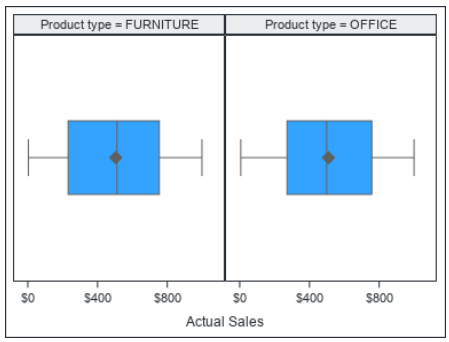



```
proc sgplot data=sashelp.prdsale;
  hbox actual;
run;
```

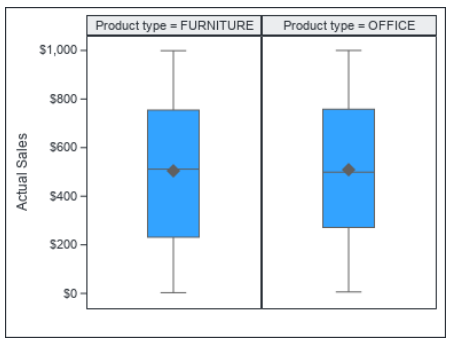



```
proc sgplot data=sashelp.prdsale;
  vbox actual;
run;
```

The following two examples use the SGPANEL procedure to create a horizontal box plot and a vertical plot, respectively. The box plots are paneled by product type.




```
proc sgpanel data=sashelp.prdsale;
  panelby prodtype;
  hbox actual;
run;
```




```
proc sgpanel data=sashelp.prdsale;
  panelby prodtype;
  vbox actual;
run;
```

Options are available that enable you to customize the box plot and enhance its appearance. For example, you can do the following:

- control the box width, the whisker cap shape, and the visual attributes for the mean marker, median line, and the connect lines. You can also hide the whisker caps, mean marker, median line, and the outliers.
- specify data labels and font attributes for the labels.
- specify the method to use for computing the percentiles for the plot.
- group the data by the values of a variable. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.
- control the display of grouped boxes. For example, you can specify whether the boxes are overlaid or clustered, and the width of each cluster.
- specify an amount to offset graph elements from the category midpoints or from the discrete axis tick marks.
- specify legend labels and plot transparency.
- assign the analysis variable to the secondary axis (X2 or Y2). This option is available only for the SGPLOT procedure.
- specify the value of an ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

Note: This list does not include all available options.

See Also

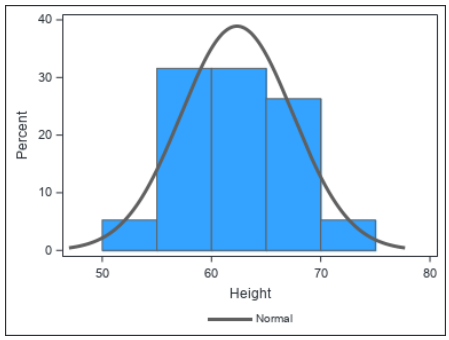
- "HBOX Statement" on page 343 (SGPANEL procedure)
- "VBOX Statement" on page 704 (SGPANEL procedure)
- "HBOX Statement" on page 1077 (SGPLOT procedure)
- "VBOX Statement" on page 1462 (SGPLOT procedure)

About Density Plots

After creating a histogram, you might use a density plot to fit various distributions to the data. The most common density plot uses the normal distribution, which is defined by the mean and the standard deviation.

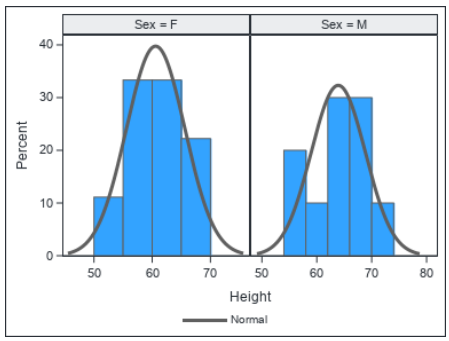
A density plot can be used by itself, combined with another density plot, and overlaid on a histogram.

The following examples show a density plot overlaid on a histogram. Examples are provided for the SGPLOT and SGPANEL procedures.




```
proc sgplot data=sashelp.class;
  histogram height;
  density height;
run;
```

The SGPANEL example shows output that is paneled by gender. The UNISCALE= ROW option specifies that only the shared row axes are identical. The column axes vary based on the values of the height for the respective genders.




```
proc sgpanel data=sashelp.class;
 panelby sex /
    uniscale=row;
 histogram height;
 density height;
run;
```

Options are available that enable you to customize the density plot and enhance its appearance. For example, you can do the following:

- control the visual attributes of the density line.
- specify a kernel distribution instead of normal. You can also specify the scaling that is used for the response axis.
- specify legend labels and plot transparency.

Note: This list does not include all available options.

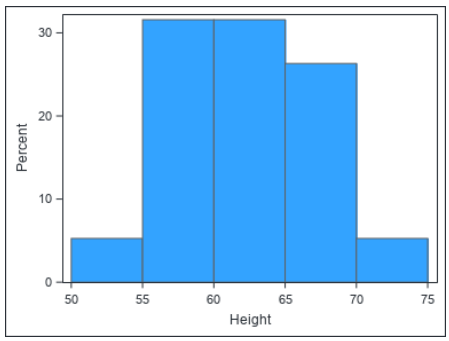
See Also

- “DENSITY Statement” on page 221 (SGPANEL procedure)
- “DENSITY Statement” on page 946 (SGPLOT procedure)

About Histograms

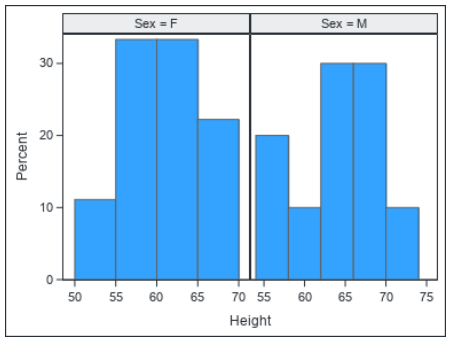
Histograms consist of a series of columns representing the frequency of a variable over a discrete interval or class.

The following examples show the height distribution for a class of students. Examples are provided for the SGPLOT and SGPANEL procedures.




```
proc sgplot data=sashelp.class;
  histogram height;
run;
```

The SGPANEL example shows output that is paneled by gender. The UNISCALE= ROW option ensures that only the shared row axes are identical. The column axes vary based on the values of the height for the respective genders.




```
proc sgpanel data=sashelp.class;
 panelby sex /
    uniscale=row;
 histogram height;
run;
```

Options are available that enable you to customize the histogram and enhance its appearance. For example, you can do the following:

- control the visual attributes of the bins, such as fill color and outlines.
- specify the number of bins, their width, and the X coordinate of the first bin.
- specify legend labels and plot transparency.
- assign the response variable and the calculated values to the secondary axis (X2 or Y2). This option is available only for the SGPLOT procedure.

Note: This list does not include all available options.

See Also

- "HISTOGRAM Statement" on page 397 (SGPANEL procedure)
- "HISTOGRAM Statement" on page 1136 (SGPLOT procedure)

Categorization Plots and Charts

About Categorization Plots and Charts

Categorization plots and charts produce a series of graph elements, one for each selected category of cases. For example, the relation between the age and the risk of a heart attack might differ between males and females. Categorization plots and charts can reveal patterns, complex interactions, exceptions, and anomalies.

You can use the SGPLOT and SGPANEL procedures to produce a variety of categorization plots and charts. The plot and chart statements include many options for controlling how the output is displayed. The options that are available depend on the plot type. The following sections describe each type and the options that are available.

The categorization plots are described in the following sections. If you run the examples, your output might differ somewhat depending on the size of your graphics. The examples here were specified to be a particular size, usually 4 inches or 4.5 inches wide. You can specify the size in the ODS GRAPHICS statement. See "WIDTH=*dimension*" on page 123.

See Also

"Plot Type Compatibility" on page 1779

About Bar Charts

Overview of Standard and Parameterized Bar Charts

Bar charts use bars to represent statistics based on the values of a category variable. Bar charts are useful for displaying magnitudes and emphasizing differences.

You can use the SGPLOT and SGPANEL procedures to create the following:

- horizontal and vertical bar charts that summarize the values of a category variable. Use the HBAR and VBAR statements for the charts. These charts are compatible only with other categorization charts and plots. A response variable is optional. If not specified, the summary is the category variable frequency count.
- basic-compatible horizontal and vertical bar charts that summarize the values of a category variable. Use the HBARBASIC and VBARBASIC statements for the charts. These bar charts are compatible with other categorization charts as well as basic plots, such as scatter and series plots, and box plots. A response variable is optional. If not specified, the summary is the category variable frequency count.
- parameterized horizontal and vertical bar charts that require a response variable in addition to the category variable. The response variable contains pre-summarized computed values such as a sum or a mean for each unique value of the category variable. Use the HBARPARM and VBARPARM statements for the charts. All of the summarizations that are performed by the HBAR, VBAR, HBARBASIC, and VBARBASIC statements can be used. Use the HBARPARM and VBARPARM statements with the presummarized data for the charts. These bar charts are compatible with other basic charts and plots.

Options are available that enable you to customize both types of bar charts and enhance their appearance. For example, you can do the following:

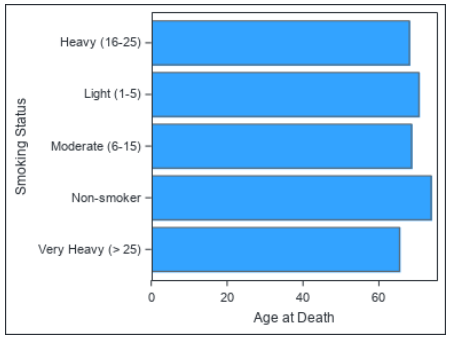
- control the visual attributes of the bars, such as bar width, fill color, fill skin, and outlines.
- add data labels and specify font attributes for the labels.
- group the data by the values of a variable. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.
- control the display of grouped bars. For example, you can specify the width of each cluster.
- specify an amount to offset graph elements from the category midpoints or from the discrete axis tick marks.

- specify legend labels, plot transparency, and URLs for web pages to be displayed when parts of the plot are selected within an HTML page.
- specify the value of an ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

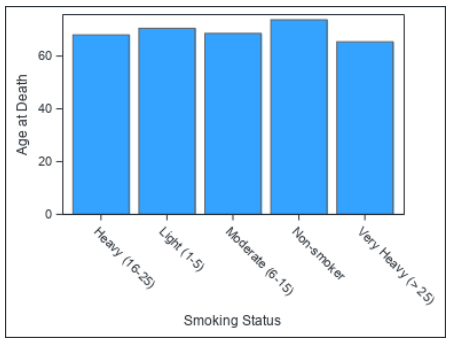
Note: This list does not include all available options.

Bar Chart Examples

The following examples show statistics for different categories of smokers. The examples use the SGPLOT procedure to create a horizontal bar chart and a vertical bar chart, respectively. By default, the charts show the frequency for each category. The examples specify an optional response variable to show the average age at death for each category rather than the frequency.

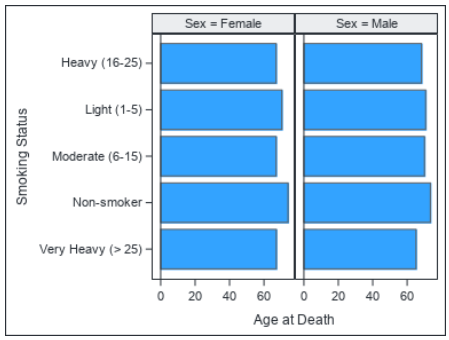



```
proc sgplot data=sashelp.heart;
  hbar smoking_status /
    response=ageatdeath
    stat=mean;
run;
```

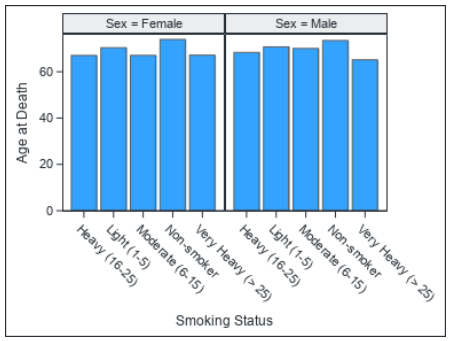



```
proc sgplot data=sashelp.heart;
  vbar smoking_status /
    response=ageatdeath
    stat=mean;
run;
```

The following two examples use the SGPANEL procedure to create a horizontal chart and a vertical chart, respectively. The bar charts are paneled by gender.




```
proc sgpanel data=sashelp.heart;
  panelby sex;
  hbar smoking_status /
    response=ageatdeath
    stat=mean;
run;
```




```
proc sgpanel data=sashelp.heart;
  panelby sex;
  vbar smoking_status /
    response=ageatdeath
    stat=mean;
run;
```

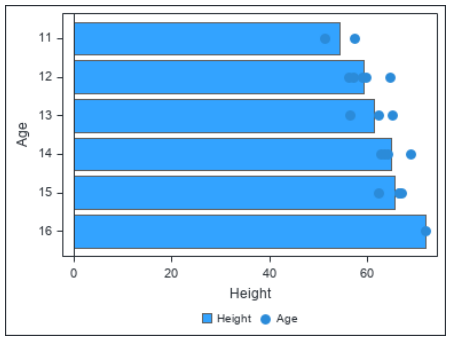
Bar charts includes options that are not applicable to parameterized bar charts. For example, you can do the following:

- specify the response variable and the statistic to use for its axis
- specify the order in which the response values are arranged
- show limit lines, specify the statistic to use for the limit lines, and specify the confidence level
- for grouped data, you can specify whether the bars are stacked or clustered
- specify how many times observations are repeated for computational purposes

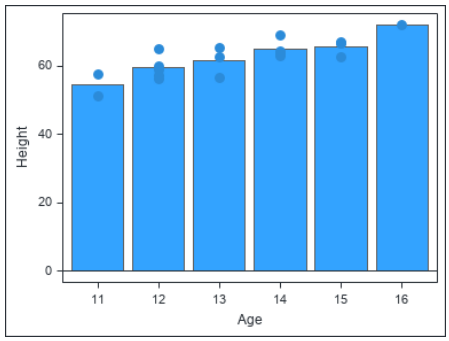
Note: This list does not include all available options.

Basic-Compatible Bar Chart Examples

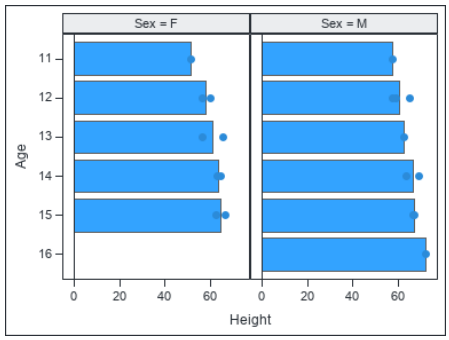
The following examples combine a bar chart with a scatter plot to show the age and height of students in a class. The examples specify an optional response variable and statistic to show the height average rather than the frequency.



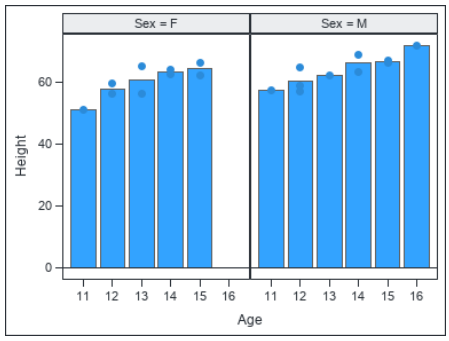

```
proc sgplot data=sashelp.class;
  hbarbasic age /
    response=height stat=mean;
  scatter x=height y=age;
run;
```




```
proc sgplot data=sashelp.class noautolegend;
  vbarbasic age /
    response=height stat=mean;
  scatter x=age y=height;
run;
```




```
proc sgpanel data=sashelp.class
     noautolegend;
  panelby sex;
  hbarbasic age /
    response=height stat=mean;
  scatter x=height y=age;
run;
```




```
proc sgpanel data=sashelp.class
     noautolegend;
  panelby sex;
  vbarbasic age /
    response=height stat=mean;
  scatter x=age y=height;
run;
```

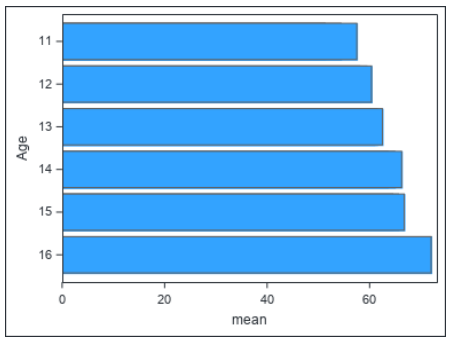
Parameterized Bar Chart Examples

Note: Parameterized bar charts are considered basic charts rather than categorization charts. However, all bar charts are described here in one place.

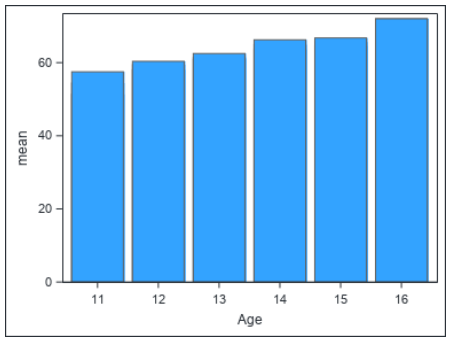
The following examples show height averages for a class of students. The averages are obtained using the following program.

```
proc means data=sashelp.class alpha=.05 clm mean std;
  class age sex;
  var height;
  output out=classMean uclm=uclm lclm=lclm mean=mean;
run;
```

The following two examples use the SGPLOT procedure to create a horizontal chart and a vertical chart, respectively. The response variable contains the computed mean values that were created with the MEANS procedure.

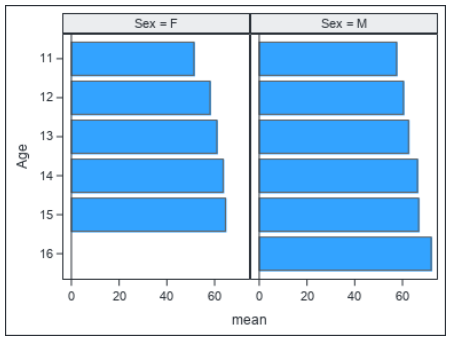



```
proc sgplot data=classMean;
  hbarparm category=age response=mean;
run;
```

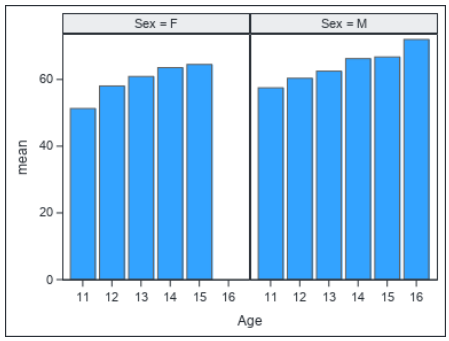



```
proc sgplot data=classMean;
  vbarparm category=age response=mean;
run;
```

The following two examples use the SGPANEL procedure to create horizontal and vertical bar charts, respectively. The charts are paneled by gender.

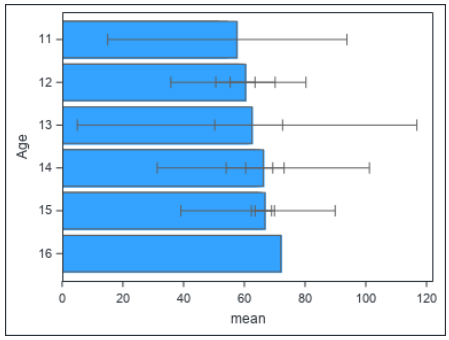



```
proc sgpanel data=classMean;
  panelby sex;
  hbarparm category=age response=mean;
run;
```




```
proc sgpanel data=classMean;
  panelby sex;
  vbarparm category=age response=mean;
run;
```

You can also assign variables to the upper and lower limits of the bar chart. Parameterized bar charts enable you to pass in your own precomputed limits.




```
proc sgplot data=classMean;
  hbarparm category=age response=mean /
    limitlower=lclm
    limitupper=uclm;
run;
```

Bar Chart Statement Option Applicability

There are several statement options that are common to all of the bar chart statements. However, there are options that apply only to a subset of the bar chart statements. If you want to try a different bar chart statement in your SAS program, you must know which options in your current bar chart statement apply to the new statement and which options are available for you to use in the new statement. The following table lists the bar chart statement options that are not common to all of the bar chart statements. For each option, an X indicates the bar chart statements to which it applies.

Table 2.1 *Statement Options That Are Not Common to All Bar Chart Statements*

Option	HBAR	HBARBASIC	HBARPARM	VBAR	VBARBASIC	VBARPARM
ALPHA=	X			X		
CATEGORYORDER=	X			X		
COLORSTAT=	X	X		X	X	
DATALABEL=	X		X	X		X
DATALABELFORMAT=		X			X	
DATALABELPOS=	X		X	X		X
FREQ=	X			X		
LIMITATTRS=	X		X	X		X
LIMITCAPSCALE=	X		X	X		X
LIMITS=	X			X		
LIMITLOWER=			X			X
LIMITUPPER=			X			X
LIMITSTAT=	X			X		
NOSTATLABEL	X			X		
NUMSTD=	X			X		
SPLITCHAR=				X		X
SPLITCHARNODROP				X		X
STAT=	X	X		X	X	
STATLABEL	X			X		
WEIGHT=	X			X		

See Also

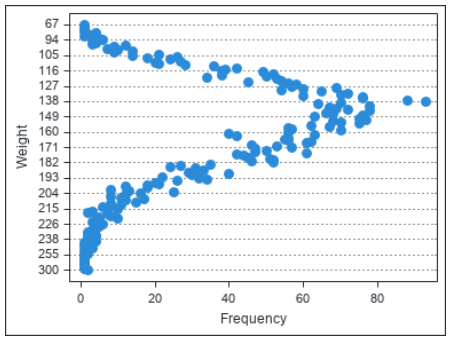
- "Plot Type Compatibility" on page 1779
- "HBAR Statement" on page 276 (SGPANEL procedure)

- "VBAR Statement" on page 634 (SGPANEL procedure)
- "HBAR Statement" on page 1008 (SGPLOT procedure)
- "VBAR Statement" on page 1389 (SGPLOT procedure)
- "HBARBASIC Statement" on page 1034 (SGPLOT procedure)
- "VBARBASIC Statement" on page 1418 (SGPLOT procedure)
- "HBARPARM Statement" on page 322 (SGPANEL procedure)
- "VBARPARM Statement" on page 681 (SGPANEL procedure)
- "HBARPARM Statement" on page 1055 (SGPLOT procedure)
- "VBARPARM Statement" on page 1438 (SGPLOT procedure)

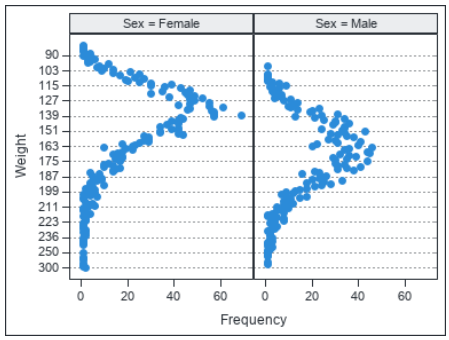
About Dot Plots

Dot plots summarize horizontally the values of a category variable. By default, each dot represents the frequency for each value of the category variable.

The following examples show the frequency of different weights of patients in a study. The examples use the SGPLOT and SGPANEL procedures.




```
proc sgplot  data=sashelp.heart;
  dot weight;
run;
```




```
proc sgpanel  data=sashelp.heart;
  panelby sex;
  dot weight;
run;
```

Options are available that enable you to customize the dot plot and enhance its appearance. For example, you can do the following:

- specify an optional response variable and show the mean, the sum, or the frequency for that variable. You can also specify the order in which the response values are arranged.
- show limits for the plot. You can also specify the statistic for the limit lines and visual attributes of the lines.
- specify the color, size, and symbol for the markers.
- add data labels and specify font attributes for the labels.
- control the display of grouped markers, lines, and bars. For example, you can specify whether the groups are overlaid or clustered, and the ordering of dots within a group.
- specify an amount to offset graph elements from the category midpoints or from the discrete axis tick marks.
- specify legend labels, plot transparency, and URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

See Also

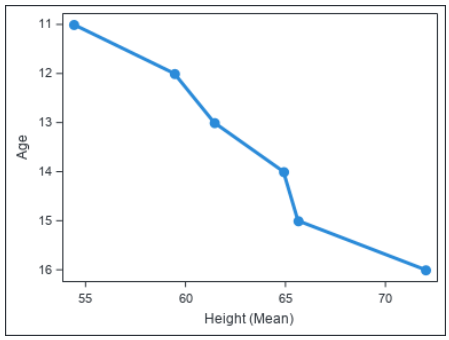
- "DOT Statement" on page 228 (SGPANEL procedure)
- "DOT Statement" on page 954 (SGPLOT procedure)

About Line Charts

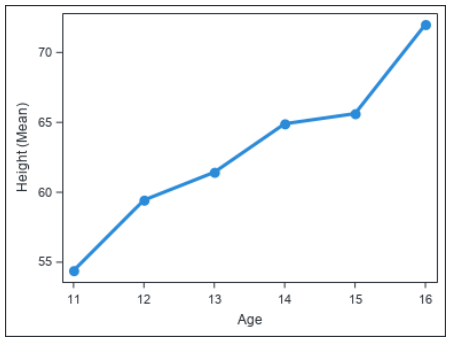
Line charts displays information as a series of data points connected by straight line segments. The SGPLOT and SGPANEL procedures have separate statements for creating horizontal and vertical line charts.

The following examples show mean weight values for a class. Examples are provided for the SGPLOT and SGPANEL procedures. The examples specify an optional response variable and use the mean statistic for that variable. The examples also add data point markers.

These two examples use the SGPLOT procedure to create a horizontal chart and a vertical chart, respectively.

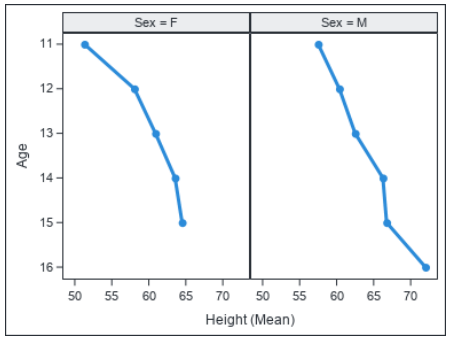



```
proc sgplot data=sashelp.class;
  hline age /
    response=height
    stat=mean
    markers;
run;
```

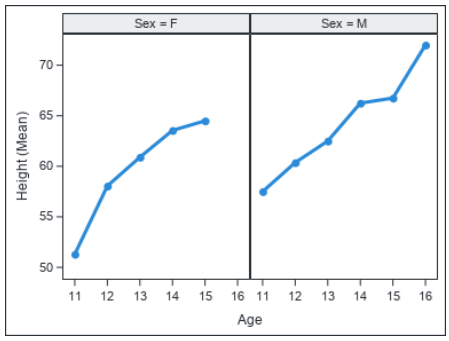



```
proc sgplot data=sashelp.class;
  vline age /
    response=height
    stat=mean
    markers;
run;
```

The following two examples use the SGPANEL procedure to create panels of horizontal and vertical charts, respectively.




```
proc sgpanel data=sashelp.class;
  panelby sex;
  hline age /
    response=height
    stat=mean
    markers;
run;
```




```
proc sgpanel data=sashelp.class;
  panelby sex;
  vline age /
    response=height
    stat=mean
    markers;
run;
```

Options are available that enable you to customize the line chart and enhance its appearance. For example, you can do the following:

- specify an optional response variable and show the mean, the sum, or the frequency for that variable. You can also specify the order in which the response values are arranged.
- show limits for the chart. You can also specify the statistic for the limit lines and visual attributes of the lines.
- add data point markers and specify the color, size, and symbol for the markers.
- add curve and data labels and specify font attributes for the labels.
- control the display of grouped lines. For example, you can specify whether the groups are overlaid or clustered, the width of each cluster, and the ordering of lines within a group.
- specify an amount to offset graph elements from the category midpoints or from the discrete axis tick marks.
- specify legend labels, plot transparency, and URLs for web pages to be displayed when parts of the plot are selected within an HTML page.
- assign the category variable, the response variable, or both variables to the secondary axis (X2 or Y2). This option is available only for the SGPLOT procedure.
- specify the value of an ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

Note: This list does not include all available options.

See Also

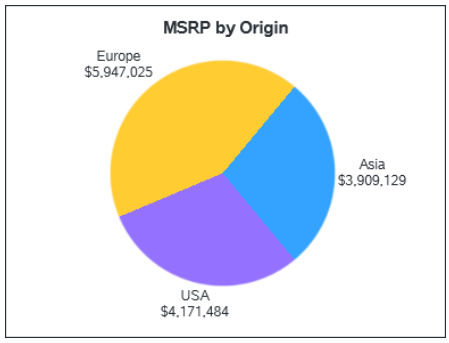
- "HLINE Statement" on page 407 (SGPANEL procedure)
- "VLINE Statement" on page 734 (SGPANEL procedure)
- "HLINE Statement" on page 1145 (SGPLOT procedure)
- "VLINE Statement" on page 1493 (SGPLOT procedure)

About Pie and Donut Charts (Preproduction)

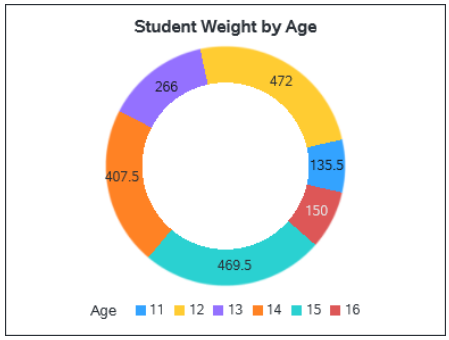
The SGPIE procedure produces pie charts and donut charts. The angles of pie and donut slices are used to graphically represent the value of a statistic for a data range. The charts are useful for examining how the values of a variable contribute to the whole and for quickly comparing the values of several variables.

Note: The SGPIE procedure is a preproduction feature, which means that it has not been fully developed, tested, or documented.

Table 2.2 *Pie and Donut Charts*



See the code for this example.



See the code for this example.

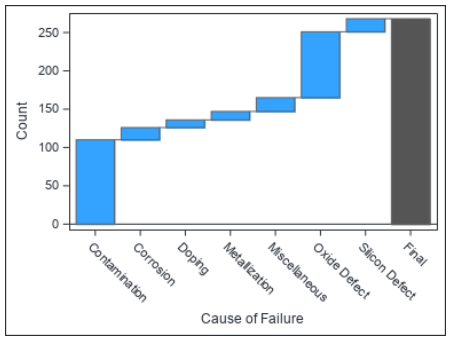
See Also

- "DONUT Statement" on page 843
- "PIE Statement" on page 859

About Waterfall Charts

Waterfall charts show how the value of a variable increases or decreases until it reaches a final value. In the chart, bars represent an initial value of Y and a series of intermediate values identified by X leading to a final value of Y. Waterfall charts are available only for the SGPLOT procedure.

The following example shows average failure counts for capacitors.




```
proc sgplot data=sashelp.failure;
   waterfall category=cause response=count;
run;
```

Options are available that enable you to customize the waterfall chart and enhance its appearance. For example, you can do the following:

- specify the statistic for the response variable.
- specify an initial bar for the chart. You can also specify the tick value that is used for the initial bar and visual attributes of the bar.
- control the appearance of the bars. For example, you can do the following:
  - show or hide the bar outline
  - show or hide the bar fill
  - use a special effect (data skin) for the fill
  - specify a variable to use for the bar colors
  - specify attributes separately for the final bar
- add data labels and specify font attributes for the labels.
- specify plot transparency and URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Note: This list does not include all available options.

See Also

"WATERFALL Statement" on page 1514 (SGPLOT procedure)

PART 2

Overview: Using the Procedures

Chapter 3
Concepts for Creating Graphs 69

Chapter 4
Elements of a Program 81

Chapter 5
About the Examples, Sample Data, and Additional Resources 87

3

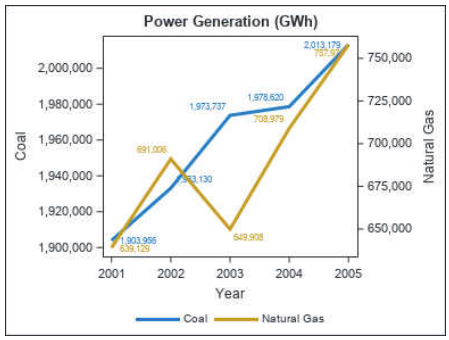
Concepts for Creating Graphs

Creating Single-Cell Graphs . . . 69
See Also . . . 71
Creating Multi-Cell Classification Panels . . . 71
Creating Paneled Scatter Plots . . . 72
Rendering Graphs from GTL Templates . . . 73
Producing Maps Using the SGMAP Procedure . . . 74
Plotting a Cloud Analytic Services (CAS) In-Memory Table . . . 75
Overview of Using Data That Is Processed in CAS . . . 75
User-Defined Formats . . . 75
Data Order . . . 76
Example: Preprocessing Data in CAS and Plotting the Results . . . 76
About ODS Destinations and Styles . . . 79
About ODS . . . 79
About ODS Destinations . . . 79
About ODS Styles . . . 79

Creating Single-Cell Graphs

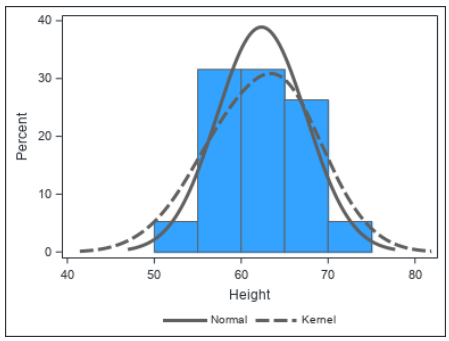
The SGPLOT procedure creates single-cell graphs with a wide range of plot types including density, dot, needle, series, bar, histograms, box, and others. The procedure can compute and display loess fits, polynomial fits, penalized B-spline fits, and ellipses. You can also add text, legends, and reference lines. Options are available for specifying colors, marker symbols, and other attributes of plot features. You can customize the axes by using axis statements such as XAXIS and YAXIS.

Plot statements can be combined to create more informative graphs. The following example shows two series plots that are overlaid in a single graph. Each plot is assigned to a different vertical axis. Data labels have been added for easy reference.




```
title "Power Generation (GWh)";
proc sgplot data=sashelp.electric(where=
  (year >= 2001 and customer="Residential"));
  xaxis type=discrete;
  series x=year y=coal / datalabel;
  series x=year y=naturalgas /
      datalabel y2axis;
run;
title;
```

The following example creates a graph with a histogram, a normal density curve, and a kernel density curve.

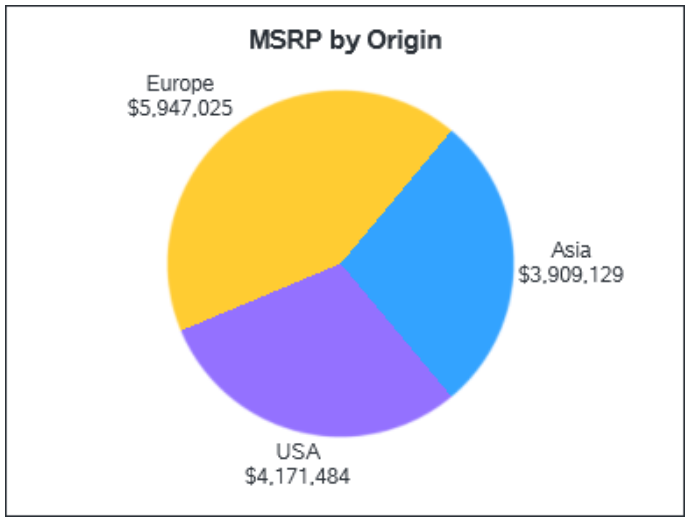



```
proc sgplot data=sashelp.class;
  histogram height;
  density height;
  density height / type=kernel;
run;
```

You can use the SGPIE procedure to create pie and donut charts. This procedure is considered preproduction.

Here is a pie chart:

Figure 3.1 *Pie Chart*



See Also

- Chapter 10, “SGPLOT Procedure,” on page 883
- Chapter 9, “SGPIE Procedure,” on page 839

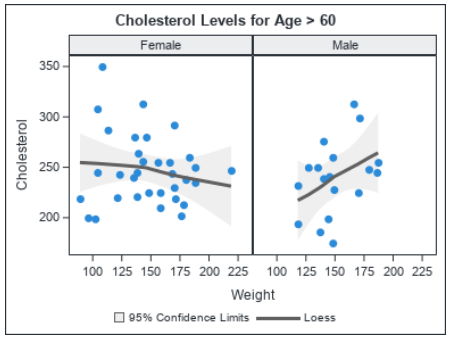
Creating Multi-Cell Classification Panels

The SGPANEL procedure creates a panel for the values of one or more classification variables. Each graph cell in the panel can contain either a single plot or multiple overlaid plots.

The SGPANEL procedure supports most of the plots and overlays that the SGPLOT procedure supports. For this reason, the two procedures have an almost identical syntax. As with the SGPLOT procedure, options are available for specifying colors, marker symbols, and other attributes.

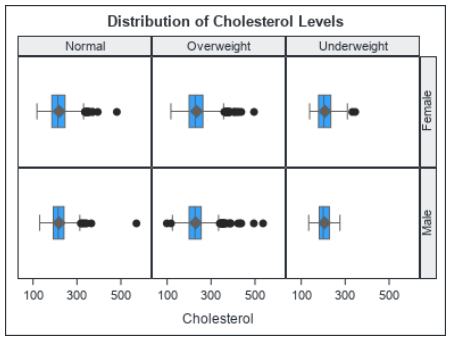
The procedure syntax supports four types of panel layouts: PANEL, LATTICE, COLUMNLATTICE, and ROWLATTICE.

The following example creates a panel of loess curves using the default PANEL layout. In the PANEL layout, each graph cell represents a specific crossing of values for one or more classification variables. A label above each cell identifies the crossing of values that is represented in the cell. By default, cells are created only for crossings that are represented in the data set.




```
title1 "Cholesterol Levels for Age > 60";
proc sgpanel data=sashelp.heart(
    where=(AgeAtStart > 60)) ;
  panelby sex / novarname;
  loess x=weight y=cholesterol / clm;
run;
title1;
```

The following example creates a panel of box plots in a LATTICE layout. The graph cells are arranged in rows and columns by using the values of two classification variables. Labels above each column and to the right of each row identify the classification value that is represented by that row or column. A cell is created for each crossing of classification values.




```
title1 "Distribution of Cholesterol Levels";
proc sgpanel data=sashelp.heart;
  panelby weight_status sex / layout=lattice
                              novarname;
  hbox cholesterol;
run;
title1;
```

For more information about the SGPANEL procedure and the procedure syntax, see Chapter 8, “SGPANEL Procedure,” on page 155.

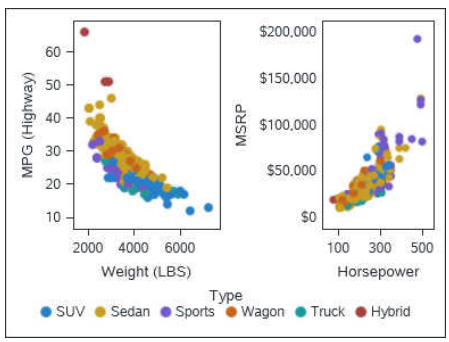
Creating Paneled Scatter Plots

The SGSCATTER procedure creates a paneled graph for multiple combinations of variables.

The procedure syntax supports the following features:

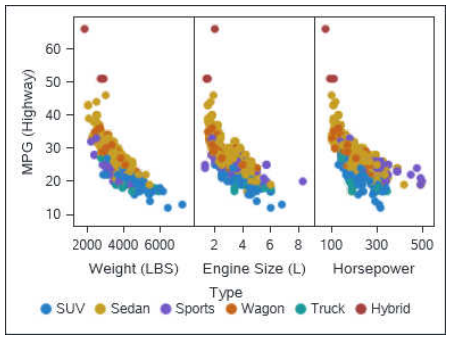
- three types of graph layouts: PLOT, COMPARE, and MATRIX
- basic scatter plots
- fit and confidence plots: loess curves, regression curves, penalized B-spline curves, and ellipses
- distribution plots: histograms and density curves (in the diagonal cells of a matrix)
- legends

The following example creates a panel using the PLOT layout. The PLOT statement creates a paneled graph of scatter plots where each cell has its own independent set of axes.



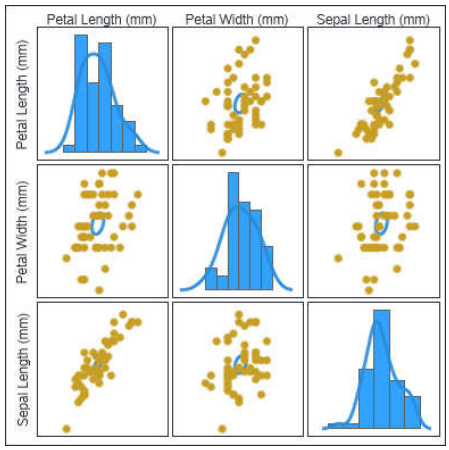

```
proc sgscatter data=sashelp.cars;
  plot mpg_highway*weight msrp*horsepower
          / group=type;
run;
```

The following example creates a panel using the COMPARE layout. The COMPARE statement creates a paneled graph that uses common axes for each row and column of cells. Cells are created for all crossing of the X and Y variables.




```
proc sgscatter data=sashelp.cars;
  compare y=mpg_highway
          x=(weight enginesize horsepower )
          / group=type;
run;
```

The following example creates a panel using the MATRIX layout. The MATRIX statement creates a matrix of scatter plots where each cell represents a different combination of variables. In the diagonal cells, you can place labels or histograms with or without density curves.



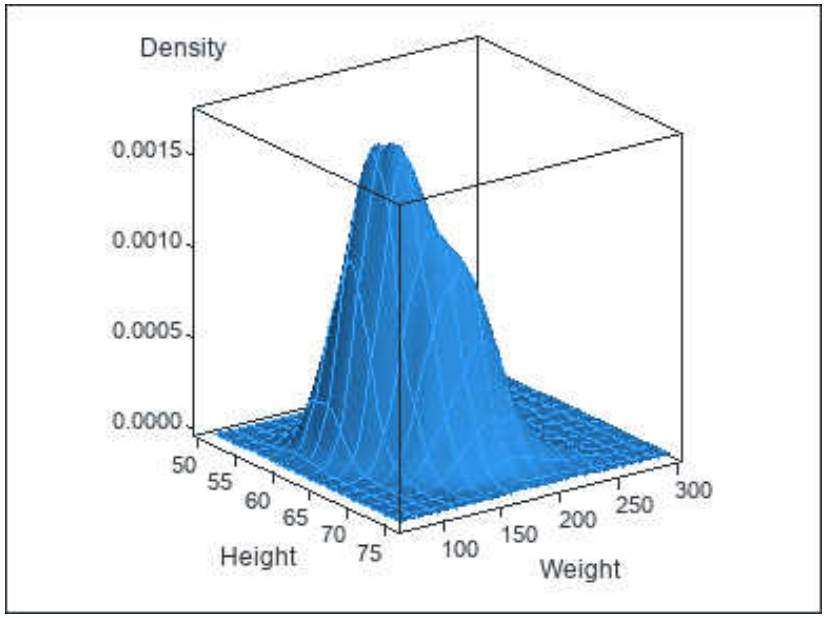

```
proc sgscatter data=sashelp.iris
               (where=(species eq "Virginica"));
matrix petallength petalwidth sepallength /
       markerattrs=GraphData2
       ellipse=(type=mean)
       diagonal=(histogram kernel);
run;
```

For more information about the SGSCATTER procedure and the procedure syntax, see Chapter 12, “SGSCATTER Procedure,” on page 1699.

Rendering Graphs from GTL Templates

The SGRENDER procedure creates graphical output from templates that are created using the Graph Template Language (GTL). You can use the GTL to create many different types of plots, paneled graphs, and matrices, some of which cannot be created with the ODS Graphics procedures.

The following example shows a layout that you can create by using the TEMPLATE procedure and the SGRENDER procedure.




```
proc template;
  define statgraph surface;
  begingraph;
     layout overlay3d;
      surfaceplotparm x=height y=weight z=density;
     endlayout;
  endgraph;
  end;
run;

proc sgrender data=sashelp.gridded template=surface;
run;
```

For more information about the SGRENDER procedure, see Chapter 11, "SGRENDER Procedure," on page 1677. For more information about the GTL, see *SAS Graph Template Language: User's Guide*.

Producing Maps Using the SGMAP Procedure

The SGMAP procedure provides concise syntax for creating maps. You can also overlay plots such as scatter plots, bubble plots, and text plots onto the map. The procedure uses ODS Graphics to produce output. ODS Graphics provides the power and flexibility to create many types of complex and appealing maps.

This document does not describe the SGMAP procedure. For complete documentation about the SGMAP procedure, see *SAS/GRAPH and Base SAS: Mapping Reference*.

Plotting a Cloud Analytic Services (CAS) In-Memory Table

Overview of Using Data That Is Processed in CAS

When data resides in the Cloud Analytic Services (CAS) environment, the data can be accessed by the ODS Graphics procedures through the CAS LIBNAME engine. The CAS LIBNAME engine enables the procedures to access and download CAS tables to the SAS client, such as SAS Data and AI Studio (known as SAS Studio prior to 2026.06). For more information about creating a CAS libref, see *SAS Cloud Analytic Services: User's Guide*.

For this release of SAS Viya, the analytics for the ODS Graphics procedures are performed in the SAS client instead of the CAS environment. Therefore, for large data situations, it is recommended that you summarize or reduce the data before attempting to graph it. For example, in the CAS procedure, you can use the AGGREGATE action to summarize your large data to another CAS table. Then, you can reference the summarized CAS data from the SGPLOT or SGPANEL procedures to create a bar chart using the HBARPARM or VBARPARM statements.

User-Defined Formats

When using user-defined formats, be aware that a format created in the FORMAT procedure is not automatically made available to the CAS environment. To make sure the format is available to both the CAS environment and the SAS client, use the CASFMTLIB option in the FORMAT procedure to export the format to the CAS environment. With this option, the format is also written locally in the SAS client environment. The CASFMTLIB option enables the format used for the summarization process in CAS to match the format used for the graph rendering.

See Also

SAS Cloud Analytic Services: User-Defined Formats

Data Order

The ODS Graphics procedures use data order in many instances to display your data. For example, when you create a graph that contains grouped data, the group attribute assignment of the GraphData1 ... GraphData*n* style elements normally uses data order.

Data order in CAS tables can be unpredictable. Graphs generated using SAS local data are consistent across multiple runs. Graphs generated using CAS data might vary across multiple runs. Because of this unpredictability, when the procedures use CAS data, the group attribute assignment of GraphData1 ... GraphData*n* style elements is changed from data order to ascending order. For character data, the assignment is based on an ascending formatted order. For numeric data, the assignment is based on an unformatted ordinal order.

This sorting affects the following:

- style attributes
- group order
- order for the GroupDisplay= option
- discrete axes

The connect order for series, band, step, and line plots uses the axis order. The connect order specifies how to connect the data points to form the step or line.

Note: If you use a discrete attribute map, the group assignment mentioned above has no effect.

In addition, the following options that support DATA order do not support that value when using CAS data:

- SORT= option in the PANELBY statement (SGPANEL procedure)
- GROUPORDER= option in several statements including those for categorization plots, box plots, high-low plots, and others
- SORTORDER= option in the KEYLEGEND statement
- DISCRETEORDER option on the axis statement

Example: Preprocessing Data in CAS and Plotting the Results

This example shows how to summarize an in-memory table in CAS and how to graph the result using the SGPLOT procedure. For very large in-memory tables, it is recommended that you preprocess the data in CAS to reduce the size of the data

before you graph it. The example uses the data in SAS data set Sashelp.Cars as sample data.

The MDSUMMARY procedure is used to summarize the data before it is graphed. The MDSUMMARY procedure generates descriptive statistics of numeric variables such as the sample mean, sample variance, sample size, sum of squares, and so on. The procedure is supported on the CAS server and can process a very large in-memory table. For more information about the MDSUMMARY procedure, see "MDSUMMARY Procedure" in *SAS Cloud Analytic Services: User's Guide*.

Here is the SAS code for this example.

```
*options cashost="cloud.example.com" casport=5560; /* 1 */
cas casauto sessopts=(caslib=casuser);             /* 2 */

libname mycas cas;                                 /* 3 */

proc casutil;                                      /* 4 */
  load data=sashelp.cars casout="cars";
quit;

proc mdsummary data=mycas.cars;                    /* 5 */
  var mpg_highway;
  groupby type origin;
  output out=mycas.cars_summary;
run;

proc print data=mycas.cars_summary;                /* 6 */
  var origin type _mean_;
run;


title "Summarized Highway MPG";                    /* 7 */

proc sgplot data=mycas.cars_summary;               /* 8 */
  format _mean_ 2.;
  vbarparm category=type response=_mean_
     / group=origin;
  yaxis label="Summary MPG Values";
run;
title;

cas casauto terminate;
```

1 If necessary, set system options CASHOST= and CASPORT= to the host name and port for your CAS server, and then uncomment the OPTIONS statement.

2 Connect to your CAS server. Specify caslib CASUSER as the active caslib.

3 Create a CAS engine libref to the active caslib, CASUSER.

4 Load the SAS data set Sashelp.Cars into the in-memory table Cars.

5 Use the MDSUMMARY procedure to summarize variable Mpg_Highway, and group the results by variables Type and Origin. Store the results in the in-memory table Cars_Summary in caslib CASUSER. The table includes a row of summary statistics for each unique combination of origin and type.

6 Optional: Print columns Type, Origin, and _Mean_ in table Cars_Summary.

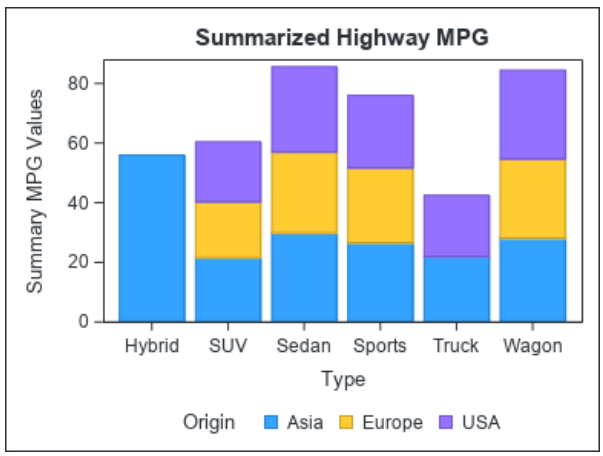
7 Optional: Specify a title for the graph.

8 Use the SGPLOT procedure to generate the bar chart using data Mycas.Cars_Summary. The procedure creates a parameterized vertical bar chart that shows the mean statistic for highway miles-per-gallon.

The summarized data generated by the MDSUMMARY procedures contains 15 observations as shown here:

Obs	Origin	Type	_Mean_
1	Europe	Sedan	27.115384615
2	Europe	SUV	18.7
3	USA	Truck	20.5
4	Asia	Sedan	29.968085106
5	Asia	Truck	22
6	Europe	Sports	25.130434783
7	Asia	SUV	21.68
8	USA	Sedan	28.544444444
9	Asia	Wagon	28.181818182
10	Asia	Hybrid	56
11	Asia	Sports	26.647058824
12	USA	SUV	20.04
13	USA	Sports	24.222222222
14	Europe	Wagon	26.583333333
15	USA	Wagon	29.714285714

Here is the resulting bar chart:



The group values and the category axis tick values are automatically sorted in ascending order by default when an in-memory table is used.

For more information about Cloud Analytic Services programming in SAS, see *SAS Viya Platform Programming: Getting Started*.

About ODS Destinations and Styles

About ODS

Base SAS Output Delivery System (ODS) manages all output created by the procedures and enables you to control the output destination and format. ODS also enables you to control the style and other output features.

About ODS Destinations

For the creation of ODS graphics, a valid ODS destination must be active. By default, your Integrated Development Environment (IDE) opens a default ODS destination, typically ODS HTML5. Refer to the documentation for your IDE for information about your program output. ODS can send your graphs to a number of different destinations. If you keep the default destination open and open another, the resultant output is sent to all open destinations.

For more information, see "Specifying ODS Destinations" on page 1884.

About ODS Styles

ODS styles determine the overall appearance of your output. By default, ODS applies a style to all output. A style is a template, or set of instructions, that determines the colors, fonts, line styles, fill colors, and other presentation aspects of your output. Each destination is associated with a default style. Your IDE might be configured to use a different ODS style with its default ODS destination. Refer to the documentation for your IDE for information about your program output.

The ODS Graphics procedures automatically obtain their default appearance attributes from the current ODS style. However, you can use appearance options in your plot statements to override the default style attributes.

For more information, see Chapter 15, "Controlling the Appearance of Your Graphs," on page 1839.

SAS ships predefined styles. Some of these predefined styles are described in "Recommended Styles" on page 1842. For help with finding the list of all supported styles, see "About the Default Styles" on page 1842.

4

Elements of a Program

A Typical Program 81
See Also 0
About the PROC Step 82
About the PROC Step 82
Procedure Statements and Options 83
Plot Statements and Options 83
(Optional) Legend Statement and Options 84
(Optional) Axis Statements and Options 84
Other Required Statements 84
About SAS Statements 85
About the ODS GRAPHICS Statement Options 85
See Also 86
About Using an Annotation Data Set 86
About Using an Attribute Map Data Set 86

A Typical Program

Your programs must include at least one procedure (PROC step), which in turn contains a number of statements related to the procedure. The programs can also include ODS statements, ODS GRAPHICS statements, and Base SAS statements. In addition, the programs can specify an annotation data set or an attribute map data set.

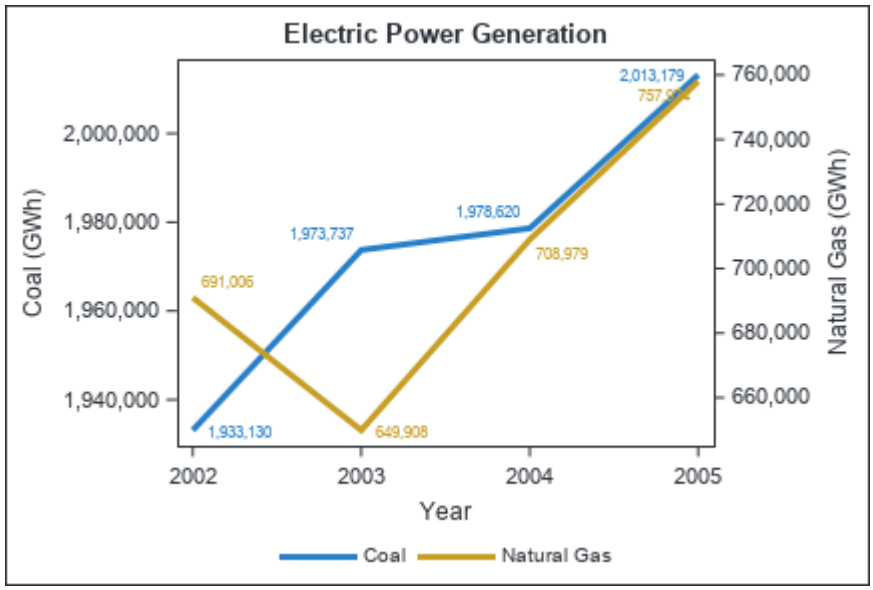
The sample program below identifies the basic elements of a typical program.

Example Code 4.1 *Sample Program*

```
/*  Global SAS Statements */
ods graphics on / width=4.5in;        /* 1 */
title "Electric Power Generation";    /* 2 */

/* Procedure Step  */
proc sgplot data=sashelp.electric;    /* 3 */
```

```
  where (year >= 2002) and
      (customer="Residential");       /* 4 */
  series x=year y=coal / datalabel;   /* 5 */
  series x=year y=naturalgas /
       datalabel y2axis;              /* 6 */
  xaxis type=discrete;
  yaxis label="Coal (GWh)";           /* 7 */
  y2axis label="Natural Gas (GWh)";
run;                                  /* 8 */

/*  Global SAS Statement */
title;                                /* 9 */
ods graphics on / reset=width;        /* 10 */
```

Here is the output for the sample program.



The following sections describe each element of the program in more detail and explain how the elements relate to one another.

About the PROC Step

About the PROC Step

Note: The descriptions in this and subsequent topics refer to the example that is shown in "A Typical Program" on page 81.

A group of SAS procedure statements is called a PROC step.

The PROC step consists of the following:

- a beginning procedure (PROC) 3 statement with options
- typically, statements specifying plot types, variables, and options
- an ending RUN 8 statement

These statements can identify and analyze the data in SAS data sets. They can generate the graphics output and control the appearance of the output. Statements can define variables and perform other operations on your data. You can also specify global statements and options within the PROC step.

Procedure Statements and Options

The procedure statement 3 identifies which procedure you are invoking, such as the SGPLOT procedure, the SGPANEL procedure, and so on.

The statement also specifies which input data set is to be used. A data set is not required for all of the procedures. For example, the SGRENDER procedure defaults to the most recently created SAS data set if none is provided.

The statement can include options that are related to the procedure. For example, the DESCRIPTION= option can be used with several procedures to provide a description for the output image.

Plot Statements and Options

Plot statements are used within the procedure to identify the type of plot that you want the procedure to produce. The SGPLOT and SGPANEL procedures require at least one plot statement.

Multiple plot statements can be used, as shown in the example. A SERIES statement 5 is used to create a series plot that shows power generation from coal. A second SERIES statement creates a series plot that shows power generation from natural gas.

Options are available for specifying colors, marker symbols, and other attributes of plot features. In the example, both series plots specify that data labels are displayed. The second SERIES statement uses the Y2AXIS option to plot natural gas power output along the Y axis on the right side of the plot 6.

The SGPLOT procedure also enables you to add a text inset to a plot using the INSET statement (not shown in the example). The INSET statement adds a text box within the axes of the plot. Options are available for specifying the visual attributes of the text box and the text.

(Optional) Legend Statement and Options

By default, legends are created automatically for some plots, depending on their content. The graph shown in the example has an automatically generated legend.

You can manually add a legend to the graph for the SGPLOT, SGPANEL, and SGSCATTER procedures. When you manually add a legend, options are available for specifying the legend title, its position in the graph, and other attributes.

(Optional) Axis Statements and Options

The SGPLOT and SGPANEL procedures contain statements that enable you to change the type and appearance for the axes of the graph. By default, the type of each axis is determined by the types of plots that use the axis and the data that is applied to the axis. You can change the type of axis that is used for a plot. For example, to display independent data values rather than a range of numeric values on the axis, specify the TYPE=DISCRETE option. (Not all plot types support discrete axes.)

The SGPLOT procedure supports the use of secondary axes, as shown in the example 6. Secondary axes are denoted as X2 and Y2 axes. The secondary axes support the same options as the primary axes.

When you use an axis statement, options are available for showing or hiding axis features, such as ticks and labels, and for specifying other attributes. The graph shown in "A Typical Program" on page 81 has three axis statements 7. The first statement changes the X axis to be discrete. The other two statements change the labels for the Y and Y2 axes.

Other Required Statements

The SGPANEL and SGSCATTER procedures are not shown in the example. These procedures include some important statements that are required within the procedure block.

- The SGPANEL procedure requires a PANELBY statement. This statement specifies one or more classification variables for the panel, the layout type, and other options for the panel. For more information, see "PANELBY Statement" on page 168.
- The SGSCATTER procedure requires one of these three statements:

  PLOT creates a paneled graph of scatter plots where each graph cell has its own independent set of axes.

COMPARE creates a shared axis panel, also called an MxN matrix.

MATRIX creates a scatter plot matrix.

For more information, see "Statements for Creating Panels" on page 1702.

About SAS Statements

The ODS Graphics procedures support a number of SAS statements. Some of these, such as the TITLE statement, are global statements that can be used anywhere in your program. Global statements apply to all the output generated subsequently in the program, unless they are overridden by a procedure option or another global statement.

As shown in "A Typical Program" on page 81, the TITLE statement is used toward the beginning and end of the program. The first TITLE statement 2 specifies the title. The second TITLE statement 9 cancels the current title.

The example program also uses a WHERE statement 4 to subset the data that is used in the graph. In the example, the WHERE statement selects observations based on their date (2002 or greater) and the type of customer (residential).

For more information, see "Global Statements" on page 93.

About the ODS GRAPHICS Statement Options

You can use the ODS GRAPHICS statement options to control many aspects of your graphics. The ODS GRAPHICS statement is a global statement that can be used anywhere in your program. The settings that you specify remain in effect for all graphics until you change or reset these settings with another ODS GRAPHICS statement.

As shown in "A Typical Program" on page 81, the ODS GRAPHICS statement is used at the beginning and end of the program to modify the size of the graph. The first program statement 1 specifies the graph size. The last program statement 10 sets all ODS Graphics options back to their defaults.

See Also

"ODS GRAPHICS Statement" on page 102

About Using an Annotation Data Set

The SG annotation feature enables you to add shapes, arrows, text, images, and other annotations to graph output.

Two main steps are required to add annotation elements to a graph:

1. Create an annotation data set, which contains the commands for creating the annotation elements.
2. Modify the procedure to use the annotation data set. You can use annotation in the SGPLOT, SGPANEL, and SGSCATTER procedures.

For more information, see Chapter 20, "Annotating ODS Graphics," on page 1935.

About Using an Attribute Map Data Set

The attribute map feature enables you to control the visual attributes that are applied to specific data values in your graphs. For example, if you create a graph that plots items sold in different countries, you can specify the display attributes for the sales data of each country by name.

Attribute maps enable you to assign the same graphical properties to specific values or ranges of values regardless of data order or the data range.

The ODS Graphics procedures support two types of attribute maps:

discrete attribute map
: maps discrete grouped data values to graphical properties.

range attribute map
: maps ranges of continuous numeric values to graphical properties.

For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

5

About the Examples, Sample Data, and Additional Resources

About the Examples in This Book 87
About the SASHELP Data Library 88
Examples and Resources on the Web 88

About the Examples in This Book

The example programs that are shown in this document often provide the code that you need to generate the graphs that are shown in the figures. We encourage you to copy and paste the example code into your SAS session and generate the graphs for yourself. The examples are written to be runnable. Unless otherwise noted, the examples use the Illuminate ODS style. By default, they are typically rendered as 640 pixel by 480 pixel images in the default HTML output.

Because of size limitations, the graphs in this document are typically not shown in their default size of 640 pixels by 480 pixels. They are scaled down to meet the size requirements of our documentation production system. When graphs are reduced in size, the smaller graphs might have scaled down font sizes. Also, their numeric axes might display the tick values differently. Thus, the graphs that you generate from the example programs might not always look identical to the graphs that are shown in the figures. However, both graphs accurately represent the data.

When you produce your own graphical output, you can change the graph size and attributes, if needed. You can use the ODS destination options and the ODS GRAPHICS statement options to control many aspects of your graph output. For more information, see Chapter 16, “Managing Your Graphics with ODS,” on page 1883.

About the SASHELP Data Library

Many examples process sample data contained in a SAS data set that is stored in the SASHELP library. Each example specifies the data set name.

Examples and Resources on the Web

The SAS website contains a large number of examples that can help you visualize and code your graphs. The examples cover a range of SAS technologies including the ODS Graphics procedures.

- Graphically Speaking is a blog focused on using ODS Graphics for data visualization in SAS. The blog covers topics related to the ODS Graphics procedures and the SAS Graph Template Language. The Graphicall Speaking blog is available at the following URL:

  https://blogs.sas.com/content/graphicallyspeaking/

- The SAS Training Post is a blog that provides tutorials, tips, and practical information about SAS. Dr. Robert Allison contributed a great deal of SAS/GRAPH and ODS Graphics data visualization examples to this blog during his tenure at SAS. His blog is no longer active, but it provides an extensive set of very useful examples. The SAS Training Post is available at the following URL:

  https://blogs.sas.com/content/sastraining/author/robertallison/

- The SAS Knowledge Base contains an abundance of searchable samples and SAS Notes. You can browse by topic, search for a particular note or a particular technology such as the name of a procedure, and conduct other searches. The SAS Knowledge Base is available at the following URL:

  https://support.sas.com/en/knowledge-base.html

  Note: The SAS Knowledge Base content is currently available only in English.

- The Graphics Samples Output Gallery in the SAS Knowledge Base is a collection of graphs organized by SAS procedure. The graphs link to the source code. The gallery is maintained by SAS Technical Support. It is available at the following URL:

  https://support.sas.com/en/knowledge-base/graph-samples-gallery.html

- The Focus Area Graphics site provides a simple interface to business and analytical graphs. The site is maintained by the SAS Data Visualization team. It is available at the following URL:

https://support.sas.com/rnd/datavisualization/index.htm

- You can share your questions, suggestions, and experiences related to graphics on the Graphics Programming community site. It is available at the following URL:

  https://communities.sas.com/t5/Graphics-Programming/bd-p/sas_graph

In addition, SAS offers instructor-led training and self-paced e-learning courses to help you get started with platform graphics software. The SAS Training site provides information about the courses that are available. It is available at the following URL:

sas.com/training

PART 3

SAS Statements Used with the Procedures

Chapter 6
Global Statements 93

Chapter 7
Common Procedure Statements 141

6

Global Statements

About the Global Statements 93

Dictionary 94
FOOTNOTE Statement 94
ODS GRAPHICS Statement 102
TITLE Statement 131

About the Global Statements

Global statements are statements that can be specified anywhere in a SAS program. These statements typically specify attributes of a graph such as titles and footnotes for the duration of a SAS session. When executed, these statements remain in effect until modified or canceled, or until the SAS session is terminated.

The procedures support the following global statements:

FOOTNOTE
adds footnotes to your graphs.

ODS GRAPHICS
enables you manage the settings for your graphics output.

TITLE
adds titles to your graphs.

Dictionary

FOOTNOTE Statement

The FOOTNOTE statement controls the content, appearance, and placement of footnote text. You can write up to 10 lines of text at the bottom of the graph.

Valid in: anywhere in your program

Syntax

FOOTNOTE <*1–10*> <*ods-format-options*> <'*text-string*'>;

Without Arguments

Using FOOTNOTE without arguments cancels all existing footnotes.

Optional Arguments

<1-10>
specifies the relative line that contains the footnote text.

Valid in Anywhere

Default If you omit a number, SAS assumes a value of 1. Therefore, you can specify FOOTNOTE or FOOTNOTE1 for the first footnote line.

Tips For footnotes, lines are pushed up from the bottom. The FOOTNOTE statement with the highest number appears on the bottom line.

You can create footnotes that contain blank lines between the lines of text. For example, if you specify text with a FOOTNOTE statement and a FOOTNOTE3 statement, there is a blank line between the two lines of text.

ods-format-options
specifies formatting options for the ODS HTML5, PDF, and RTF destinations.

BCOLOR=*color*
specifies the background color of the footnote block.

You can optionally draw a box around a text string with the BOX= option. The BCOLOR= option controls the background color of the box. By default, the background color of the box is the same as the background color for the

entire graph. You can specify colors using a number of different color-naming schemes.

Valid in Base SAS, ODS Graphics, SAS/GRAPH

Alias BC=

Restriction If you omit the BOX= option, the BCOLOR= option is ignored.

BOLD

specifies that the font weight is bold for the text string.

Valid in Base SAS, ODS Graphics

Default For footnotes, the default font weight is specified by the FONTWEIGHT attribute of the GraphFootnoteText style element in the current style.

Restriction The BOLD option is for template-based procedures only.

COLOR=*color*

specifies the color for the text. You can specify colors using a number of different color-naming schemes.

The color stays in effect until another COLOR= specification is encountered. Change colors as often as you like. For example, the following code produces a footnote where the first word is red and the second word is blue:

```
footnote color=red "Total Sales" color=blue;
```

Valid in Base SAS. ODS Graphics, SAS/GRAPH

Alias C=

Default For footnotes, the default font color is specified by the COLOR attribute of the GraphFootnoteText style element in the current style.

See "BCOLOR=*color*" on page 94

Example "Customizing Titles and Footnotes By Using the Output Delivery System" in *SAS Global Statements: Reference*

FONT=*system-font*

specifies the system font to use for the text string.

Valid in Base SAS, ODS Graphics, SAS/GRAPH

Alias F=

Default For footnotes, the default font is specified by the FONTFAMILY attribute of the GraphFootnoteText style element in the current style.

Note SAS/GRAPH software fonts such as SWISS cannot be used with statistical graphics procedures.

HEIGHT=*numeric-value<units>*
specifies the size of the text. You can also specify the unit of measurement. Without a unit of measure, the number becomes a relative size.

If you do not specify a unit, then the size of the text is approximately 12*numeric-value* points. For example, if you specify HEIGHT=2, then the text size is approximately 24 points.

numeric-value

<units>
is a nonnegative number.

Measurement Units

cm	Centimeters
in	Inches
pct or %	Percentage
pt	A printer's point

Valid in: Base SAS, ODS Graphics, SAS/GRAPH

Alias: H=

Default: For the first footnote (footnote or footnote1), the default font size is specified by the FONTSIZE attribute of the GraphTitleText style element in the current style. For subsequent footnotes (FOOTNOTE2...n)), the size is the FONTSIZE attribute of the GraphLabelFont style attribute.

Style reference: Height attribute of the GraphTitle1Text (FOOTNOTE1) and the GraphTitleText (FOOTNOTE2...n) elements

Examples

"Ordering Axis Tick Marks with SAS Date Values" in *SAS/GRAPH: Reference*

"Enhancing Titles" in *SAS/GRAPH: Reference*

ITALIC
specifies that the font style is italic for the text string.

Valid in: Base SAS, ODS Graphics

Default: For footnotes, the default font style is specified by the FONTSTYLE attribute of the GraphTitleText style element in the current style.

JUSTIFY= LEFT | CENTER | RIGHT
specifies the alignment of the text string. You can specify one of the following values:

LEFT
specifies left justification.

Alias L

CENTER
aligns the text in the center.

Alias C

Note By default, footnotes are sometimes centered over the data area of the graph. However, when you explicitly specify center justification, footnotes are always centered over the graph area. All the text strings following JUSTIFY= are treated as a single string and are displayed as one line that is left-, right-, or center-aligned.

RIGHT
aligns the text to the right.

Alias R

Valid in Base SAS, ODS Graphics, and SAS/GRAPH

Alias J=

Default CENTER

Notes The JUSTIFY= option affects all of the text strings that follow it in your FOOTNOTE statement. You can specify multiple alignments by using more than one JUSTIFY= option. For example, the following code creates a footnote where the first string is aligned to the left and the second string is aligned to the right:

```
footnote justify=left "Example 2" justify=right "Graph 3";
```

You can use the JUSTIFY= option to produce multiple lines of text by repeating the JUSTIFY= option with the same value before the text string for each line. Multiple lines of text with the same justification are blocked together. For example, this FOOTNOTE statement produces a three-line footnote, with each line right-justified:

```
footnote  justify=right "First Line"
     justify=right "Second Line"
     justify=right "Third Line";
```

You can get the same effect with three FOOTNOTE statements, each specifying JUSTIFY=RIGHT. You can produce a block of text by specifying the same justification for multiple text strings. You can then change the justification for an additional text string. The result is that the text is placed on the same line as the first string specified in the statement.

Using the JUSTIFY= option after one text string and before another can reset some options to their default values.

Example "Customizing Titles and Footnotes By Using the Output Delivery System" in *SAS Global Statements: Reference*

LINK='*URL*'
specifies a uniform resource locator (URL) to which a footnote links.

Valid in Base SAS, SAS/GRAPH

ODS destination HTML, RTF, PRINTER

Tip The visual properties for LINK= always come from the current style.

UNDERLIN=*0-3*
specifies whether the subsequent text is underlined. UNDERLIN=0 indicates no underlining. Values 1, 2, and 3 indicate the same type of underlining.

Valid in SAS/GRAPH

Alias U=

See " ROTATE=degrees" in *SAS/GRAPH: Reference*

Note The defaults for how ODS renders the FOOTNOTE statement come from style elements that relate to system footnotes in the current style. The FOOTNOTE statement syntax with *ods-format-options* is a way to override the settings that are provided by the current style. The current style varies according to the ODS destination. For more information about how to determine the current style, see "Understanding Styles, Style Elements, and Style Attributes" in *SAS Output Delivery System: Procedures Guide* and "Concepts: TEMPLATE Procedure" in *SAS Output Delivery System: Procedures Guide*.

Tip You can specify these options by letter, word, or words by preceding each letter or word of the *text* by the option. For example, this code makes the footnote "Red, White, and Blue" appear in different colors.

```
footnote color=red "Red," color=white "White, and" color=blue "Blue";
```

<'text-string'>
is a text string with a maximum length of 255 bytes. You must enclose text strings in either single or double quotation marks. The text appears exactly as you enter it in the statement, including uppercase and lowercase characters and spaces. Footnotes automatically wrap to additional lines if necessary. If the length of the specified title is greater than the value of the LINESIZE option, the title is truncated to the line size.

To use single quotation marks or apostrophes within the footnote, you can do either of the following:

- use a pair of single quotation marks together:

```
footnote 'All''s Well That Ends Well';
```

- enclose the text in double quotation marks:

```
footnote "All's Well That Ends Well";
```

You can specify multiple text strings in FOOTNOTE statements. If you want each string to be separated by a blank, add a blank to the beginning of the second and subsequent text strings rather than at the end the preceding string, as in this example:

```
footnote color=red "Sales:" color=blue " 2020";
```

With fonts that support Unicode, you can specify special characters by specifying a hexadecimal constant in quotation marks. A trailing `x` identifies a string as a hexadecimal constant. You must designate the hexadecimal constant as a Unicode specification and escape it: `(*ESC*){Unicode '`*`hexadecimal-value`*`'x}`. For example:

```
footnote "Regression with Confidence Limits ((*ESC*){unicode '03B1'x}=.05)";
```

This statement produces the footnote, "Regression with Confidence Limits (α = .05)" because '03B1'x is the hexadecimal value for the lowercase Greek letter alpha in all Unicode fonts.

> **TIP** You can instead specify lowercase Greek letters by name, and uppercase Greek letters by name_u: `(*ESC*){Unicode alpha}=.05` and `(*ESC*){Unicode alpha_u}=.05`.

For more information see "Specifying Special Characters Using Character and Hexadecimal Codes" in *SAS/GRAPH: Reference*.

You can also customize footnotes by inserting BY variable lines (#BYLINE), BY variable values (#BYVAL), or BY variable names (#BYVAR), in footnotes that are specified in procedure steps. Embed the items in the specified footnote text string at the position where you want the substitution text to appear. See "Using #BYLINE, #BYVAL, and #BYVAR Substitution in URL and FOOTNOTE Text Strings" in *SAS Global Statements: Reference*.

Tips For compatibility with previous releases, SAS accepts some text without quotation marks. When you write new programs or update existing programs, *always* enclose text in quotation marks.

You can use macro variables and macros to change the information in FOOTNOTE statements. If the footnote is enclosed in double quotation marks (""), the text indicated is substituted into the footnote. If the footnote is enclosed in single quotation marks ('), the text is not substituted.

See For more information about including quotation marks as part of the footnote, see "SAS Constants in Expressions" in *SAS Programmer's Guide: Essentials*.

Details

Summary of Text Options

ods-format-options can be one or more of the following:

- appearance options:
  - BOLD
  - COLOR= *color*
  - FONT= "*system-font*"
  - HEIGHT= *numeric-value* <*units*>
  - ITALIC
- placement and spacing options:
  - JUSTIFY= LEFT | CENTER | RIGHT
  - LSPACE= *numeric-value* <*units*>
- boxing and drawing options:
  - BCOLOR= *color*
  - BOX= *numeric-value*
  - BSPACE= *numeric-value* <*units*>

The following options are not supported by the ODS Graphics procedures:

- ANGLE=
- BLANK=
- DRAW=
- LANGLE=
- LINK=
- MOVE=
- ROTATE=
- UNDERLIN=
- WRAP

Using FOOTNOTE Statement

A FOOTNOTE statement takes effect when the step or RUN group with which it is associated executes. After you specify a footnote for a line, SAS repeats the same footnote on all pages until you cancel or redefine the footnote for that line. When a FOOTNOTE statement is specified for a given line, it cancels the previous FOOTNOTE statement for that line and for all footnote lines with higher numbers.

Operating Environment Information: If the length of the specified title is greater than the value of the LINESIZE option, the title is truncated to the line size.

Differences When Using TITLE and FOOTNOTE Statements with the SGRENDER Procedure

Titles and footnotes behave differently for the SGRENDER procedure than for the other ODS Graphics procedures.

Normally, the title and footnote are displayed within the graph's border. However, when you render a Graph Template Language (GTL) graph using PROC SGRENDER, the title and footnote are displayed in the ODS document, outside of the graph's border. To add titles and footnotes to the graph, use the GTL ENTRYTITLE and ENTRYFOOTNOTE statements.

Substituting BY Line Values in a Text String

These options are available if a BY statement is in effect:

#BYLINE
substitutes the entire BY line without leading or trailing blanks for #BYLINE in the text string. The BY line uses the format *variable-name=value*.

#BYVAL*n* | #BYVAL(*BY-variable-name*)
substitutes the current value of the specified BY variable for #BYVAL in the text string. Specify the variable with one of these:

n
specifies a variable by its position in the BY statement. For example, #BYVAL2 specifies the second variable in the BY statement.

BY-variable-name
specifies a variable from the BY statement by its name. For example, #BYVAL(YEAR) specifies the BY variable, YEAR. *variable-name* is not case sensitive.

#BYVAR*n* | #BYVAR(*BY-variable-name*)
substitutes the name of the BY-variable or the label associated with the variable (whatever the BY line would normally display) for #BYVAR in the text string. Specify the variable with one of these:

n
specifies a variable by its position in the BY statement. For example, #BYVAR2 specifies the second variable in the BY statement.

BY-variable-name
specifies a variable from the BY statement by its name. For example, #BYVAR(SITES) specifies the BY variable, SITES. *Variable-name* is not case sensitive.

Note: A BY variable name displayed in a title or footnote is always in uppercase. If a label is used, then it appears as specified in the LABEL statement.

To use the #BYVAR and #BYVAL substitutions, insert the item in the text string at the position where you want the substitution text to appear. Both #BYVAR and #BYVAL specifications must be followed by a delimiting character. The character can be either a space or other non-alphanumeric character, such as a quotation mark. If no delimiting character is provided, then the specification is ignored and its

text remains intact and is displayed with the rest of the string. To allow a #BYVAR or #BYVAL substitution to be followed immediately by other text, with no delimiter, use a trailing dot (as with macro variables). The trailing dot is not displayed in the resolved text. If you want a period to be displayed as the last character in the resolved text, use two dots after the #BYVAR or #BYVAL substitution.

The substitution for #BYVAR or #BYVAL does not occur in the following cases:

- if you use a #BYVAR or #BYVAL specification for a variable that is not named in the BY statement. For example, you might use #BYVAL2 when there is only one BY-variable or #BYVAL(ABC) when ABC is non-existent or is not a BY-variable.
- if there is no BY statement at all

For both cases, no error or warning message is issued. The option specification is displayed with the rest of the string. The graph continues to display a BY line at the top of the page unless you suppress it by using the NOBYLINE option in an OPTION statement.

Units of Measurement

Some of the options in the FOOTNOTE statement give you the option to specify the unit of measurement. The following table lists the units that are supported:

Table 6.1 *Measurement Units*

Unit	Description
CM	centimeters
IN	inches
PCT or %	percentage
PT	point size, calculated at 100 dots per inch

The "HEIGHT=*numeric-value<units>*" on page 96 allows for the specification of a few more units than are mentioned in the previous table.

ODS GRAPHICS Statement

Enables or disables ODS Graphics processing and sets graphics environment options. This statement affects ODS template-based (ODS Graphics) graphics only. The ODS GRAPHICS statement does not affect device-based graphics (SAS/GRAPH).

Valid in: Anywhere

Category: ODS: Output Control

Default: ON.

Interaction: SAS/GRAPH device-based global statements such as GOPTIONS, SYMBOL, PATTERN, AXIS, and LEGEND do not affect template-based graphics. The ODS GRAPHICS statement does not affect device-based graphics.

Syntax

ODS GRAPHICS <OFF | ON> < / *options* > ;

Summary of Optional Arguments

ANTIALIAS
: smooths jagged edges of all components in the graph.

ANTIALIAS= ON | OFF
: specifies whether anti-aliasing is applied to the rendering of the line and markers in any graph.

ANTIALIASMAX= *n*
: specifies the maximum number of graphics elements before anti-aliasing is disabled.

ATTRPRIORITY=COLOR | NONE
: specifies a priority for cycling of the group attributes.

BORDER
: draws a border around each graph.

BORDER=ON | OFF
: specifies whether to draw a border around each graph.

BYLINE=NOBYLINE | TITLE | FOOTNOTE
: specifies how the BY line is displayed in graphs.

DATASKINMAX=*n*
: specifies the maximum number of graphical elements allowed per plot when a data skin is applied.

DISCRETEMAX=*n*
: specifies the maximum number of discrete values to be shown in any graph.

DRILLTARGET="_blank" | "_self" | "_parent" | "_top" | "frame-name"
: specifies the window that displays the drill-down output.

GROUPMAX=*n*
: specifies the maximum number of group values to be shown in any graph.

HEIGHT=*dimension*
: specifies the height of a graph.

IMAGEMAP
: specifies to generate data tips.

IMAGEMAP=ON | OFF
: specifies whether data tips are generated.

IMAGENAME="*filename*"
: specifies the base image file name.

LABELMAX=*n*
: specifies the maximum number of labeled areas before data label collision avoidance is disabled.

LABELPLACEMENT= GREEDY | SA
: specifies the label-placement algorithm to use for positioning labels in the graphs.

LEGENDAREAMAX=*n*
: specifies an integer that is interpreted as the maximum percentage of the overall graphics area that a legend can occupy.

LINEPATTERNOBSMAX=*n*
: specifies the maximum the number of vertices for a patterned line.

LOESSOBSMAX=*n*
: specifies an upper limit for the number of observations that can be used with a loess plot.

NBINSMAX=*n*
: specifies the maximum number of bins that are processed for histograms.

NOANTIALIAS
: does not smooth jagged edges of components other than text in the graph.

NOBORDER
: does not draw a border around the graph.

NOIMAGEMAP
: specifies not to generate data tips.

NOSCALE
: does not scale the components of a graph proportionally.

NOSCALEMARKERS
: does not scale the markers with the graph size.

NOSUBPIXEL
: never uses subpixel rendering.

NXYBINSMAX=*n*
: specifies the maximum number of bins that are processed for heat maps.

OBSMAX=*n*
: specifies the maximum number of observations that are processed.

OFF
: disables ODS Graphics processing.

ON
: enables ODS Graphics processing.

OUTPUTFMT=*file-type* | STATIC
: specifies the output format used to generate image or vector graphic files.

PANELCELLMAX=*n*
: specifies the maximum number of cells in a graph panel where the number of cells is determined dynamically by classification variables.

POP
: restores the most recently pushed settings from the stack.

PUSH
: pushes the current ODS GRAPHICS settings to a stack.

RESET
: resets all *options* to their defaults.

RESET=ALL | *option*
: resets one or all of the ODS GRAPHICS options to their defaults.

SCALE
: scales the components of a graph proportionally.

SCALE=ON | OFF
: specifies whether the content of a graph is scaled proportionally.

SCALEMARKERS
: scales the markers with the graph size.

SCALEMARKERS=ON | OFF
: specifies whether the plot markers are to be scaled with the graph size.

SHOW
: writes the current ODS Graphics settings to the SAS log.

STACKDEPTHMAX=*n*
: specifies the maximum stack depth for PUSH and POP requests.

SUBPIXEL
: always uses subpixel rendering, when applicable, for rendering lines and bars.

SUBPIXEL=ON | OFF
: specifies whether subpixel rendering should be used for rendering ODS Graphics.

TIPMAX=*n*
: specifies the maximum number of distinct data tip areas allowed before data tips are disabled.

TOTALCELLMAX=*n*
: specifies the maximum number of total cells in a graph where the number of cells is determined dynamically by classification variables.

WIDTH=*dimension*
: specifies the width of any graph.

Without Arguments

If ODS Graphics is currently disabled, then specifying the ODS GRAPHICS statement without options enables it. If ODS Graphics is currently enabled, then specifying the ODS GRAPHICS statement leaves it enabled.

Optional Arguments

You can specify one of the following arguments before '/:'

ON
: enables ODS Graphics processing. This is the default if no argument is used.

Alias YES

Example Enable ODS graphics:

```
ods graphics on </option(s)>;
```

OFF
disables ODS Graphics processing.

Alias NO

Example Disable ODS graphics:
```
ods graphics off;
```

Optional Arguments that Follow '/'

You can specify one or more of the following arguments after '/:'

ANTIALIAS
smooths jagged edges of all components in the graph. Anti-aliasing smooths the appearance of lines and some markers. Text displayed in the graph is always anti-aliased. For graphical displays that plot large numbers of points it is recommended that NOANTIALIAS or ANTIALIAS=OFF be specified for performance considerations.

Default ANTIALIAS

Restrictions If the number of markers or lines in the plot exceeds the number specified by the ANTIALIASMAX= option, then the ANTIALIAS option is disabled. This is true even if you specify the option ANTIALIAS=ON or ANTIALIAS.

This option is ignored by the SGMAP, SGPIE, and SGSCATTER procedures.

ANTIALIAS= ON | OFF
specifies whether anti-aliasing is applied to the rendering of the line and markers in any graph. Anti-aliasing smooths the appearance of lines and some markers. Text displayed in the graph is always anti-aliased. For graphical displays that plot large numbers of points it is recommended that ANTIALIAS=OFF be specified for performance considerations.

ON
smooths jagged edges of all components in the graph.

Alias YES

OFF
does not smooth jagged edges of components other than text in the graph.

Alias NO

Default ANTIALIAS or ANTIALIAS=ON | YES

Restrictions If the number of markers or lines in the plot exceeds the number specified by the ANTIALIASMAX= option, then the ANTIALIAS option is disabled. This is true even if you specify the option ANTIALIAS=ON or ANTIALIAS.

This option is ignored by the SGMAP, SGPIE, and SGSCATTER procedures.

ANTIALIASMAX= *n*
specifies the maximum number of graphics elements before anti-aliasing is disabled. For example, if there are more than 400 scatter point markers to be anti-aliased and ANTIALIASMAX=400, then no markers are anti-aliased. The default value is 4000.

Note: The ANTIALIASMAX= option specifies the maximum number of graphics elements to be anti-aliased in each plot on a per-plot basis. The default remains at 4000. If any plot in a graph contains more than 4000 elements, anti-aliasing is disabled for that plot. Anti-aliasing is enabled for the rest of the graph in that case.

n
specifies a positive integer.

Default 4000

Restriction This option is ignored by the SGMAP, SGPIE, and SGSCATTER procedures.

ATTRPRIORITY=COLOR | NONE
specifies a priority for cycling of the group attributes.

COLOR
assigns priority to the color attribute rotation by cycling through the list of colors while holding the marker symbol and line pattern constant. When all of the colors are exhausted, the marker symbol and line style attributes increment to the next element, and then the colors in the list are repeated. This pattern repeats as needed.

NONE
does not use an attribute priority in the rotation pattern, even if one is set in the active style's AttrPriority attribute. The rotation pattern cycles progressively through the attribute lists.

Default The AttrPriority attribute of the graph style element, or NONE if the current style does not define the AttrPriority style attribute.

Interaction The default lists of data colors, contrast colors, marker symbols, and line patterns are set in the active style's GraphData1–GraphData*n* elements.

Tip Use the ATTRPRIORITY=NONE option if you want groups to be distinguished by color, marker, and line changes for all styles that use color.

BORDER
draws a border around each graph.

A border is a line that is drawn along the outer edges of the graph area. See "Components of a Graph" in *SAS Graph Template Language: Reference*.

Default BORDER

BORDER=ON | OFF
specifies whether to draw a border around each graph.

A border is a line that is drawn along the outer edges of the graph area. See "Components of a Graph" in *SAS Graph Template Language: Reference*.

ON
draws a border around each graph.

Alias YES

OFF
does not draw a border around each graph.

Alias NO

Default BORDER=ON

BYLINE=NOBYLINE | TITLE | FOOTNOTE
specifies how the BY line is displayed in graphs. The option specifies how the BY line is displayed when an analysis is run with a BY statement. By default, no BY line is displayed.

The following code is an example of how the placement of the BY line is controlled in most graph templates:

```
if (_BYTITLE_)
   entrytitle _BYLINE_ / textattrs=GraphValueText;
else
   if (_BYFOOTNOTE_)
      entryfootnote halign=left _BYLINE_;
   endif;
endif;
```

You can modify the graph template if you want to change how the BY line is displayed. Because most graphs have titles and few graphs have footnotes, the BY line looks better when it is displayed as a footnote. For complete documentation about the Graph Template Language, see *SAS Graph Template Language: User's Guide*.

When the BYLINE= option is specified, and there are BY groups, ODS creates a BY line and sets the appropriate special dynamic variables. The following table lists the special dynamic variables for BY lines. For complete documentation about special dynamic variables, see "Special Dynamic Variables" in *SAS Graph Template Language: User's Guide*.

Table 6.2 *Special Dynamic Variables for BY Lines*

BYFOOTNOTE	This variable is set to 1 when you specify a BY statement and the ODS GRAPHICS BYLINE= option

	is set to FOOTNOTE. Otherwise, the variable is set to 0 or is NULL.
BYTITLE	This variable is set to 1 when you specify a BY statement and the ODS GRAPHICS BYLINE= option is set to TITLE. Otherwise, the variable is set to 0 or is NULL.
BYLINE	This variable contains the text string that can be displayed as a title or footnote.

The variables in the table are set automatically only for analytical procedures that support ODS Graphics. For all other procedures, the variables are not set automatically (NULL). To determine whether the procedure that you are using supports ODS Graphics, refer to the procedure documentation.

NOBYLINE
specifies that no BY line is displayed. NOBYLINE is the default.

FOOTNOTE
specifies that the BY line is displayed as a left-justified graph footnote. This is the recommended setting.

TITLE
specifies that the BY line is displayed as a centered graph title. Specifying TITLE is not recommended because graphs are not designed to have additional title lines.

Default NOBYLINE

Restriction This option has no effect on the ODS Graphics procedures SGPLOT, SGPANEL, SGPIE, SGSCATTER, and SGRENDER. To remove BY lines in these procedures, use the NOBYLINE SAS system option. See "BYLINE System Option" in *SAS System Options: Reference*. The SGMAP procedure does not support the BY statement.

DATASKINMAX=*n*
specifies the maximum number of graphical elements allowed per plot when a data skin is applied.

n
specifies a positive integer.

Default 200

DISCRETEMAX=*n*
specifies the maximum number of discrete values to be shown in any graph. Bar charts and box plots are examples of affected plot types. Scatter plots and other plot types can be affected if the data to be plotted is discrete or the axis is discrete.

n
specifies a positive integer.

Default 1000

Tips Some plot layers might be unaffected by the DISCRETEMAX= option, and those layers are rendered. If all layers are affected, a blank graph is rendered.

If the value specified by the DISCRETEMAX= option is exceeded by any plot layer in the graph, that layer is not drawn and a warning message is issued. In that case, use the DISCRETEMAX= option to increase the maximum number of discrete values that are allowed. The log message includes a suggested value for DISCRETEMAX=.

DRILLTARGET="_blank" | "_self" | "_parent" | "_top" | "frame-name"
specifies the window that displays the drill-down output.

Note: This option is supported only for HTML.

"_blank"
opens a new browser window to display the drilldown output.

Default _blank is the default.

Requirements You must enclose *_blank* in quotation marks.

You must specify *_blank* in lowercase.

"_self"
opens the drill-down output in the same window.

Requirements You must enclose *_self* in quotation marks.

You must specify *_self* in lowercase.

"_parent"
opens the drill-down output in the parent frame.

Requirements You must enclose *_parent* in quotation marks.

You must specify *_parent* in lowercase.

"_top"
opens the drill-down output in the full body of the window.

Requirements You must enclose *_top* in quotation marks.

You must specify *_top* in lowercase.

"frame-name"
opens the drill down output in the named frame in the current window. If the name does not exist, the output is opened in a new window.

Requirement You must enclose *frame–name* in quotation marks.

GROUPMAX=*n*
specifies the maximum number of group values to be shown in any graph. Any graph that supports the GROUP= option is affected.

n
specifies a positive integer.

Default 1000

Tip If the value specified by the GROUPMAX= option is exceeded by any plot layer in the graph, that layer is rendered. The system ignores the GROUP= option and issues a warning message. In that case, use the GROUPMAX= option to increase the maximum number of group values that are allowed. The log message includes a suggested value for GROUPMAX=.

HEIGHT=*dimension*
specifies the height of a graph.

dimension
is a nonnegative number followed by one of these units of measure:

Table 6.3 *Units of Measure for Dimension*

cm	Centimeters
in	Inches
mm	Millimeters
pct or %	Percentage
pt	Point size (72 points = 1 inch)
px	Pixels

Default Determined by the software, or the value of the DesignHeight= option in a STATGRAPH template. Typically, the value is 480px.

Tips If only the HEIGHT= option is specified, then the default aspect ratio of the graph is maintained.

When you change the size of a graph, by default graph elements such as markers, lines, text, and titles are scaled proportionally. This scaling takes effect even when you specify dimensions for the graph elements. To disable the scaling, specify the NOSCALE option.

IMAGEMAP
specifies to generate data tips.

Default NOIMAGEMAP

Restriction This option affects HTML output only.

Note Image maps are supported with SVG output using HTML5. However, image maps are supported only when the HTML5 SVG mode is INLINE (the default value).

IMAGEMAP=ON | OFF

controls data tips and drill down generation. Data tips are pieces of explanatory text that appear when you hold the mouse pointer over the data portions of a graph contained in an HTML page.

ON

specifies to generate data tips.

Alias YES

OFF

specifies not to generate data tips.

Alias NO

Default NOIMAGEMAP or IMAGEMAP=OFF | NO

Restriction This option affects HTML output only.

Note Image maps are supported with SVG output using HTML5. However, image maps are supported only when the HTML5 SVG mode is INLINE (the default value).

IMAGENAME="*filename*"

specifies the base image file name. If more than one image is generated, each is assigned a file name that consists of a base name followed by a number in order to create unique names. This numbering can be reset with the RESET=INDEX option. Path information (if needed) can be set with the GPATH= option on the ODS destination statement. The default path is the current output directory. A file extension for *filename* is automatically generated based on the OUTPUTFMT= option.

Note: If a BY statement is in effect, you can uniquely name image files based on BY groups. To do this, insert #BY text into the image name. For more information, see "Substituting BY Values in the Image Name" on page 125.

Default The name of the output object.

Restrictions *filename* must be a single name. It must not include any path specification or image-format name extension.

filename can contain only the following types of characters: letter, digit, underscore, and space.

Requirement You must enclose *filename* in quotation marks.

See "Specifying and Resetting the Image Name" on page 124

LABELMAX=*n*

specifies the maximum number of labeled areas before data label collision avoidance is disabled. For example, if there are more than 50 points to be labeled and LABELMAX=50, then collision avoidance is turned off and the labels are all displayed at the top right of the data points.

n

specifies a positive integer.

Default 200

Restriction Data label collision avoidance is turned off under the following conditions:

- The number of observations with nonmissing labels exceeds the value specified by LABELMAX=.
- The number of observations exceeds five times the value specified by LABELMAX=.

A message is then sent to the SAS log.

Tip To turn off collision avoidance, specify LABELMAX=0.

LABELPLACEMENT= GREEDY | SA

specifies the label-placement algorithm to use for positioning labels in the graphs. The following labels are affected:

- data labels for needle plots, scatter plots, series plots, step plots, and vector plots
- vertex labels for line charts
- curve labels when the curve label is positioned at the start or end of the curve

GREEDY

specifies the Greedy method for managing label collision. The Greedy method tries different placement combinations in order to find an optimal approximation that avoids collisions. Label placement using this method is often less optimal than label placement using the Simulated Annealing (SA) method. However, depending on the number of data points and the potential for label collisions, the Greedy process can be significantly faster.

SA

specifies the Simulated Annealing method for managing label collision. The SA method attempts to determine the global minimization-of-cost function, which is based on a simulated annealing algorithm. The resulting label placement is usually better than placement using the Greedy method. However, depending on the number of data points and the potential for label collisions, the SA method can be significantly slower.

Restriction For BANDPLOT and LINECHART, the SA method has no effect on the curve labels when the CURVELABELPOSITION= option specifies START or END.

Default GREEDY

LEGENDAREAMAX=*n*
specifies an integer that is interpreted as the maximum percentage of the overall graphics area that a legend can occupy.

n
specifies a positive integer.

Alias MAXLEGENDAREA=

Default 20

Range 0–100

Tip To turn off the legend, specify LEGENDAREAMAX=0. No warning is issued when the legend is turned off in this way.

LINEPATTERNOBSMAX=*n*
specifies the maximum the number of vertices for a patterned line. If the number of vertices exceeds the specified limit, the plot is not drawn.

n
specifies a positive integer.

Default 10000

Restriction This option is ignored by the SGMAP, SGPIE, and SGSCATTER procedures.

Note If the number of vertices exceeds the specified limit, the plot is not drawn and a note is written to the SAS log. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the log message, or change the line pattern to SOLID.

LOESSOBSMAX=*n*
specifies an upper limit for the number of observations that can be used with a loess plot. If the number of observations of the loess plot exceeds the specified limit, the loess plot is not drawn.

For example, the following specifies that the most observations a loess plot can have is 1000.

```
LOESSOBSMAX=1000
```

Alias LOESSMAXOBS=

Default 5000

NBINSMAX=*n*
specifies the maximum number of bins that are processed for histograms. This option affects both computed and parameterized histograms, when available.

n
specifies a positive integer.

Default 10000

Requirement The NBINSMAX= value must be greater than or equal to the default number of bins in the histogram. If the NBINXMAX= value is less than the default number of bins, an error occurs and the histogram is not drawn. In that case, an error message is written to the SAS log, which provides a minimum NBINSMAX= value that you can use.

Interaction If you specify the number of bins in the histogram plot statement NBINS= option, that option is honored if the NBINS= value specified is less than or equal to the NBINXMAX= value. If the NBINS= value specified exceeds the NBINSMAX= value, an error occurs, and the histogram is not drawn.

NOANTIALIAS
does not smooth jagged edges of components other than text in the graph.

Restriction This option is ignored by the SGMAP, SGPIE, and SGSCATTER procedures.

NOBORDER
does not draw a border around the graph.

NOIMAGEMAP
specifies not to generate data tips.

NOSCALE
does not scale the components of a graph proportionally.

NOSCALEMARKERS
does not scale the markers with the graph size.

NOSUBPIXEL
never uses subpixel rendering.

Restriction This option is ignored by the SGMAP, SGPIE, and SGSCATTER procedures.

NXYBINSMAX=*n*
specifies the maximum number of bins that are processed for heat maps. This option affects both computed and parameterized heat maps. The option applies to the product of the X and Y bins.

n
specifies a positive integer.

Default 100000

Interaction If you specify the number of bins in the heat map statement, that option is honored regardless of the NXYBINSMAX= option.

OBSMAX=*n*
specifies the maximum number of observations that are processed. If the number of observations in the data set exceeds the value specified for OBSMAX=, the procedure step terminates with a log message.

n
specifies a positive integer.

Alias MAXOBS=

Default 2 million observations

OUTPUTFMT=*file-type* | STATIC
specifies the output format used to generate image or vector graphic files. If the image or vector graphic format is not valid for the active output destination, the format is automatically changed to the default format for that destination.

file-type
is the image or vector graphic format to be generated. See "Supported Image Formats for Output Destinations" on page 127.

STATIC
uses the best quality static image format for the active output destination. This is the default output format.

> **TIP** The STATIC keyword can be used to reset the output format to its default state.

Default STATIC

See "Specifying the Image Format" on page 126

PANELCELLMAX=*n*
specifies the maximum number of cells in a graph panel where the number of cells is determined dynamically by classification variables. If the number of cells in the panel exceeds the specified limit, the panel is not drawn.

n
specifies a positive integer.

Default 1000

Note Graphs with DataPanel or DataLattice templates layouts are affected. In the ODS Graphics Procedures, this option affects graphs that are created with the SGPANEL procedure. If the value specified by the PANELCELLMAX= option is exceeded by any of these layouts, an empty graph is rendered and a warning message is issued.

Tip You can use the TOTALCELLMAX= option to control the total maximum number of cells in the graph.

POP
restores the most recently pushed settings from the stack. For each PUSH action, you can specify a POP request. ODS issues a warning if you specify POP without a corresponding PUSH. In that case, nothing is popped because nothing has been pushed.

PUSH
pushes the current ODS GRAPHICS settings to a stack. This feature enables you to temporarily save your custom settings in a stack and later restore those settings.

The pushed settings remain in the stack in the current SAS session until they are popped or the stack is emptied.

Interaction You can specify PUSH as many times as you like up to the limit that is defined by the STACKDEPTHMAX= option. You can also use STACKDEPTHMAX= to empty the stack. For more information, see "Managing the Stack Depth" on page 131.

Note Order of specification is important when using the PUSH and POP options. For more information, see "About PUSH and POP" on page 130.

Tip Use the SHOW option to show the current ODS GRAPHICS settings.

See "Temporarily Saving and Restoring ODS GRAPHICS Settings" on page 130

RESET
resets all *options* to their defaults.

RESET=ALL | *option*
resets one or all of the ODS GRAPHICS options to their defaults.

ALL
resets all of the ODS GRAPHICS options to their defaults.

option
specifies the name of an option to reset.

Option can be one of the following option names:

ANTIALIAS	IMAGENAME	OUTPUTFMT
ANTIALIASMAX	INDEX	PANELCELLMAX
ATTRPRIORITY	LABELMAX	SCALE
BORDER	LABELPLACEMENT	SCALEMARKERS
BYLINE	LEGENDAREAMAX	STACKDEPTHMAX
DATASKINMAX	LINEPATTERNOBSMAX	SUBPIXEL
DISCRETEMAX	LOESSOBSMAX	TIPMAX
DRILLTARGET	LOESSOBSMAX	TOTALCELLMAX
GROUPMAX	NBINSMAX	WIDTH
HEIGHT	NXYBINSMAX	
IMAGEMAP	OBSMAX	

SCALE
scales the components of a graph proportionally.

Default SCALE

SCALE=ON | OFF
specifies whether the content of a graph is scaled proportionally.

ON
scales the components of a graph proportionally.

Alias YES

OFF
does not scale the components of a graph proportionally.

Alias NO

Default SCALE=ON

SCALEMARKERS
scales the markers with the graph size.

Default SCALEMARKERS

SCALEMARKERS=ON | OFF
specifies whether the plot markers are to be scaled with the graph size. The scaling factor is based on the height of the graph cells and the height of the graph.

ON
scales the markers with the graph size.

Alias YES

OFF
does not scale the markers with the graph size.

Alias NO

Default SCALEMARKERS=ON

Restriction Scaling is done only if the graph contains multiple cells or single nested cells.

SHOW
writes the current ODS Graphics settings to the SAS log.

This option enables you to verify which settings are in effect. The option is especially useful when you use the PUSH and POP options to restore settings. For more information, see "Temporarily Saving and Restoring ODS GRAPHICS Settings" on page 130.

If no options have been explicitly set with an ODS GRAPHICS statement, then SHOW lists those options for which the default values are determined by the ODS GRAPHICS statement. Options that derive their default value by other means such as ATTRPRIORITY=, IMAGENAME=, HEIGHT=, and WIDTH= are not listed by default.

The following statement resets all settings and shows the default values.

```
ods graphics / reset=all show;
```

Here is an example of the default output for 2023.05 and later releases:

```
ODS Graphics Settings
---------------------
Output format:                     STATIC
By line:                           NOBYLINE
Antialias:                         ON
Border:                            ON
Image map:                         OFF
Scale:                             ON
Scale markers:                     ON
Maximum tips:                      500
Maximum labels:                    200
Maximum antialias observations:    4000
Maximum legend area:               20%
Maximum group values:              1000
Maximum discrete values:           1000
Maximum panel cells:               1000
Maximum Loess observations:        5000
Maximum data skins:                200
Maximum stack depth:               1024
Stack depth:                       0
Maximum Observations:              2000000
Maximum Histogram Bins:            10000
Maximum Heatmap Bins:              100000
Maximum Obs for Patterned Lines:   10000
Maximum Total Cells per BY-group:  2000
```

Note: The default list is slightly different in the prior releases.

If you have explicitly set other options in an ODS GRAPHICS statement, then SHOW includes those settings along with the defaults.

Order of specification is important when using the SHOW option. For example, the following statement shows the current settings and then sets the NOBORDER option.

```
ods graphics / show noborder;
```

However, the following statement sets the NOBORDER option and then shows the settings. The NOBORDER setting is shown in the log along with the other settings that are in effect.

```
ods graphics / noborder show;
```

The following statement resets all settings. It then sets the image width and height, and then shows the default settings along with the specified width.

```
ods graphics / reset=all width=5in height=4in show;
```

Here are the default values plus the image width and height, as displayed in the SAS log:

```
ODS Graphics Settings
---------------------
Output format:                     STATIC
By line:                           NOBYLINE
```

```
Antialias:                          ON
Border:                             ON
Image map:                          OFF
Scale:                              ON
Scale markers:                      ON
Maximum tips:                       500
Maximum labels:                     200
Maximum antialias observations:     4000
Maximum legend area:                20%
Maximum group values:               1000
Maximum discrete values:            1000
Maximum panel cells:                1000
Maximum Loess observations:         5000
Maximum data skins:                 200
Image width:                        5in
Image height:                       4in
Maximum stack depth:                1024
Stack depth:                        0
Maximum Observations:               2000000
Maximum Histogram Bins:             10000
Maximum Heatmap Bins:               100000
Maximum Obs for Patterned Lines:    10000
Maximum Total Cells per BY-group:   2000
```

Tip If you have specified the settings for some options but want to see the default values without losing your specified settings, issue the following two statements. The first statement pushes your specified settings, resets all settings, and then lists options for which ODS currently knows the default values. The second statement restores your previous settings.

```
ods graphics / push reset=all show;
ods graphics / pop;
```

STACKDEPTHMAX=*n*

specifies the maximum stack depth for PUSH and POP requests. The stack is used to temporarily store ODS GRAPHICS settings when you issue PUSH requests. PUSH saves the current settings to the stack and increments the stack depth. POP restores the most recently saved settings from the stack and decrements the stack depth.

n

specifies a positive integer.

If *n* is less than the current stack depth, then the stack is popped until its depth equals *n*. Popping the stack does not affect other option settings.

Defaults 1024 is the default maximum depth

0 is the default depth

Tips To empty the stack and then reset it to the default maximum depth, issue the following statement:

```
ods graphics / stackdepthmax=0 reset=stackdepthmax;
```

You can use any of the following commands to reset the stack to its default maximum depth:

```
reset=stackdepthmax
```

```
reset=all
reset
stackdepthmax=1024
```

See "Managing the Stack Depth" on page 131

SUBPIXEL
always uses subpixel rendering, when applicable, for rendering lines and bars.

Default Subpixel rendering is always enabled for vector-graphics output. It is enabled by default for image output, unless the graph contains a scatter plot or a scatter-plot matrix. In those cases, subpixel rendering is disabled by default.

Restriction This option is ignored by the SGMAP, SGPIE, and SGSCATTER procedures.

Requirement Antialiasing must be enabled for this option to have any effect. Antialiasing is enabled by default. To re-enable antialiasing, use the ANTIALIAS or ANTIALIAS=ON option in the ODS GRAPHICS statement.

Tip For a large amount of data, antialiasing is disabled when the number of observations exceeds the default maximum of 4000 observations. In that case, subpixel rendering is also disabled. To increase the maximum, use the ANTIALIASMAX= option in the ODS GRAPHICS statement.

See "Subpixel Rendering" on page 1857

SUBPIXEL=ON | OFF
specifies whether subpixel rendering should be used for rendering ODS Graphics. Subpixel rendering produces smoother curves and more precise bar spacing.

ON
always uses subpixel rendering, when applicable, for rendering lines and bars.

Alias YES

OFF
never uses subpixel rendering.

Alias NO

Default Subpixel rendering is always enabled for vector-graphics output. It is enabled by default for image output, unless the graph contains a scatter plot or a scatter-plot matrix. In those cases, subpixel rendering is disabled by default.

Restriction This option is ignored by the SGMAP, SGPIE, and SGSCATTER procedures.

Requirement Antialiasing must be enabled for this option to have any effect. Antialiasing is enabled by default. To re-enable antialiasing, use the ANTIALIAS or ANTIALIAS=ON option in the ODS GRAPHICS statement.

Tip For a large amount of data, antialiasing is disabled when the number of observations exceeds the default maximum of 4000 observations. In that case, subpixel rendering is also disabled. To increase the maximum, use the ANTIALIASMAX= option in the ODS GRAPHICS statement.

See "Subpixel Rendering" on page 1857

TIPMAX=*n*

specifies the maximum number of distinct data tip areas allowed before data tips are disabled. For example, if there are more than 400 points in a scatterplot, and TIPMAX=400, then no data tips appear. The default maximum value is 500.

Note: The TIPMAX= option specifies the maximum number of data-tip areas allowed before data tips are disabled. This threshold is applied separately for each plot. The default is 500. If any plot in a graph contains more than 500 data-tip areas, data tips are disabled for that plot. Data tips are enabled for the remaining plots in the graph.

n

specifies a positive integer.

Default 500

TOTALCELLMAX=*n*

specifies the maximum number of total cells in a graph where the number of cells is determined dynamically by classification variables.

If the number of cells exceeds the specified limit, the graph is not drawn. This option is useful when multiple classification variables or large amounts of data result in a large number of cells.

n

specifies a positive integer.

Default 2000

Notes If the number of cells exceeds the limit specified for TOTALCELLMAX=, the graph is not drawn and an error message is written to the SAS log. The error message includes a suggested value for TOTALCELLMAX=. You can use the TOTALCELLMAX= option to increase the maximum number of cells. Note, however, that the processing time to render the graph increases with an increase in the maximum number of cells.

Graphs with DataPanel or DataLattice templates layouts are affected. In the ODS Graphics Procedures, this option affects graphs that are created with the SGPANEL procedure.

Tip You can use the PANELCELLMAX= option to control the maximum number of cells in a graph panel.

WIDTH=*dimension*
specifies the width of any graph.

dimension
is a nonnegative number followed by one of these units of measure:

Table 6.4 *Units of Measure for Dimension*

cm	Centimeters
in	Inches
mm	Millimeters
pct or %	Percentage
pt	Point size (72 points = 1 inch)
px	Pixels

Default Determined by the software, or the value of the DesignWidth= option in a STATGRAPH template. Typically, this value is 640px.

Tips If only the WIDTH= option is specified, then the default aspect of the graph is maintained.

When you change the size of a graph, by default graph elements such as markers, lines, text, and titles are scaled proportionally. This scaling takes effect even when you specify dimensions for the graph elements. To disable the scaling, specify the NOSCALE option.

Details

Using the ODS GRAPHICS Statement

You can enable ODS Graphics by using one of the following equivalent statements:

```
ods graphics on;
ods graphics;
```

When you specify one of these statements before your procedure invocation, Base, SAS/STAT, SAS/ETS, and SAS/QC procedures support ODS Graphics, either by default, or when you specify procedure options for requesting particular graphs.

To disable ODS Graphics, specify the following statement:

```
ods graphics off;
```

Note: ODS Graphics is ON by default for these procedures: SGMAP, SGPANEL, SGPIE, SGPLOT, SGRENDER, and SGSCATTER. For other products, the initial state of ODS Graphics is determined by a SAS Registry setting.

Using the ODS GRAPHICS Statement for Batch Jobs

To generate ODS Graphics output in Linux batch jobs, you must set the DISPLAY system option before creating the output. To set the display, enter the following command:

```
export DISPLAY=<ip_address>:0
```

The *ip_address* is the TCP/IP address, or the name of a Linux terminal. Usually, the IP address of the Linux system where SAS is running would be used. If you do not set the DISPLAY variable, then you get an error message in the SAS log.

Specifying and Resetting the Image Name

Specifying the Image Name

For ODS Graphics output, by default, the ODS object name is used as the “root” name for the image output file. The following example creates a GIF image named REGPLOT:

```
ods graphics / imagename="regplot" outputfmt=gif;
```

The assigned name REGPLOT is treated as a "root" name and the first output created is named REGPLOT. Subsequent graphs are named REGPLOT1, REGPLOT2, and so on, with an increasing index counter. This numbering can be reset with the RESET=INDEX option.

Resetting the Image Name

The RESET=INDEX option enables you to reset the file name numbering sequence.

For a usage example, suppose that you are developing a paper or a presentation and it takes several submissions to get the desired output. You can use the RESET or RESET=INDEX option to force each output to replace itself:

```
ods graphics / reset=index ... ;
```

This specification causes all subsequent images to be created with the default or current image name.

When specifying this option, you can also specify the value for the index counter. The value that you specify determines the suffix for the next subsequent image. For example:

```
ods graphics / reset=index(100) imagename="MyName";
```

The next graph that you produce is named MYNAME100.

Substituting BY Values in the Image Name

If a BY statement is in effect for the data, then you can uniquely name image files based on BY groups. You do this by inserting any of the #BY* substitution items into the image file name. You can specify BY variable values or BY variable names for IMAGENAME=. You can combine the #BY* substitution item with other text. To do this, insert the #BY* item in the specified text string at the position where you want the substitution text to appear.

As explained in the IMAGENAME= description, the file extension is automatically generated based on the OUTPUTFMT= option.

Here are descriptions of the #BY* substitution items:

#BYVAL*n* | #BYVAL(*BY-variable-name*)
substitutes the current value of the specified BY variable for #BYVAL in the file name string. Specify the variable with one of these:

n
specifies a variable by its position in the BY statement. You must provide the position even if your BY statement contains only one variable. Otherwise, no substitution occurs.

BY-variable-name
specifies a variable from the BY statement by its name. *BY-variable-name* is not case sensitive.

Examples The following example specifies the first variable in the BY statement. The value of that variable is used for the image file name.

```
imagename="#byval1"
```

The following example specifies the second variable in the BY statement preceded by the text *sales_*:

```
imagename="sales_#byval2"
```

The following example specifies the YEAR variable in the BY statement. The value of that variable is used for the image file name:

```
imagename="#byval(year)"
```

The following example specifies the YEAR variable in the BY statement followed with the text *_sales*:

```
imagename="#byval(year)_sales"
```

#BYVAR*n* | #BYVAR(*BY-variable-name*)
substitutes the name of the BY variable or the label associated with the variable (whatever the BY line would normally display) for #BYVAR in the file name string. Specify the variable with one of these:

n
specifies a variable by its position in the BY statement. You must provide the position even if your BY statement contains only one variable. Otherwise, no substitution occurs.

BY-variable-name
specifies a variable from the BY statement by its name. *BY-variable-name* is not case sensitive.

Examples The following example specifies the first variable in the BY statement. The name of that variable is used for the image file name.

```
imagename="#byvar1"
```

The following example specifies the second variable in the BY statement preceded by the text *sales_*:

```
imagename="sales_#byvar2"
```

The following example specifies the YEAR variable in the BY statement. The name of that variable is used for the image file name:

```
imagename="#byvar(year)"
```

The following example specifies the YEAR variable in the BY statement followed with the text *_sales*:

```
imagename="#byvar(year)_sales"
```

Note: When using this feature, consider any file name length restrictions that might apply to your platform host. The file name length can vary greatly depending on the input data.

Specifying the Image Format

Each ODS destination uses a default format for its output. You can use the OUTPUTFMT= option in the ODS GRAPHICS statement to change the output format.

Note: Unless you have a special requirement for changing the image format, we recommend that you not change it. The default PNG format or default vector graphic format is far superior to other formats, such as GIF, because either supports transparency and a large number of colors. Also, PNG and vector graphics images require much less disk storage space than JPEG or TIFF formats.

If you want to generate vector graphics images, you can use the following OUTPUTFMT= values for each destination:

Table 6.5 *Generating Vector Graphics Output with ODS*

ODS Destination	**OUTPUTFMT=*value***
ODS EPUB	SVG graphics images are generated by default
ODS destination for Excel	SVG graphics images are generated by default
ODS HTML5	SVG graphics images are generated by default
ODS PDF	Vector graphics images are generated by default

ODS Destination	OUTPUTFMT=*value*
ODS destination for PowerPoint	SVG graphics images are generated by default
ODS RTF	`OUTPUTFMT=EMF`
ODS Measured RTF	`OUTPUTFMT=EMF`
ODS destination for Word	SVG graphics images are generated by default

There are cases in which vector graphics images cannot be generated for the output format that you specify. In those cases, a PNG image is generated instead. The following table lists the cases in which vector graphics images cannot be generated for the EMF, PDF, and SVG output formats.

Table 6.6 *Cases in Which Vector Graphics Are Not Supported*

	Supported (YES or NO)		
Case	**EMF**	**PDF**	**SVG**
3-D bivariate histograms	NO	NO	NO
Data skins and transparency[1]	NO	YES	YES
Gradient-fill contour plots[2]	NO	NO	NO
Images and transparency[1]	NO	YES	YES
Surface plots	NO	NO	NO

1 Data transparency or fill transparency.

2 GTL CONTOURPLOTPARM statement with GRADIENT, LINEGRADIENT, or LABELEDLINEGRADIENT specified for CONTOURTYPE=.

Note: The SGMAP procedure does not support vector-based output.

Supported Image Formats for Output Destinations

The following table lists all of the supported image formats for some ODS output destinations.

Table 6.7 Supported Image Formats for Output Destinations

Output Destination	Default Image Format	Supported Image Formats
EPUB	SVG	GIF, JPEG, JPG, PNG, SVG
ODS destination for Excel	SVG	EMF, JPEG, JPG, PNG, SVG
HTML5	SVG	GIF, JPEG, JPG, PNG, SVG
PDF	Native PDF	Native PDF, GIF, JPEG, JPG, PNG
ODS destination for PowerPoint	SVG	BMP, EMF, GIF, JPEG, JPG, PNG, SVG, TIFF
RTF and Measured RTF	EMF	EMF, JFIF, JPEG, JPG, PNG
ODS destination for Word	SVG	BMP, EMF, GIF, JPEG, JPG, PNG, SVG, TIFF

Description of Supported Image Formats

The following table provides descriptions of the supported image formats for ODS output destinations

Table 6.8 Description of Supported Image Formats

Image Format	Description
BMP (Microsoft Windows Device Independent Bitmap)	Supports color-mapped and true color images that are stored as uncompressed or run-length encoded data. BMP was developed by Microsoft Corporation.
CGM (Computer Graphics Metafile)	A free and open international standard file format for 2-D vector graphics, raster graphics, and text. This format is defined by ISO/IEC 8632.
DIB (Microsoft Windows Device Independent Bitmap)	See the description of BMP.

Image Format	Description
EMF (Enhanced Metafile Format)	Supports standard Enhanced Metafile Format.
EMF Plus (Enhanced Metafile Format Plus Extensions)	Supports Enhanced Metafile Plus Extensions that provides additional functionality, such as support of RGBA colors.
EMF Dual (Enhanced Metafile Format and Enhanced Metafile Format Plus Extensions)	Produces both EMF and EMF Plus formats simultaneously in the same output.
GIF (Graphics Interchange Format)	Supports only color-mapped images. GIF is owned by CompuServe, Inc.
JFIF (JPEG File Interchange Format)	Supports JPEG image compression. JFIF software is developed by the Independent JPEG Group.
JPEG or JPG (Joint Photographic Experts Group)	A file format that is used for storing noninteractive images.
PBM (Portable Bitmap Utilities)	Supports gray-scale, color, RGB, and bitmap files. The Portable Bitmap Utilities are a set of free utility programs that were developed primarily by Jef Poskanzer.
PDF (Portable Document Format)	A file format for electronic distribution and exchange of documents.
PNG (Portable Network Graphic)	Supports true color, gray-scale, and 8-bit images.
STATIC	Chooses the best image format for the current ODS destination.
SVG (Scalable Vector Graphics)	Is an XML language for describing two-dimensional vector graphics.
TIFF (Tagged Image File Format)	Internally supports a number of compression types and image types, including bitmapped, color-mapped, gray-scaled, and true color. TIFF was developed by Aldus Corporation and Microsoft Corporation and is used by a wide variety of applications (available if licensed).

Image Format	Description
XBM	X Window Bitmap
XPM	X Window Pixmap

Temporarily Saving and Restoring ODS GRAPHICS Settings

About PUSH and POP

Although you can use the RESET option to restore the default ODS GRAPHICS settings, there might be times when you want to save your current custom settings and later restore them. ODS enables you to temporarily store your custom settings in a stack created for this purpose, perform some other task with different settings, and then restore the previous settings.

The PUSH option saves the current ODS GRAPHICS settings to the stack and increments the stack depth. The POP option restores the most recently stored settings from the stack and decrements the stack depth.

This feature is useful when you write macros. Within a macro you can PUSH at the start of the macro and POP at the end. This enables your macro to have custom ODS GRAPHICS behaviors without affecting the calling environment.

You can specify PUSH as many times as you like up to the limit that is defined by the STACKDEPTHMAX= option. The pushed settings remain in the stack in the current SAS session until they are popped or the stack is emptied. For more information, see “Managing the Stack Depth” on page 131. For each PUSH option, you can specify a POP option. ODS issues a warning if you specify POP without a corresponding PUSH. In that case, nothing is popped because nothing has been pushed to the stack.

Order of specification is important when using the PUSH option. For example, the following statement pushes the NOBORDER option to the stack along with any other custom settings that are in effect.

```
ods graphics / noborder push;
```

A subsequent POP option restores the pushed settings including NOBORDER.

However, the following statement pushes the current custom settings and then sets the NOBORDER option.

```
ods graphics / push noborder;
```

Here, the subsequent POP option restores whatever border setting was in effect when the PUSH option was specified.

TIP Use the SHOW option to show the ODS GRAPHICS settings that are currently in effect.

Settings That Can Be Pushed

The PUSH and POP options apply to all ODS GRAPHICS options except the following: PUSH, POP, RESET=INDEX, and SHOW.

How Code Errors Affect the PUSH Operation

If the ODS GRAPHICS statement contains a syntax error, then the PUSH option is ignored.

For example, the PUSH option is ignored in the following statement:

```
ods graphics / antialias=bogus push;
```

A syntax error (BOGUS) in ANTIALIAS causes the parser to ignore the remaining options. However, a simple semantics error does not prevent the remaining options from being handled. In the following statement, the PUSH option is honored.

```
ods graphics / antialiasmax=-1 push;
```

In this statement, ANTIALIASMAX= –1 is invalid. The option expects a zero or a positive integer. In this case, a warning is issued to the log, but the PUSH occurs.

Note: Syntax errors in your code can have other unexpected results that are not described here.

Managing the Stack Depth

By default, the stack supports up to 1024 pushes. You can change the default by using the STACKDEPTHMAX= option.

If the specified STACKDEPTHMAX= value is less than the current stack depth, then the stack is popped until its depth equals the specified value. Popping the stack does not affect other option settings.

If you want to empty the stack, issue the following statement:

```
ods graphics / stackdepthmax=0 reset=stackdepthmax;
```

This statement first empties the stack of all PUSH requests and then restores the stack size to 1024.

TITLE Statement

The TITLE statement controls the content, appearance, and placement of title text. You can write up to 10 lines of text at the top of the graph.

Valid in: anywhere in your program

Syntax

TITLE <*1–10*> <*ods-format-options*> <'*text-string*'>;

Without Arguments

Using TITLE without arguments cancels all existing titles.

Optional Arguments

<1–10>
specifies the relative line that contains the title text.

Valid in Anywhere

Default If you omit a number, SAS assumes a value of 1. Therefore, you can specify TITLE or TITLE1 for the first title line.

Tips The TITLE statement with the highest number appears on the bottom line.

You can create titles that contain blank lines between the lines of text. For example, if you specify text with a TITLE statement and a TITLE3 statement, there is a blank line between the two lines of text.

<ods-format-options>
specifies formatting options for the ODS HTML5, PDF, and RTF destinations.

BCOLOR=*color*
specifies the background color of the title block.

You can optionally draw a box around a text string with the BOX= option. The BCOLOR= option controls the background color of the box. By default, the background color of the box is the same as the background color for the entire graph. You can specify colors using a number of different color-naming schemes.

Valid in Base SAS, ODS Graphics, SAS/GRAPH

Alias BC=

Restriction If you omit the BOX= option, the BCOLOR= option is ignored.

BOLD
specifies that the font weight is bold for the text string.

Valid in Base SAS, ODS Graphics

Default For titles, the default font weight is specified by the FONTWEIGHT attribute of the GraphTitleText style element in the current style.

Restriction The BOLD option is for template-based procedures only.

COLOR=*color*

specifies the color for the text, box, or line that follows it. You can specify colors using a number of different color-naming schemes.

The color stays in effect until another COLOR= specification is encountered. Change colors as often as you like. For example, the following code produces a title where the first word is red and the second word is blue:

```
title color=red "Total Sales" color=blue;
```

Valid in Base SAS. ODS Graphics, SAS/GRAPH

Alias C=

Default For titles, the default text color is specified by the COLOR attribute of the GraphTitleText style element in the current style.

See "BCOLOR=*color*" on page 132

Example "Customizing Titles and Footnotes By Using the Output Delivery System" in *SAS Global Statements: Reference*

FONT='*system-font*'

specifies the system font to use for the subsequent text string.

Valid in Base SAS, ODS Graphics, SAS/GRAPH

Alias F=

Default For titles, the default font is specified by the FONTFAMILY attribute of the GraphTitleText style element in the current style.

Note SAS/GRAPH software fonts such as SWISS cannot be used with statistical graphics procedures.

HEIGHT=*numeric-value<units>*

specifies the size of the text. You can also specify the unit of measurement. Without a unit of measure, the number becomes a relative size.

If you do not specify a unit, then the size of the text is approximately 12*numeric-value* points. For example, if you specify HEIGHT=2, then the text size is approximately 24 points.

numeric-value

<units>
is a nonnegative number.

Measurement Units

cm	Centimeters
in	Inches
pct or %	Percentage
pt	A printer's point

Valid in Base SAS, ODS Graphics, SAS/GRAPH

Alias H=

Default For the first title (title or title1), the default font size is specified by the FONTSIZE attribute of the GraphTitleText style element in the current style. For subsequent titles (titleN), the size is the FONSIZE attribute of the GraphLabelFont style attribute.

Style reference Height attribute of the GraphTitle1Text (TITLE1) and the GraphTitleText (TITLE2...n) elements

Example "Customizing Titles and Footnotes By Using the Output Delivery System" in *SAS Global Statements: Reference*

ITALIC

specifies that the font style is italic for the text string.

Valid in Base SAS, ODS Graphics

Default For titles, the default font style is specified by the FONTSTYLE attribute of the GraphTitleText style element in the current style.

JUSTIFY= LEFT | CENTER | RIGHT

specifies the alignment of the text string. You can specify one of the following values:

LEFT

aligns the text to the left.

Alias L

CENTER

aligns the text in the center.

Alias C

Note By default, titles are sometimes centered over the data area of the graph. However, when you explicitly specify center justification, titles are always centered over the graph area. All the text strings following JUSTIFY= are treated as a single string and are displayed as one line that is left-, right-, or center-aligned.

RIGHT

aligns the text to the right.

Alias R

Alias J=

Default CENTER

Notes The JUSTIFY= option affects all of the text strings that follow it in your TITLE statement. You can specify multiple alignments by using more than one JUSTIFY= option. For example, the following

code creates a title where the first string is aligned to the left and the second string is aligned to the right:

```
title justify=left "Example 2" justify=right "Graph 3";
```

You can use the JUSTIFY= option to produce multiple lines of text by repeating the JUSTIFY= option with the same value before the text string for each line. Multiple lines of text with the same justification are blocked together. For example, this TITLE statement produces a three-line title with each line right-justified:

```
title  justify=right "First Line"
     justify=right "Second Line"
     justify=right "Third Line";
```

You can get the same effect with three TITLE statements, each specifying JUSTIFY=RIGHT. You can produce a block of text by specifying the same justification for multiple text strings. You can then change the justification for an additional text string. The result is that the text is placed on the same line as the first string specified in the statement.

Using the JUSTIFY= option after one text string and before another can reset some options to their default values.

Example "Customizing Titles and Footnotes By Using the Output Delivery System" in *SAS Global Statements: Reference*

LINK='*URL*'

specifies a uniform resource locator (URL) to which a title links.

Valid in Base SAS, SAS/GRAPH

Tip The visual properties for LINK= always come from the current style.

UNDERLIN=*0-3*

specifies whether the subsequent text is underlined. UNDERLIN=0 indicates no underlining. Values 1, 2, and 3 indicate the same type of underlining.

Valid in SAS/GRAPH

Alias U=

See " ROTATE=degrees" in *SAS/GRAPH: Reference*

Note The defaults for how ODS renders the TITLE statement come from style elements relating to system titles in the current style. The TITLE statement syntax with *ods-format-options* is a way to override the settings provided by the current style. The current style varies according to the ODS destination. For more information about how to determine the current style, see "Understanding Styles, Style Elements, and Style Attributes" in *SAS Output Delivery System: Procedures Guide*. Also see "TEMPLATE Procedure: Creating a Style Template" in *SAS Output Delivery System: Procedures Guide*.

Tip You can specify these options by letter, word, or words by preceding each letter or word of *text-string* by the option. For example, this code makes the title "Red, White, and Blue" appear in different colors

```
title color=red "Red," color=white "White, and" color=blue "Blue";
```

<'text-string'>

is a text string with a maximum length if 255 bytes. You must enclose text strings in either single or double quotation marks. The text appears exactly as you enter it in the statement, including uppercase and lowercase characters and spaces. Titles automatically wrap to additional lines if necessary. If the length of the specified title is greater than the value of the LINESIZE option, the title is truncated to the line size.

To use single quotation marks or apostrophes within the title, you can do either of the following:

- use a pair of single quotation marks together:

```
title 'All''s Well That Ends Well';
```

- enclose the text in double quotation marks:

```
title "All's Well That Ends Well";
```

You can specify multiple text strings in TITLE statements. If you want each string to be separated by a blank, add a blank to the beginning of the second and subsequent text strings rather than at the end the preceding string, as in this example:

```
title color=red "Sales:" color=blue " 2020";
```

With fonts that support Unicode, you can specify special characters by specifying a hexadecimal constant in quotation marks. A trailing `x` identifies a string as a hexadecimal constant. You must designate the hexadecimal constant as a Unicode specification and escape it: `(*ESC*){Unicode '`*`hexadecimal-value`*`'x}`. For example:

```
title "Regression with Confidence Limits ((*ESC*){unicode '03B1'x}=.05)";
```

This statement produces the title, "Regression with Confidence Limits (α = .05)" because '03B1'x is the hexadecimal value for the lowercase Greek letter alpha in all Unicode fonts.

TIP You can instead specify lowercase Greek letters by name, and uppercase Greek letters by name_u: `(*ESC*){Unicode alpha}=.05` and `(*ESC*){Unicode alpha_u}=.05`.

For more information, see "Specifying Special Characters Using Character and Hexadecimal Codes" in *SAS/GRAPH: Reference*.

You can also customize titles by inserting BY variable lines (#BYLINE), BY variable values (#BYVAL), or BY variable names (#BYVAR), in titles that are specified in procedure steps. Embed the items in the specified title text string at the position where you want the substitution text to appear.

Note For SVG output, the value of the TITLE statement is used for the SVGTITLE option if the latter is not specified. If more than one TITLE statement is specified, the titles are concatenated, separated by a semicolon. The maximum number of characters for the SVGTITLE option is 256 characters.

Tips For compatibility with previous releases, SAS accepts some text without quotation marks. When writing new programs or updating existing programs, always enclose text in quotation marks.

You can use macro variables and macros to change the information in TITLE statements. The SAS macro facility resolves the macro variable. If the title is enclosed in double quotation marks (""), the text indicated is substituted into the title. If the title is enclosed in single quotation marks (''), the text is not substituted.

See For more information about including quotation marks as part of the title, see "SAS Constants in Expressions" in *SAS Programmer's Guide: Essentials*.

Details

Summary of Text Options

ods-format-options can be one or more of the following:

- appearance options:
  - BOLD
  - COLOR= *color*
  - FONT= "*system-font*"
  - HEIGHT= *numeric-value* <*units*>
  - ITALIC
- placement and spacing options:
  - JUSTIFY= LEFT | CENTER | RIGHT
  - LSPACE= *numeric-value* <*units*>
- boxing and drawing options:
  - BCOLOR= *color*
  - BOX= *numeric-value*
  - BSPACE= *numeric-value* <*units*>

The following options are not supported by the ODS Graphics procedures:

- ANGLE=
- BLANK=
- DRAW=

- LANGLE=
- LINK=
- MOVE=
- ROTATE=
- UNDERLIN=
- WRAP

Using TITLE Statement

A TITLE statement takes effect when the step or RUN group with which it is associated executes. After you specify a title for a line, it is used for all subsequent output until you cancel the title or define another title for that line. A TITLE statement for a given line cancels the previous TITLE statement for that line and for all lines with larger numbers.

Note: If the length of the specified title is greater than the value of the LINESIZE option, the title is truncated to the line size.

Differences When Using TITLE and FOOTNOTE Statements with the SGRENDER Procedure

Titles and footnotes behave differently for the SGRENDER procedure than for the other ODS Graphics procedures.

Normally, the title and footnote are displayed within the graph's border. However, when you render a Graph Template Language (GTL) graph using PROC SGRENDER, the title and footnote are displayed in the ODS document, outside of the graph's border. To add titles and footnotes to the graph, use the GTL ENTRYTITLE and ENTRYFOOTNOTE statements.

Substituting BY Line Values in a Text String

These options are available if a BY statement is in effect:

#BYLINE
substitutes the entire BY line without leading or trailing blanks for #BYLINE in the text string. The BY line uses the format *variable-name*=*value*.

#BYVAL*n* | #BYVAL(*BY-variable-name*)
substitutes the current value of the specified BY variable for #BYVAL in the text string. Specify the variable with one of these:

n
specifies a variable by its position in the BY statement. For example, #BYVAL2 specifies the second variable in the BY statement.

BY-variable-name
specifies a variable from the BY statement by its name. For example, #BYVAL(YEAR) specifies the BY variable, YEAR. *variable-name* is not case sensitive.

#BYVAR*n* | #BYVAR(*BY-variable-name*)
substitutes the name of the BY-variable or the label associated with the variable (whatever the BY line would normally display) for #BYVAR in the text string. Specify the variable with one of these:

n
specifies a variable by its position in the BY statement. For example, #BYVAR2 specifies the second variable in the BY statement.

BY-variable-name
specifies a variable from the BY statement by its name. For example, #BYVAR(SITES) specifies the BY variable, SITES. *Variable-name* is not case sensitive.

Note: A BY variable name displayed in a title or footnote is always in uppercase. If a label is used, then it appears as specified in the LABEL statement.

To use the #BYVAR and #BYVAL substitutions, insert the item in the text string at the position where you want the substitution text to appear. Both #BYVAR and #BYVAL specifications must be followed by a delimiting character. The character can be either a space or other non-alphanumeric character, such as a quotation mark. If no delimiting character is provided, then the specification is ignored and its text remains intact and is displayed with the rest of the string. To allow a #BYVAR or #BYVAL substitution to be followed immediately by other text, with no delimiter, use a trailing dot (as with macro variables). The trailing dot is not displayed in the resolved text. If you want a period to be displayed as the last character in the resolved text, use two dots after the #BYVAR or #BYVAL substitution.

The substitution for #BYVAR or #BYVAL does not occur in the following cases:

- if you use a #BYVAR or #BYVAL specification for a variable that is not named in the BY statement. For example, you might use #BYVAL2 when there is only one BY-variable or #BYVAL(ABC) when ABC is non-existent or is not a BY-variable.
- if there is no BY statement at all

For both cases, no error or warning message is issued. The option specification is displayed with the rest of the string. The graph continues to display a BY line at the top of the page unless you suppress it by using the NOBYLINE option in an OPTION statement.

Units of Measurement

Some of the options in the TITLE statement give you the option to specify the unit of measurement. The following table lists the units that are supported:

Table 6.9 *Measurement Units*

Unit	Description
CM	centimeters

Unit	Description
IN	inches
PCT or %	percentage
PT	point size, calculated at 100 dots per inch

The “HEIGHT=*numeric-value<units>*” on page 133 allows for the specification of a few more units than are mentioned in the previous table.

7

Common Procedure Statements

About the Common Procedure Statements ... **142**
See Also ... 142

ATTRIB Statement ... **142**
About the ATTRIB Statement ... 142
Procedures with Which ATTRIB Can Be Used ... 143
See Also ... 143

BY Statement ... **143**
About the BY Statement ... 143
Procedures with Which BY Can Be Used ... 144
Preparing Data for BY-Group Processing ... 144
Controlling BY Lines ... 145
See Also ... 145

FORMAT Statement ... **146**
About the FORMAT Statement ... 146
Procedures with Which FORMAT Can Be Used ... 146
Using the FORMAT Statement with ODS Graphics Procedures ... 147
See Also ... 150

LABEL Statement ... **150**
About the LABEL Statement ... 150
Procedures with Which LABEL Can Be Used ... 150
See Also ... 151

WHERE Statement ... **151**
About the WHERE Statement ... 151
Procedures with Which WHERE Can Be Used ... 151
See Also ... 152

About the Common Procedure Statements

The procedures can use some of the SAS language statements that you typically use with the Base SAS procedures or with the DATA step, such as LABEL, WHERE, and FORMAT.

The following statements can be used by any of the ODS Graphics procedures:

ATTRIB
: associates attributes with one or more data set variables.

BY
: processes your data by using one or more classification variables, and produces a separate graph for each unique combination of values.

FORMAT
: associates SAS formats or user-defined formats with variables.

LABEL
: associates descriptive labels with variables.

WHERE
: selects observations from SAS data sets that meet a particular condition.

See Also

- *SAS DATA Step Statements: Reference*

ATTRIB Statement

About the ATTRIB Statement

The ATTRIB statement associates attributes with one or more data set variables. It can be used in a PROC step or in a DATA step. However, the rules for usage in a PROC step differ from those in a DATA step. In a PROC step, you can use the ATTRIB statement to change attributes such as format and label for one or more variables for the duration of the PROC step. In some procedures, you can also

specify the informat. For more information about using the ATTRIB statement in a PROC step, see *Base SAS Procedures Guide*. In a DATA step, the ATTRIB statement permanently changes attributes such as format, informat, label, and length for one or more variables.

You can use the ATTRIB statement to replace individual attribute statements such as FORMAT, INFORMAT, LABEL, and LENGTH.

Procedures with Which ATTRIB Can Be Used

The following table lists the ODS Graphics procedures with which you can use the ATTRIB statement.

Table 7.1 *ODS Graphics Procedures That Support the ATTRIB Statement*

Procedure	Syntax Description
SGPANEL	"ATTRIB Statement" (p. 822)
SGPIE	"ATTRIB Statement" (p. 870)
SGPLOT	"ATTRIB Statement" (p. 1650)
SGRENDER	"ATTRIB Statement" (p. 1684)
SGSCATTER	"ATTRIB Statement" (p. 1763)

See Also

"ATTRIB Statement" in *SAS DATA Step Statements: Reference*

BY Statement

About the BY Statement

The BY statement divides the observations from an input data set into groups for processing. Each set of contiguous observations with the same value for a specified variable is called a *BY group*. A variable that defines BY groups is called a *BY*

variable and is the variable that is specified in the BY statement. When you use a BY statement, the graphics procedure performs the following operations:

- processes each group of observations independently
- generates a separate graph or output for each BY group
- automatically adds a heading called a *BY line* to each graph identifying the BY group represented in the graph

By default, the procedure expects the observations in the input data set to be sorted in ascending order of the BY variable values.

Procedures with Which BY Can Be Used

The following table lists the ODS Graphics procedures with which you can use the BY statement.

Table 7.2 *ODS Graphics Procedures That Support the BY Statement*

Procedure	Syntax Description
SGPANEL	"BY Statement" (p. 823)
SGPIE	"BY Statement" (p. 871)
SGPLOT	"BY Statement" (p. 1651)
SGRENDER	"BY Statement" (p. 1685)
SGSCATTER	"BY Statement" (p. 1764)

Preparing Data for BY-Group Processing

Unless you specify the NOTSORTED option, observations in the input data set must be in ascending numeric or alphabetic order. To prepare the data set, you can sort it with the SORT procedure using the same BY statement that you plan to use in the target procedure. Or you can create an appropriate index on the BY variables.

If the procedure encounters an observation that is out of the proper order, it issues an error message.

If you need to group data in some other order, you can still use BY-group processing. To do so, process the data so that observations are arranged in contiguous groups that have the same BY-variable values and specify the NOTSORTED option in the BY statement.

Controlling BY Lines

Understanding Default Behavior

By default, the BY statement prints a BY line above each graph that contains the variable name followed by an equal sign and the variable value. For example, if you specify BY SITE in the procedure, the default heading when the value of SITE is `London` would be SITE=London.

Suppressing the BY Line

To suppress the entire BY line, use the NOBYLINE option in an OPTION statement.

Suppressing the Name of the BY Variable

You can suppress the variable name and the equal sign in the heading and leave only the BY value. Use the LABEL statement to assign a null label ("00"X) to the BY variable. For example, this statement assigns a null label to the SITE variable:

```
label site="00"x;
```

You can also use the NOBYLINE option and then use the #BYVAL1 substitution in a TITLE statement:

```
options nobyline;
title #byval1;
```

See Also

"BY Statement" in *SAS DATA Step Statements: Reference*

FORMAT Statement

About the FORMAT Statement

The FORMAT statement associates formats with one or more variables in a PROC step or in a DATA step. You can specify one or more formats and associate each with one or more variables. Formats can be standard SAS formats or user-defined formats that were created using the FORMAT procedure. When used in a PROC step, the formats are applied for the duration of the PROC step. When specified in a DATA step, the formats are permanently applied to the variables.

Procedures with Which FORMAT Can Be Used

The following table lists the ODS Graphics procedures with which you can use the FORMAT statement.

Table 7.3 *ODS Graphics Procedures That Support the FORMAT Statement*

Procedure	FORMAT Statement Syntax Description
SGPANEL	"FORMAT Statement" (p. 824)
SGPIE	"FORMAT Statement" (p. 872)
SGPLOT	"FORMAT Statement" (p. 1652)
SGRENDER	"FORMAT Statement" (p. 1686)
SGSCATTER	"FORMAT Statement" (p. 1765)

Using the FORMAT Statement with ODS Graphics Procedures

Associating Formats with Variables

The FORMAT statement can be used in an SGPLOT, SGSCATTER, SGPANEL, or SGRENDER procedure step to associate SAS formats and user-defined formats with variables. Also, several procedure statement options can be used to associate a format with a variable or computed value. Examples include the TICKVALUEFORMAT= option for formatting axis tick values, and the TIPFORMAT= option for formatting data tips. For information about SAS formats, see *SAS Formats and Informats: Reference*. For information about user-defined formats, see "FORMAT Procedure" in *Base SAS Procedures Guide*.

Not all SAS formats and user-defined formats are supported with the SGPLOT, SGSCATTER, SGPANEL, and SGRENDER procedures. See "User-Defined Formats That Are Not Supported" in *SAS Graph Template Language: Reference*. When an unsupported format is encountered, a note similar to the following is written to the SAS log:

```
TICKVALUEFORMAT=bestx. is invalid. The format is invalid or unsupported.
 The default will be used.
```

Using Locale-Sensitive Decimal Separators with Numeric Formats

System option NLDECSEPARATOR= is not supported by ODS Graphics. When system option NLDECSEPARATOR is in effect, syntax errors or null debug errors might result when using a procedure enabled by ODS Graphics. For locale-sensitive decimal separators in your graph's numeric values, specify system option NONLDECSEPARATOR, and then use appropriate NL formats to format your numeric variables. See "Dictionary of Formats for NLS" in *SAS National Language Support (NLS): Reference Guide*.

SAS Formats That Are Not Supported

Unsupported Numeric Formats

The following numeric formats are not supported in ODS Graphics:

BESTD	BESTX	D	FLOAT	FRACT
FREE	IB	IBR	IEEE	IEEER
ODDSR	PCPIB	PD	PIB	PIBR
PK	RB	SSN	WORDF	WORDS
Z	ZD			

Unsupported Date and Time Formats Related to ISO 8601

The following date and time formats are not supported in ODS Graphics:

$N8601B	$N8601BA	$N8601E	$N8601EA	$N8601EH
$N8601EX	$N8601H	$N8601X	B8601DA	B8601DN
B8601DT	B8601DZ	B8601LZ	B8601TM	B8601TZ
E8601DA	E8601DN	E8601DT	E8601DZ	E8601LZ
E8601TM	E8601TZ	IS8601DA	IS8601DN	IS8601DT
IS8601DZ	IS8601LZ	IS8601TM	IS8601TZ	

Other Unsupported Date and Time Formats

The following date and time formats are not supported in ODS Graphics:

HDATE	HEBDATE	JDATEMDW	JDATEMNW	JDATEWK
JDATEYDW	JDATEYM	JDATEYMD	JDATEYMW	JDATEYT
JDATEYTW	JNENGO	JNENGOT	JNENGOTW	JNENGOW
JTIMEH	JTIMEHM	JTIMEHMS	JTIMEHW	JTIMEMW
JTIMESW	MDYAMPM	MINGUO	NENGO	NLDATEYQ
NLDATEYR	NLDATEYW	NLDATMYQ	NLDATMYR	NLDATMYW
NLSTRMON	NLSTRQTR	NLSTRWK	PDJULG	PDJULI
TWMDY	XYYMMDD	YYQZ		

Unsupported Currency Formats

The following currency formats are not supported in ODS Graphics:

EURFRATS	EURFRBEF	EURFRCHF	EURFRCZK	EURFRDEM
EURFRDKK	EURFRESP	EURFRFIM	EURFRFRF	EURFRGBP
EURFRGRD	EURFRHUF	EURFRIEP	EURFRITL	EURFRLUF
EURFRNLG	EURFRNOK	EURFRPLZ	EURFRPTE	EURFRROL
EURFRRUR	EURFRSEK	EURFRSIT	EURFRTRL	EURFRYUD
EURTOATS	EURTOBEF	EURTOCHF	EURTOCZK	EURTODEM
EURTODKK	EURTOESP	EURTOFIM	EURTOFRF	EURTOGBP
EURTOGRD	EURTOHUF	EURTOIEP	EURTOITL	EURTOLUF
EURTONLG	EURTONOK	EURTOPLZ	EURTOPTE	EURTOROL
EURTORUR	EURTOSEK	EURTOSIT	EURTOTRL	EURTOYUD

User-Defined Formats That Are Not Supported

Picture Formats with Date, Time, or Datetime Directives

ODS Graphics does not support user-defined picture formats that specify date, time, or datetime directives such as %a, %W, %H, and so on. Using a picture format with directives produces unexpected results in ODS Graphics. To work around this limitation, apply the picture format to a new column in your data, and then plot the new preformatted column instead. Here is an example that creates column NewDate by formatting the Date column as MM/YYYY.

```
/* Create picture format MonYear for MM/YYYY date */
proc format;
  picture monyear (min=7)
    low-high='%0m/%Y' (datatype=date);
run;

/* Add preformatted date column NewDate to the data */
data stocks;
  set sashelp.stocks
    (where=(year(date) eq 2001 and stock eq "IBM"));
  newdate = put(date, monyear.);
  label newdate="Date: MM/YYYY";
run;
```

The NewDate column can be used in ODS Graphics plot statements instead of the Date column.

For information about the PICTURE statement in the FORMAT procedure, see *Base SAS Procedures Guide*.

Unicode Values with a User-Defined ODS Escape Character

ODS Graphics supports Unicode values in user-defined formats only if they are preceded by the (*ESC*) escape sequence as shown in the following example.

```
"(*ESC*){unicode beta}"
```

ODS Graphics does not support the use of a user-defined ODS escape character to escape Unicode values in user-defined formats.

For an example of how to use Unicode values in user-defined formats with ODS Graphics, see "Formatting the Tick Values on a Discrete Axis" in *SAS Graph Template Language: User's Guide*.

See Also

"FORMAT Statement" in *SAS DATA Step Statements: Reference*

LABEL Statement

About the LABEL Statement

The LABEL statement applies descriptive labels to variables in a data set. You can specify one or more variable and label pairs in a LABEL statement. The LABEL statement can be used in a PROC step or in a DATA step. When used in a PROC step, the labels are applied for the duration of the PROC step. When used in a DATA step, the labels are permanently applied.

Procedures with Which LABEL Can Be Used

The following table lists the ODS Graphics procedures with which you can use the LABEL statement.

Table 7.4 *ODS Graphics Procedures That Support the LABEL Statement*

Procedure	LABEL Statement Syntax Description
SGPANEL	"LABEL Statement" (p. 826)
SGPIE	"LABEL Statement" (p. 874)
SGPLOT	"LABEL Statement" (p. 1653)
SGRENDER	"LABEL Statement" (p. 1688)
SGSCATTER	"LABEL Statement" (p. 1766)

See Also

"LABEL Statement" in *SAS DATA Step Statements: Reference*

WHERE Statement

About the WHERE Statement

The WHERE statement selects observations that meet a specific condition. The condition is specified as one or more arithmetic or logical expressions, which generally consist of a sequence of operands and operators. A WHERE statement can be used to improve efficiency by eliminating observations that are not relevant. It can be used in a PROC step or in a DATA step.

Procedures with Which WHERE Can Be Used

The following table lists the ODS Graphics procedures with which you can use the WHERE statement.

Table 7.5 *ODS Graphics Procedures That Support the WHERE Statement*

Procedure	Syntax Description
SGPANEL	"WHERE Statement" (p. 828)
SGPIE	"WHERE Statement" (p. 876)
SGPLOT	"WHERE Statement" (p. 1655)
SGRENDER	"WHERE Statement" (p. 1690)
SGSCATTER	"WHERE Statement" (p. 1768)

See Also

"WHERE Statement" in *SAS DATA Step Statements: Reference*

PART 4

The Procedures

Chapter 8
SGPANEL Procedure **155**

Chapter 9
SGPIE Procedure: Preproduction **839**

Chapter 10
SGPLOT Procedure **883**

Chapter 11
SGRENDER Procedure **1677**

Chapter 12
SGSCATTER Procedure **1699**

Chapter 13
Common Concepts **1777**

Chapter 14
Commonly Used Attribute Options **1827**

8

SGPANEL Procedure

Overview: SGPANEL Procedure ***155***
What Does the SGPANEL Procedure Do? 156

Concepts: SGPANEL Procedure ***158***
Panel Creation 158
Concepts in Common with the SGPLOT Procedure 159

Syntax: SGPANEL Procedure ***160***
PROC SGPANEL Statement 162
PANELBY Statement 168
STYLEATTRS Statement 180
BAND Statement 188
BLOCK Statement 198
BUBBLE Statement 207
DENSITY Statement 221
DOT Statement 228
DROPLINE Statement 247
ELLIPSE Statement 252
ELLIPSEPARM Statement 260
FRINGE Statement 267
GRADLEGEND Statement 272
HBAR Statement 276
HBARBASIC Statement 302
HBARPARM Statement 322
HBOX Statement 343
HEATMAP Statement 363
HEATMAPPARM Statement 371
HIGHLOW Statement 380
HISTOGRAM Statement 397
HLINE Statement 407
INSET Statement 426
KEYLEGEND Statement 430
LEGENDITEM Statement 438
LINEPARM Statement 445
LOESS Statement 451
NEEDLE Statement 465
PBSPLINE Statement 476
POLYGON Statement 492
REFLINE Statement 513
REG Statement 520

SCATTER Statement 535
SERIES Statement 553
SPLINE Statement 574
STEP Statement 589
SYMBOLCHAR Statement 606
SYMBOLIMAGE Statement 613
TEXT Statement 618
VBAR Statement 634
VBARBASIC Statement 661
VBARPARM Statement 681
VBOX Statement 704
VECTOR Statement 724
VLINE Statement 734
COLAXIS Statement 754
COLAXISTABLE Statement 776
ROWAXIS Statement 788
ROWAXISTABLE Statement 809
ATTRIB Statement 822
BY Statement 823
FORMAT Statement 824
LABEL Statement 826
WHERE Statement 828

Examples: SGPANEL Procedure ***829***
About the SGPANEL Procedure Examples 829
Example 1: Creating a Panel of Graph Cells with Histograms and Density Plots . . 832
Example 2: Creating a Panel of Regression Curves 833
Example 3: Creating a Panel of Bar Charts 835
Example 4: Creating a Panel of Line Charts 836

Overview: SGPANEL Procedure

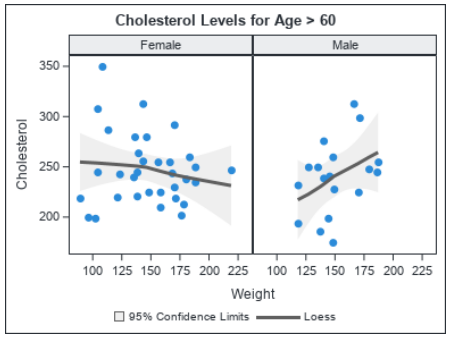
What Does the SGPANEL Procedure Do?

The SGPANEL procedure creates a panel of graph cells for the values of one or more classification variables. For example, if a data set contains three variables (A, B and C) and you want to compare the scatter plots of B*C for each value of A, then you can use the SGPANEL procedure to create this panel. The SGPANEL procedure creates a layout for you automatically and splits the panel into multiple graphs if necessary.

The SGPANEL procedure can create a wide variety of plot types, and overlay multiple plots together in each graph cell in the panel. It can also produce several types of layout.

Here are some examples of panels that the SGPANEL procedure can create.

Table 8.1 *Examples of Panels That Can Be Generated by the SGPANEL Procedure*

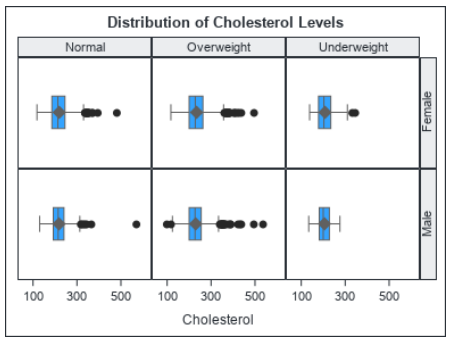


The following code creates a panel of loess curves:

```
title1 "Cholesterol Levels for Age > 60";
proc sgpanel data=sashelp.heart(
    where=(AgeAtStart > 60)) ;
  panelby sex / novarname;
  loess x=weight y=cholesterol / clm;
run;
title1;
```

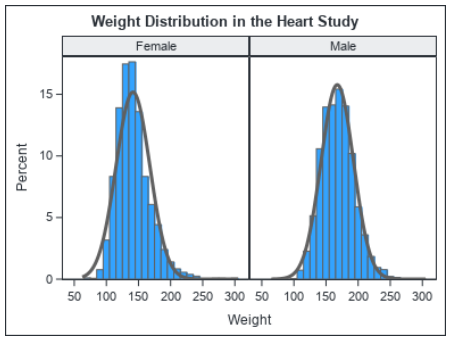
The following code creates a panel of vertical bar charts:

```
title1 "Product Sales";
proc sgpanel data=sashelp.prdsale;
  panelby quarter;
  rowaxis label="Sales";
  vbar product / response=predict stat=mean
                 transparency=0.3;
  vbar product / response=actual stat=mean
                 barwidth=0.5 transparency=0.3;
 run;
title1;
```



The following code creates a panel of box plots in a lattice layout:

```
title1 "Distribution of Cholesterol Levels";
proc sgpanel data=sashelp.heart;
  panelby weight_status sex / layout=lattice
                              novarname;
  hbox cholesterol;
run;
title1;
```



The following code creates a panel of cells with a histogram and a normal density curve:

```
Title1 "Weight Distribution in the Heart Study";
proc sgpanel data=sashelp.heart noautolegend;
  panelby sex / novarname;
  histogram weight;
  density weight;
run;
Title1;
```

Concepts: SGPANEL Procedure

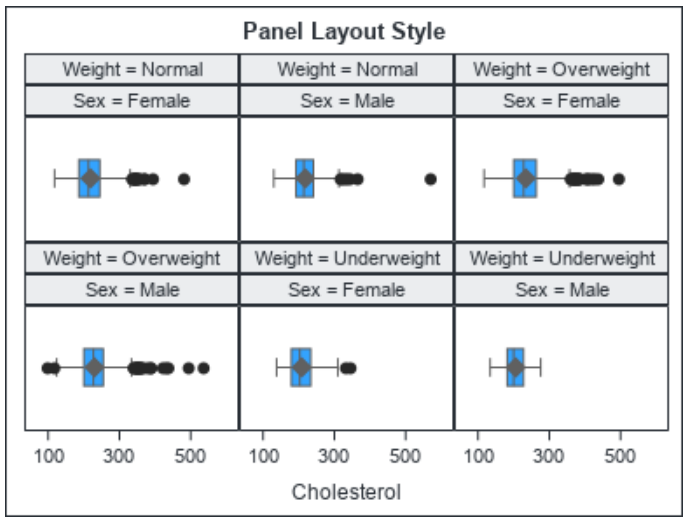
Panel Creation

The SGPANEL procedure has a required PANELBY statement that is used to define the classifier variables for the panel. This statement must be specified before any plot, axis, or legend statement or else an error occurs. You can use options in the PANELBY statement to control the attributes of the panel. For example, you can use the COLUMNS= option to specify the number of columns in the panel.

SGPANEL can use four different layouts, which are specified by the LAYOUT= option in the PANELBY statement. The layout determines how your classifier variables are used to create the panel, and also affects the number of classifier variables that you can specify.

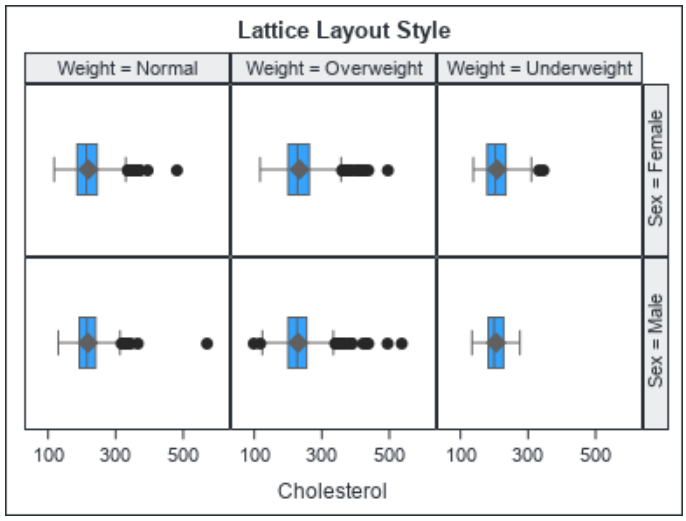
The default layout is PANEL. With this layout, you can specify any number of classifier variables. The graph cells in the panel are arranged automatically, and the classifier values are displayed above each graph cell in the panel. When you specify multiple classifier variables, the order of the classifier variables determines how the graph cells are sorted.

Figure 8.1 *Example of the PANEL Layout*



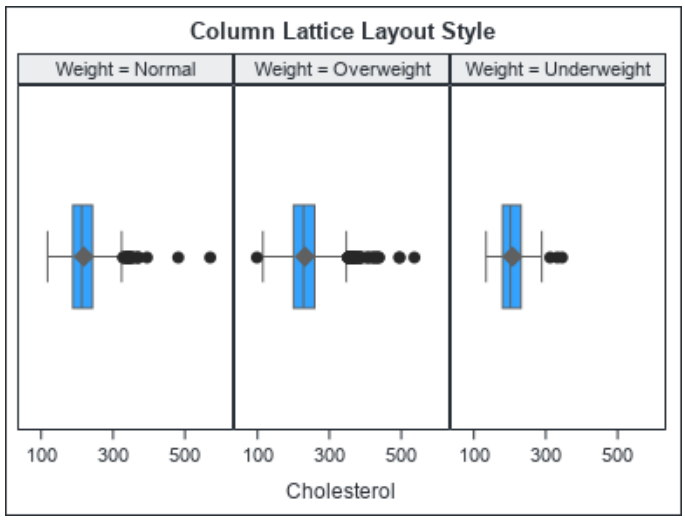
Another layout is called LATTICE. This layout requires exactly two classifier variables. The values of the first variable are assigned as columns, and the values of the second variable are assigned as rows. The classifier values are displayed above the columns and to the right side of the rows.

Figure 8.2 *Example of the LATTICE Layout*



Two additional layouts are available, which are called COLUMNLATTICE and ROWLATTICE. These layouts require exactly one classifier variable. The values of the classifier variable are assigned as cells in a single row or column.

Figure 8.3 *Example of the COLUMNLATTICE Layout*



If you have a large number of classifier variables, then the best method for creating a panel is to choose one or two classifiers for the PANELBY statement. Then specify the remaining variables in a BY statement. This method maximizes the space for the plots and generates results that are easier to interpret.

Concepts in Common with the SGPLOT Procedure

The following topics are located in the Chapter 13, “Common Concepts,” on page 1777 section. These topics describe concepts that are similar between the SGPANEL and SGPLOT procedures.

Table 8.2 Common Concepts

"Plot Type Compatibility" (p. 1779)	explains which types of plots can be used together in a graph
"Plot Axes" (p. 1784)	describes the axis types that are supported by the procedure
"Legends" (p. 1790)	explains how legends are created automatically, and how to create customized legends
"Automatic Differentiation of Visual Attributes" (p. 1797)	explains when different style attributes are automatically assigned to plots, and how to force the procedure to use different style attributes if they are not automatically assigned
"Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" (p. 1802)	describes how to split the text for data labels, curve labels, and axis tick mark values when there is not enough room to display the text normally
"Marker Fills and Outlines" (p. 1805)	describes how you can change the appearance of both the marker fill and its outline for graphs that contain markers.
"Color-Naming Schemes" (p. 1808)	describes the color-naming schemes that are supported in ODS Graphics and how to specify colors in your programs.
"Fonts" (p. 1819)	describes how to determine the fonts that are available for your graphs, how to specify fonts in your programs, and how to register your own fonts in SAS.

Syntax: SGPANEL Procedure

Requirement: The PANELBY statement and at least one plot statement are required.

```
PROC SGPANEL <options>;
  PANELBY variable(s) </options>;
  STYLEATTRS </options>
  BAND X=variable Y=variable
    UPPER=numeric-value | numeric-variable
    LOWER=numeric-value | numeric-variable </options>;
  BLOCK X=category-variable BLOCK=block-variable </options>;
  BUBBLE X=variable Y=variable SIZE=numeric-variable </options>;
  DENSITY response-variable </options>;
```

DOT *category-variable* <*options*>;

DROPLINE X=*variable* | *x-axis-value*
Y=*variable* | *y-axis-value* </*options*>;

ELLIPSE X=*numeric-variable* Y=*numeric-variable* </*options*>;

ELLIPSEPARM SEMIMAJOR=*numeric-value* | *numeric-variable*
SEMIMINOR=*numeric-value* | *numeric-variable* </*options*>

FRINGE *numeric-variable* </*options*>;

GRADLEGEND <"*name*"> </*options*>;

HBAR *category-variable* </*options*>;

HBARBASIC *category-variable* </*options*>;

HBARPARM CATEGORY=*category-variable* RESPONSE=*numeric-variable* </*options*>;

HBOX *analysis-variable* </*options*>;

HEATMAP X=*variable* Y=*variable* </*options*>;

HEATMAPPARM X=*variable* Y=*variable*
COLORGROUP=*variable* | COLORRESPONSE=*numeric-variable* </*options*>;

HIGHLOW X=*variable* | Y=*variable*
HIGH=*numeric-variable* LOW=*numeric-variable*</option(s)>;

HISTOGRAM *response-variable* </*options*>;

HLINE *category-variable* </*options*>;

INSET*variable* <...*variable-n*> </*options*>

KEYLEGEND <"*name-1*" ... "*name-n*"> </*options*>;

LEGENDITEM TYPE=*type* NAME="*text-string*" </*options*>;

LINEPARM X=*numeric-value* | *numeric-variable*
Y=*numeric-value* | *numeric-variable*
SLOPE=*numeric-value* | *numeric-variable*
</option(s)>;

LOESS X=*numeric-variable* Y=*numeric-variable* </*options*>;

NEEDLE X=*variable* Y=*numeric-variable* </*options*>;

PBSPLINE X=*numeric-variable* Y=*numeric-variable* </*options*>;

POLYGON X=*x-variable* Y=*y-variable* ID=*id-variable* </*options*>;

REFLINE *value(s)* </*options*>;

REG X=*numeric-variable* Y=*numeric-variable* </*options*>;

SCATTER X=*variable* Y=*variable* </*options*>;

SERIES X=*variable* Y=*variable* </*options*>;

SPLINE X=*variable* Y=*variable* </*options*>;

STEP X=*variable* Y=*variable* </*options*>;

SYMBOLCHAR NAME=*identifier*
CHAR="*hex-string*" | *keyword* </*options*>;

SYMBOLIMAGENAME=*identifier* IMAGE="*image-file-specification*" </*options*>;

TEXT X=*variable* Y=*variable* TEXT=*variable* </*options*>;

VBAR *category-variable* </*options*>;

VBARBASIC *category-variable* </*options*>;

VBARPARM CATEGORY=*category-variable* RESPONSE=*numeric-variable* </*options*>;

VBOX *analysis-variable* </*options*>;

VECTOR X=*numeric-variable* Y=*numeric-variable* </*options*>;

VLINE *category-variable* </*options*>;

COLAXIS <*options*>;

COLAXISTABLE *variable* < ...*variable-n*> </*options*>;

ROWAXIS <*options*>;

ROWAXISTABLE *variable* < ...*variable-n*> </*options*>;

ATTRIB *variable-list(s) attribute-list(s)*;

BY *variable-1*<...*variable-N*><*options*>;

FORMAT *variable-1*<...*variable-N*<*format-1*>><*options*>;

LABEL *variable-1*=*label-1*<...*variable-n*=*label-n*>;

WHERE *where-expression*;

PROC SGPANEL Statement

Creates a panel of plots from input data.

Requirement: An input data set is required.

Syntax

PROC SGPANEL <*options*> ;

Summary of Optional Arguments

ASPECT=*positive-number*
: specifies the aspect ratio of the plot's wall area.

CYCLEATTRS

NOCYCLEATTRS
: specifies whether plots are drawn with unique attributes in the graph.

DATA=*input-data-set*
: specifies the SAS data set that contains the variables to process.

DATTRMAP=*discrete-attribute-map-data-set*
: specifies the discrete attribute map data set that you want to use with the SGSCATTER procedure.

DESCRIPTION="*text-string*"
: specifies a description for the output image.

NOAUTOLEGEND
: disables automatic legends from being generated.

NOSUBPIXEL

SUBPIXEL
: specifies whether subpixel rendering should be used for rendering line plots and bar charts.

OPAQUE

NOOPAQUE
: specifies whether the graph background is opaque or transparent.

PAD=*dimension<units>* | (*pad-options*)
: specifies the amount of extra space to add to the left, right, top, and bottom of the graph.

PCTLEVEL=BY | CELL | GRAPH | GROUP
: specifies the scope of graph data that is calculated in percentages.

PCTNDEC=*numeric-value*
: specifies the number of decimal spaces to be used to calculate the percent values.

RATTRMAP=*range-attribute-map-data-set*
: specifies the range attribute map data set that you want to use with the procedure.

SGANNO=*annotation-data-set*
: specifies the SG annotation data set that you want to use.

Optional Arguments

ASPECT=*positive-number*
: specifies the aspect ratio of the plot's wall area. The ratio is expressed as a positive decimal fraction representing wall height divided by wall width. For example, 0.75 is a 3/4 aspect ratio, and 1.0 is a square aspect ratio.

  Small numbers, such as 0.01, produce a short, wide rectangular area. Larger numbers yield a taller, narrower rectangular area.

  Default The wall area is sized to the maximum area that can fill the available space.

CYCLEATTRS | NOCYCLEATTRS
: specifies whether plots are drawn with unique attributes in the graph. By default, the SGPANEL procedure automatically assigns unique attributes in many situations, depending on the types of plots that you specify. If the plots do not have unique attributes by default, then the CYCLEATTRS option assigns unique attributes to each plot in the graph. The NOCYCLEATTRS option prevents the procedure from assigning unique attributes.

  For example, if you specify the CYCLEATTRS option and you create a graph with a SERIES statement and a SCATTER statement, then the two plots have different colors.

  If you specify the NOCYCLEATTRS option, then plots have the same attributes unless you specify appearance options such as the LINEATTRS= option.

See "Automatic Differentiation of Visual Attributes" on page 1797

DATA=*input-data-set*
specifies the SAS data set that contains the variables to process. By default, the procedure uses the most recently created SAS data set.

DATTRMAP=*discrete-attribute-map-data-set*
specifies the discrete attribute map data set that you want to use with the SGSCATTER procedure. You specify this option only if you are using a discrete attribute map to control visual attributes of the graph.

Restriction The procedures do not support an attribute map that is stored in a table. Only standard SAS data sets are supported for attribute map data.

Requirement The values in the DATTRMAP data set must be sorted by ID. If they are not, only the first value is found.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

DESCRIPTION="*text-string*"
specifies a description for the output image. The description identifies the image in the following locations:

- the alternate text for the image in HTML output
- the table of contents that is created by the CONTENTS option in an ODS statement

The default description is "The SGPANEL Procedure".

Alias DES=

Note The name of the output image is specified by the IMAGENAME= option in the ODS GRAPHICS statement.

Tip You can disable the alternate text in HTML output by specifying an empty string. That is, `DESCRIPTION=""`.

NOAUTOLEGEND
disables automatic legends from being generated. By default, legends are created automatically for some plots, depending on their content. This option has no effect if you specify a KEYLEGEND statement.

NOSUBPIXEL | SUBPIXEL
specifies whether subpixel rendering should be used for rendering line plots and bar charts. Subpixel rendering produces smoother curves and more precise bar spacing.

Defaults When neither option is specified, the system applies SUBPIXEL when it makes sense for the graph.

Subpixel rendering is always enabled for vector-graphics output.

Requirement Antialiasing must be enabled for the SUBPIXEL option to have any effect. Antialiasing is enabled by default. To re-enable antialiasing, use the ANTIALIAS=ON option in the ODS GRAPHICS statement.

Interaction If the SUBPIXEL option is explicitly set in the ODS GRAPHICS statement, that setting is used.

Tip For a large amount of data, antialiasing is disabled when the number of observations exceeds the default maximum of 4000 observations. In that case, subpixel rendering is also disabled. To increase the maximum, use the ANTIALIASMAX= option in the ODS GRAPHICS statement.

See "Subpixel Rendering" on page 1857

"ODS GRAPHICS Statement" on page 102 for information about the ANTIALIAS= and ANTIALIASMAX= options.

OPAQUE | NOOPAQUE
specifies whether the graph background is opaque or transparent.

Default OPAQUE

Restriction The following output formats support transparent background (NOOPAQUE): EMF, PDF, PNG, PS, and SVG, with the following exception. The PS format does not support transparent background when your output format is not vector graphics (that is, your output renders as an image due to some graph feature or you used an OUTPUTFMT= override to an image format, including PNG).

Interaction When NOOPAQUE is specified, the background color is not used.

PAD=*dimension<units>* | (*pad-options*)
specifies the amount of extra space to add to the left, right, top, and bottom of the graph. The extra space can help make the spacing more aesthetic, or provide blank space for custom annotations.

You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see Chapter 20, "Annotating ODS Graphics," on page 1935.

This option creates margins around the graph for company logos, annotated notes, and so on. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Use pad options to create non-uniform padding. Edges that are not assigned padding are padded with the default amount.

pad-options can be one or more of the following:

LEFT= *numeric-value<units>*
specifies the amount of extra space to add to the left edge.

RIGHT= *numeric-value<units>*
specifies the amount of extra space to add to the right edge.

TOP= *numeric-value<units>*
specifies the amount of extra space to add to the top edge.

BOTTOM= *numeric-value<units>*
specifies the amount of extra space to add to the bottom edge.

PCTLEVEL=BY | CELL | GRAPH | GROUP
specifies the scope of graph data that is calculated in percentages. When you calculate percentages using the STAT=PERCENT option, the calculation can be performed at different levels in the graph. The percentages within the selected level attempt to round up to 100%.

BY
the percentages within each BY-group round up to 100%

Interaction For this value to take effect, a BY-group must be specified in the procedure.

CELL
the percentages within each panel cell round up to 100%.

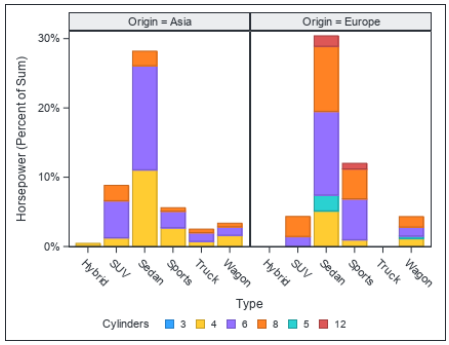
GRAPH
the percentages across the entire graph round up to 100%

GROUP
the percentages across groups within a category round up to 100%.

Interaction For this value to take effect, the GROUP= option must be specified in the plot statement.

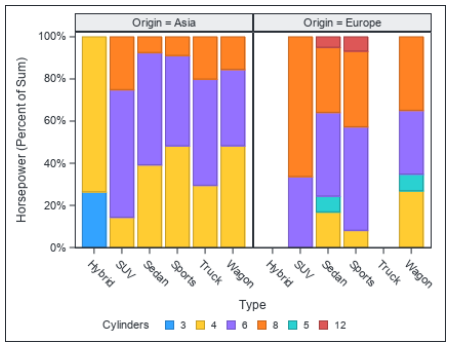
In the following examples, the first example specifies a value of GRAPH (the default), and the second example specifies a value of GROUP.

```
ods graphics on / reset=all;
ods graphics on / width=4.5in;
```



PCTLEVEL=GRAPH

```
proc sgpanel data=sashelp.cars pctlevel=graph;
panelby origin;
where origin in("Asia" "Europe");
vbar type / response=horsepower stat=percent
    group=cylinders;
run;
```



PCTLEVEL=GROUP

```
proc sgpanel data=sashelp.cars pctlevel=group;
panelby origin;
where origin in("Asia" "Europe");
vbar type / response=horsepower stat=percent
    group=cylinders;
run;
```

In the graph that specifies PCTLEVEL=GRAPH, the bars collectively add up to 100%. Each bar represents a fraction of the total.

In the graph that specifies PCTLEVEL=GROUP, each bar adds up to 100% of the category data represented by the bar. In this case, the categories are age.

Default GRAPH

Interaction For this option to take effect, STAT=PERCENT must be specified for a plot in the procedure.

Tip You can use the PCTNDEC= option in the SGPANEL procedure statement to control the number of decimals to be used when calculating the percent values. The default value is 1.

PCTNDEC=*numeric-value*

specifies the number of decimal spaces to be used to calculate the percent values.

Default The default number of decimals is based on the magnitude of the largest percentage value.

10% to 100% = 1	.1% to < 1% = 3
1% to < 10% = 2	< .1% = 4

Interaction For this option to take effect, STAT=PERCENT must be specified for a plot in the procedure.

Tip You can use the PCTLEVEL= option in the PROC SGPANEL statement to control the scope of the percent calculations.

RATTRMAP=*range-attribute-map-data-set*

specifies the range attribute map data set that you want to use with the procedure. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

Restriction The procedures do not support an attribute map that is stored in a table. Only SAS data sets are supported for attribute map data.

Requirement The values in the RATTRMAP data set must be sorted by ID. If they are not, only the first value is found.

See Chapter 19, “Using Range Attribute Maps,” on page 1921

“Overview of Attribute Maps” on page 1897

SGANNO=*annotation-data-set*
specifies the SG annotation data set that you want to use. You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see Chapter 20, “Annotating ODS Graphics,” on page 1935.

Note: The procedures do not support annotation data that is stored in a table. Only standard SAS data sets are supported for annotation data.

PANELBY Statement

Specifies one or more classification variables for the panel, the layout type, and other options for the panel.

Restrictions: The number of cells in a panel cannot exceed the maximum specified by the ODS GRAPHICS statement option PANELCELLMAX-. The default is 1,000. If the limit is exceeded, the panel is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, you can increase PANELCELLMAX= to the value suggested in the note.

The number of cells in the graph cannot exceed the maximum specified by the ODS GRAPHICS statement option TOTALCELLMAX=. The default is 2,000. If the limit is exceeded, the graph is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, you can increase TOTALCELLMAX= to the value suggested in the note.

Syntax

PANELBY *variable(s)* </*options*>;

Summary of Optional Arguments

BORDER

NOBORDER
specifies whether borders are displayed around each cell in the panel.

COLHEADERPOS=TOP | BOTTOM | BOTH
specifies the location of the column headings in the panel.

COLUMNS=*n*
specifies the number of columns in the panel.

HEADERATTRS=*style-element*<(***options***)> | (*options*)

specifies the appearance of the column headings.

HEADERBACKCOLOR=*color*
specifies a background color for the cell headers.

HEADERPACK

HEADERPACK=*integer-value-count*
specifies whether the header cells are consolidated into a comma-separated list in order to save space.

HEADERSEP="*string*"
specifies one or more characters to place between each value in packed cell headers.

LAYOUT=LATTICE | PANEL | COLUMNLATTICE | ROWLATTICE
specifies the type of layout that is used for the panel.

MISSING
processes missing values as a valid classification value and creates cells for it.

NOHEADER
hides the cell headings.

NOHEADERBORDER
removes the border from the cell headings.

NOVARNAME
removes the variable names from the cell headings of a panel layout, or from the row and column headings of a lattice layout.

NOWALL
turns off the display of the graph wall.

ONEPANEL
places the entire panel in a single output image.

PROPORTIONAL
changes the size of the cells in proportion to the content of the discrete axis.

ROWHEADERPOS=LEFT | RIGHT | BOTH
specifies the location of the row headings in the panel.

ROWS=*n*
specifies the number of rows in the panel.

SKIPEMPTYCELLS
specifies whether the external axes skip the empty cells in a partially filled grid.

SORT=*sort-option* | (*sort-option-1*...*sort-option-n*)
specifies the sort order of the panel cells.

SPACING=*n*
specifies the number of pixels between the rows and columns in the panel.

SPARSE
enables the SGPANEL procedure to create empty cells for crossings of the classification variables that are not present in the input data set.

START=TOPLEFT | BOTTOMLEFT
specifies whether the first cell in the panel is placed at the upper left corner or the lower left corner.

UNISCALE=COLUMN | ROW | ALL
scales the shared axes in the panel to be identical.

Required Argument

variable(s)
specifies one or more classification variables for the panel.

Optional Arguments

BORDER | NOBORDER
specifies whether borders are displayed around each cell in the panel. BORDER adds the borders. NOBORDER removes the borders. Depending on the current ODS style, the borders might be present by default.

COLHEADERPOS=TOP | BOTTOM | BOTH
specifies the location of the column headings in the panel. Specify one of the following values:

TOP
places column headings at the top of each column.

BOTTOM
places column headings at the bottom of each column.

BOTH
places column headings at the top and bottom of each column.

Default TOP

Interaction This option has no effect if the panel uses the PANEL layout.

COLUMNS=*n*
specifies the number of columns in the panel. By default, the number of columns is determined automatically based on the number of classifier values and the layout type. The SGPANEL procedure automatically splits the panel into multiple panel images (pages) as needed when your panel contains a large number of cells.

Specifying COLUMNS= enables you to control panel size. When the number of classification levels exceeds the number of cells in the panel, additional panels (images) are created. The last panel might be partially populated.

If you specify COLUMNS= without specifying a value for ROWS= (by default, ROWS=1), then one of the following occurs:

- The cells fill each row from left to right up to the requested number of columns before going to the next row. Each panel can have a maximum of three rows. For example, if you specify two columns, and the classifier has six values, the result is a 2-column by 3-row panel. If you specify two columns, and the classifier has eight values, the result is two panels, and each panel has two rows and two columns.
- If the number of classification variables is not divisible by the columns value, then the procedure simply creates multiple single-row panels. Each panel

has the specified number of columns. In this scenario, the last panel is only partially populated.

However, if you also specify ROWS= with a value greater than one, then the procedure uses your specified values to determine the number of panels. Specifying both ROWS= and COLUMNS= enables you to define the precise layout, including how many cells are in a panel.

HEADERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the column headings. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Examples
```
HEADERATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
HEADERATTRS=GraphLabelText
```

HEADERBACKCOLOR=*color*
specifies a background color for the cell headers. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

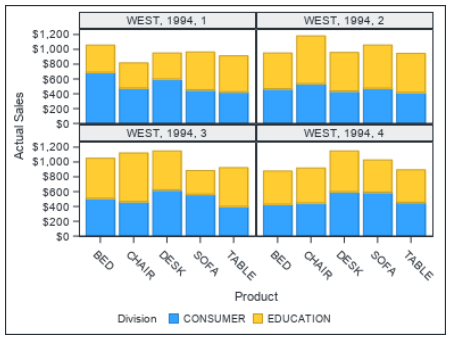
HEADERPACK
HEADERPACK=*integer-value-count*
specifies whether the header cells are consolidated into a comma-separated list in order to save space. When specified with no value, the header values are packed into one line of cell header text. If the length of the cell header text exceeds the available width, then the text is truncated. In that case, specify HEADERPACK=*integer-value-count* instead to split the text.

integer-value-count
specifies the number of headers to consolidate on a header line before splitting the text to the next line.

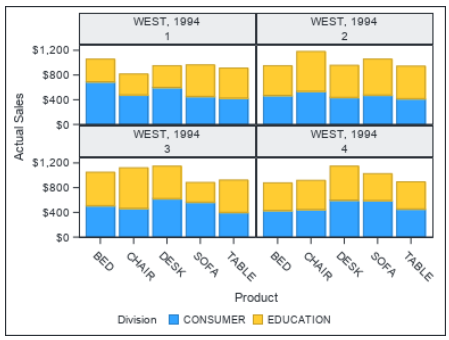
Default The number of panel-by variables

Range 1 to the number of panel-by variables

Note If the specified *integer-value-count* is out of range, the default value is used instead.




```
ods graphics on / reset width=4in;

proc sgpanel data=sashelp.prdsale;
panelby region year quarter / layout=panel
   novarname headerpack;
   vbar product / response=actual stat=mean
      group=division;
   where year eq 1994 and region eq "WEST";
run;
```




```
ods graphics on / reset width=4in;

proc sgpanel data=sashelp.prdsale;
panelby region year quarter / layout=panel
   novarname headerpack=2;
   vbar product / response=actual stat=mean
      group=division;
   where year eq 1994 and region eq "WEST";
run;
```

Default The value of each panel-by variable is placed on a separate line in the headers.

Requirement LAYOUT=PANEL (default) must also be in effect. Otherwise, this option is ignored.

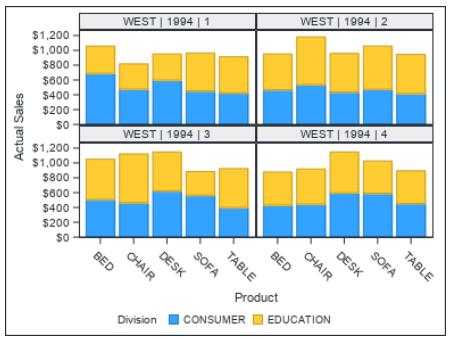
Tip Use the HEADERSEP= option to change the value separator.

HEADERSEP="*string*"

specifies one or more characters to place between each value in packed cell headers.

Default A comma followed by a space

Requirement HEADERPACK or HEADERPACK=*integer-value-count* must also be specified. Otherwise, this option is ignored.




```
ods graphics on / reset width=4in;

proc sgpanel data=sashelp.prdsale;
panelby region year quarter / layout=panel
   novarname headerpack headersep=" | ";
vbar product / response=actual stat=mean
   group=division;
where year eq 1994 and region eq "WEST";
run;
```

LAYOUT=LATTICE | PANEL | COLUMNLATTICE | ROWLATTICE
specifies the type of layout that is used for the panel. Select one of the following values:

LATTICE
when you specify two classification variables, arranges the cells so that the values of the first variable are columns and the values of the second variable are rows.

You can use LATTICE only when you specify exactly two classification variables.

PANEL
arranges the cells in rows and columns.

The headings for each cell are placed at the top of the cell.

COLUMNLATTICE
arranges the cells in a single row.

You can use the COLUMNLATTICE layout only with a single classification variable.

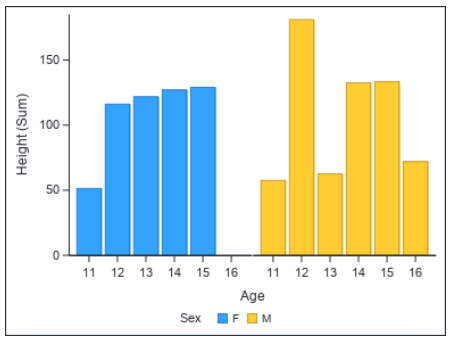
ROWLATTICE
arranges the cells in a single column.

You can use the ROWLATTICE layout only with a single classification variable.

Default PANEL

MISSING
processes missing values as a valid classification value and creates cells for it. By default, missing values are not processed as a classification value.

NOHEADER
hides the cell headings. This option is useful when a legend is displayed.




```
ods graphics on / reset=all;
ods graphics on / width=4in;

proc sgpanel data=sashelp.class;
  panelby sex / noheader noborder;
  vbar age / response=height group=sex;
run;
```

NOHEADERBORDER
removes the border from the cell headings.

Restriction NOHEADERBORDER does not remove the border around empty cells. If your panel has empty cells and you want to remove the cell borders, you can specify SKIPEMPTYCELLS with

NOHEADERBORDER to remove the empty cells and the cell borders.

NOVARNAME
removes the variable names from the cell headings of a panel layout, or from the row and column headings of a lattice layout. For example, to change a row heading from “Region=NorthEast” to “NorthEast,” specify the NOVARNAME option.

NOWALL
turns off the display of the graph wall. This option might be useful when your graph contains an annotation, and the wall color interferes with that annotation.

For most styles, the wall color is the same as the graph background, and it is impossible to see the difference. However, if this is not the case with the style that you use for a graph, then you might want to suppress the wall fill.

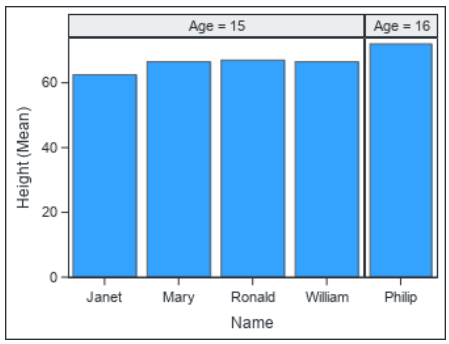
ONEPANEL
places the entire panel in a single output image. If you do not specify this option, then the panel is automatically split into multiple images as appropriate.

Interactions When you use ONEPANEL with the PANEL layout, only one of the ROWS= and COLUMNS= options can be used. If you specify both options, then the value for COLUMNS= is used.

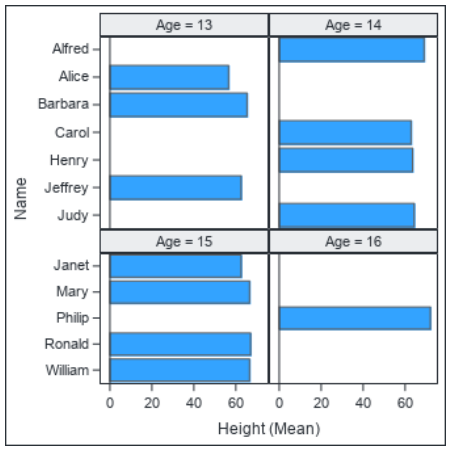
When you use ONEPANEL with the LATTICE layout, the ROWS= and COLUMNS= options have no effect.

Note This option is recommended only for panels with a small number of cells. If your panel is too large for the output image, then a blank image is created.

PROPORTIONAL
changes the size of the cells in proportion to the content of the discrete axis. The following examples show the use of PROPORTION on the X axis and on the Y axis. The first example adjusts the width of the two cells relative to the number of vertical bars along the X axis of each cell. The second example adjusts the height of the rows relative to the number of horizontal bars along the Y axis for each row.




```
ods graphics / width=3.5in;

proc sgpanel data=sashelp.class;
  where age > 14;
  panelby age / uniscale=row proportional;
  vbar name / response=height stat=mean;
run;
ods graphics / reset=width;
```




```
ods graphics / width=3.5in;

proc sgpanel data=sashelp.class;
  where age > 12;
  panelby age / uniscale=column proportional;
  hbar name / response=height stat=mean;
run;
ods graphics / reset=width;
```

Requirements The axis must be discrete.

You must also use the UNISCALE= option in the statement. Otherwise, the discrete axis cannot scale independently.

Interaction When PROPORTIONAL is used and the graph contains markers, the markers might be sized differently between panels depending on the relative sizes of the panels. To disable the scaling of the markers, you can turn off the SCALEMARKERS option in the ODS GRAPHICS statement, as shown here:

```
ODS GRAPHICS / SCALEMARKERS=OFF;
```

ROWHEADERPOS=LEFT | RIGHT | BOTH

specifies the location of the row headings in the panel. Specify one of the following values:

LEFT

places row headings at the left side of each row.

RIGHT

places row headings at the right side of each row.

BOTH

places row headings at the left side and the right side of each row.

Default RIGHT

Interaction This option has no effect if the panel uses the PANEL layout.

ROWS=*n*

specifies the number of rows in the panel. By default, the number of rows is determined automatically based on the number of classifier values and the layout type. The SGPANEL procedure automatically splits the panel into multiple panel images (pages) as needed when your panel contains a large number of cells.

If you specify ROWS= without specifying a value for COLUMNS= (by default, COLUMNS=1), then one of the following occurs:

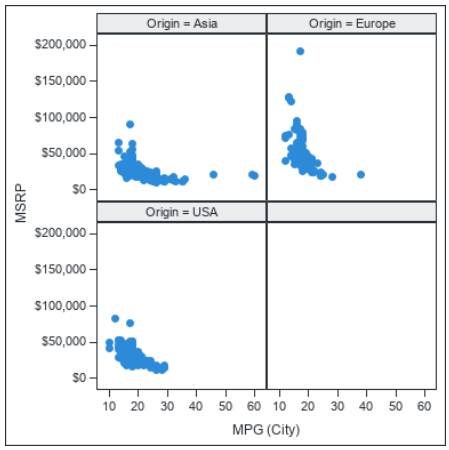
- The cells fill each row from left to right up to the requested number of rows. Each panel can have a maximum of three columns. For example, if you specify two rows, and the classifier has six values, the result is a three-column by two-row panel. If you specify two rows, and the classifier has eight values, the result is two panels. Each panel has two rows and two columns.
- If the number of classification variables is not divisible by the rows value, then the procedure simply creates multiple single-column panels. Each panel has the specified number of rows. The last panel is only partially populated.

However, if you also specify COLUMNS= with a value greater than one, then the procedure uses your specified values to determine the number of panels. Specifying both COLUMNS= and ROWS= enables you to define the precise layout, including how many cells are in a panel.

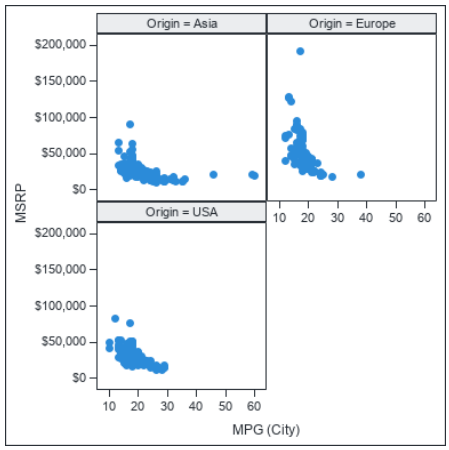
SKIPEMPTYCELLS

specifies whether the external axes skip the empty cells in a partially filled grid. If the number of classifier values does not match the number of cells in the rows and columns of the layout, then the grid is partially filled with data cells and padded with empty cells to complete the grid. Specifying the SKIPEMPTYCELLS option removes those empty cells.

In the following examples, the data cells are arranged in a two-column, two-row panel. The first graphic shows the panel padded with an empty cell. The second specifies SKIPEMPTYCELLS, and the panel contains no empty cells.




```
proc sgpanel data=sashelp.cars;
  panelby origin;
  scatter x=mpg_city y=msrp;
run;
```




```
proc sgpanel data=sashelp.cars;
panelby origin / skipemptycells;
scatter x=mpg_city y=msrp;
run;
```

The SKIPEMPTYCELLS option works whether the rows and columns are determined automatically or explicitly specified.

> **TIP** If your panel has empty cells and you want to remove the cell borders, you can specify SKIPEMPTYCELLS with NOHEADERBORDER to remove the empty cells and the cell borders.

SORT=*sort-option* | (*sort-option-1*...*sort-option-n*)
specifies the sort order of the panel cells.

Note: This option affects only the order of the cells in your panel. It does not affect the order of the cell contents.

If you have multiple classification variables, you can control the sort for each variable using (*sort-option-1*...*sort-option-n*).

sort-option can be one of the following values.

(*sort-option-1*...*sort-option-n*) can be one or more of the first five values listed. If you attempt to use any other values (such as DATA or any value after DATA), the sort request is ignored and a message is written to the log.

AUTO	sorts in ascending order for character data and numeric data. The sorts performed by PROC SORT and PROC SQL are also honored with this option. This option represents the default sorting behavior.
ASCENDING	sorts in ascending order using the unformatted values.
DESCENDING	sorts in descending order using the unformatted values.
ASCFORMAT	sorts in ascending order using the formatted values.
DESCFORMAT	sorts in descending order using the formatted values.
DATA	uses data order for the sort.

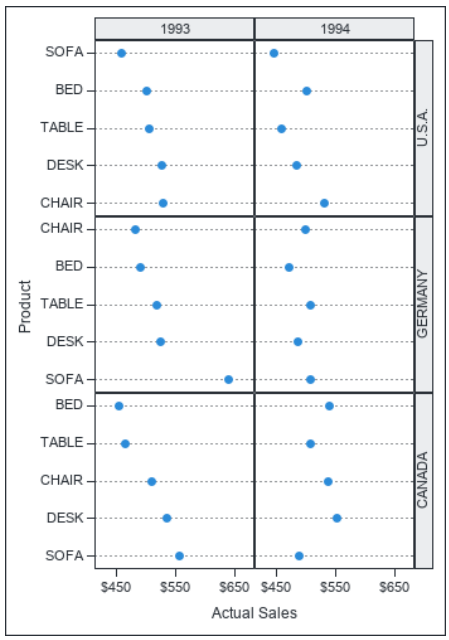
ASCMEAN	sorts by the ascending mean of the Y (or response) variable of the primary plot.
DESCMEAN	sorts by the descending mean of the Y (or response) variable of the primary plot.
ASCMEDIAN	sorts by the ascending median of the Y (or response) variable of the primary plot.
DESCMEDIAN	sorts by the descending median of the Y (or response) variable of the primary plot.
ASCFREQ	sorts by the ascending frequency of the class values.
DESCFREQ	sorts by the descending frequency of the class values.

The *sort-option* is applied to your panel variables as follows:

- When a single *sort-option* is specified, the sort is applied to all panel variables.
- When (*sort-option-1*...*sort-option-n*) values are specified, *sort-option-1* corresponds to the first PANELBY variable that is specified, *sort-option-2* corresponds to the second variable, and so on.

  If you do not specify enough *sort-option* values, the list is padded with AUTO. If you specify too many *sort-option* values, the rest of the list is ignored. If you want to skip the first variable but sort the second, use AUTO for the first variable.

The following example shows product sales by year in ascending (formatted) order and by country in descending order.




```
proc sgpanel data=sashelp.prdsale;
  panelby year country / layout=lattice
    onepanel sort=(ascformat descformat)
    uniscale=column novarname;
  dot product / response=actual stat=mean
    categoryorder=respasc;
run;
```

You cannot use a (*sort-option-1*...*sort-option-n*) list with a statistical sort, such as DESCMEAN. However, there is a way to obtain a statistical sort for multiple class variables:

1 Merge the statistic variable with the raw data by the class variables.

2 Use the SORT procedure to sort the data by the statistic variable.

3 In the SGPANEL procedure, specify SORT=DATA in the PANELBY statement.

Statistical sorts are most effective when you specify COLUMNS=1 or ROWS=1 in the PANELBY statement.

Default AUTO

Restriction DATA order is not supported with data that is stored in a table.

SPACING=*n*
specifies the number of pixels between the rows and columns in the panel.

Default 0

SPARSE
enables the SGPANEL procedure to create empty cells for crossings of the classification variables that are not present in the input data set. By default, empty cells are not created for the panel layout.

Interaction This option has no effect if you specify LAYOUT=LATTICE.

START=TOPLEFT | BOTTOMLEFT
specifies whether the first cell in the panel is placed at the upper left corner or the lower left corner. Specify one of the following values:

TOPLEFT
places the cell for the first data crossing in the upper left corner.

Cells are placed from left to right, starting in the top row. Each additional row is placed below the previous row.

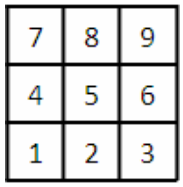
The following figure shows the placement of nine cells in a panel where START= TOPLEFT:

1	2	3
4	5	6
7	8	9

BOTTOMLEFT
places the cell for the first data crossing in the lower left corner.

Cells are placed from left to right, starting in the bottom row. Each additional row is placed above the previous row.

The following figure shows the placement of nine cells in a panel where START=BOTTOMLEFT:



Default TOPLEFT

UNISCALE=COLUMN | ROW | ALL
scales the shared axes in the panel to be identical. Specify one of the following values:

COLUMN
scales all of the column axes in the panel to have identical ranges.

The range for each row axis is scaled independently based on the data in each row.

ROW
scales all of the row axes in the panel to have identical ranges.

The range for each column axis is scaled independently based on the data in each column.

ALL
scales all of the column axes to be identical, and also scales all of the row axes to be identical.

Default ALL

Tip To arrange the cells in a single row or a single column, use the LAYOUT= option.

STYLEATTRS Statement

Specifies group attributes for a graph. The statement enables you to change group colors, markers, and so on, within the procedure, without having to change the ODS style template.

Requirement: The procedure must include at least one plot statement.

Interaction: The following options apply only when the GraphData1–GraphData*n* style elements are used: DATACOLORS=, DATACONTRASTCOLORS=, DATAFILLPATTERNS=, DATALINEPATTERNS=, and DATASYMBOLS=. For more information, see "When the Style Options Take Effect" on page 185.

Tip: When specifying data attributes such as colors, symbols, and line patterns for multiple plots, the attributes might not be assigned consistently in some cases. To ensure consistent attribute assignments, specify the option NOCYCLEATTRS on page 163 in the SGPANEL procedure statement.

Syntax

STYLEATTRS <*options*>;

Summary of Optional Arguments

Background and wall options

BACKCOLOR=*color*
specifies the background color of the graph area.

WALLCOLOR=*color*
specifies the color of the plot wall area.

Style options

DATACOLORS=(*color-list*)
specifies the fill colors for the graphics elements.

DATACONTRASTCOLORS=(*color-list*)
specifies the contrast colors for the graphics elements, such as lines and markers.

DATAFILLPATTERNS=(*fill-pattern-list*)
specifies the list of graph fill patterns.

DATALINEPATTERNS=(*line-pattern-list*)
specifies the list of line patterns for the graph data lines.

DATASYMBOLS=(*marker-symbol-list*)
specifies the list of marker symbols for the graph data.

Optional Arguments

BACKCOLOR=*color*
specifies the background color of the graph area. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The Color attribute of the GraphBackground style element

Examples backcolor=CXFF0000

backcolor=light_blue

DATACOLORS=(*color-list*)
specifies the fill colors for the graphics elements. The graphics elements can be in grouped plots or in overlaid multiple plots with the CYCLEATTRS feature in effect.

Provide a space-separated list of colors enclosed in parentheses. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The colors that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Interaction Where applicable, the COLOR= suboption of any plot option related to fill color overrides the DATACOLORS= option.

Note When this option is specified, the colors cycle through *color-list* rather than the colors that are defined in the GraphData1–GraphData*n* style elements. When the colors in *color-list* are exhausted, the colors repeat.

Example datacolors=(CXFF0000 green blue)

DATACONTRASTCOLORS=(*color-list*)
specifies the contrast colors for the graphics elements, such as lines and markers. The lines and markers can be in grouped plots or in overlaid multiple plots with the CYCLEATTRS feature in effect.

Provide a space-separated list of colors enclosed in parentheses. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The contrast colors that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Interaction Where applicable, the COLOR= suboption of any plot option related to a marker or line color overrides the DATACONTRASTCOLORS= option.

Note When this option is specified, the colors cycle through *color-list* rather than the contrast colors that are defined in the GraphData1–GraphData*n* style elements. When the colors in *color-list* are exhausted, the colors repeat.

Example datacontrastcolors=(orange cyan CXFF0000)

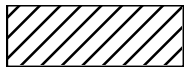
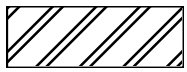
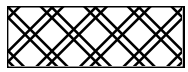
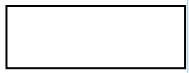
DATAFILLPATTERNS=(*fill-pattern-list*)
specifies the list of graph fill patterns.

fill-pattern-list
a space-separated list of fill patterns, enclosed in parentheses.

You can use a style attribute reference such as GraphData3:fillpattern or a two-character fill-pattern name such as L1, X5, and so on, to specify a fill pattern. The list can contain a mix of style attribute references and pattern names.

To specify a two-character fill-pattern name, specify a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5. The following table shows the patterns for each of the possible combinations.

Table 8.3 Fill Patterns

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

When this option is specified, the fill patterns rotate through this fill-pattern list rather than through the fill patterns that are defined in the GraphData1–GraphData*n* style elements. For information about the attribute rotation patterns, see "How the Attributes Are Cycled" on page 185..

Default The fill patterns that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Requirement The list of fill patterns must be enclosed in parentheses.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Example
```
datafillpatterns=(graphdata4:fillpattern L5 X2)
```

DATALINEPATTERNS=(*line-pattern-list*)
specifies the list of line patterns for the graph data lines. Provide a space-separated list of line patterns enclosed in parentheses. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default The line patterns that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Interaction Where applicable, the PATTERN= suboption of any plot option related to line attributes overrides the DATALINEPATTERNS= option.

Note When this option is specified, the line patterns cycle through *line-pattern-list* rather than the line patterns that are defined in the GraphData1–GraphData*n* style elements. When the patterns in *line-pattern-list* are exhausted, the patterns repeat.

Example `datalinepatterns=(dot solid longdash 26)`

DATASYMBOLS=(*marker-symbol-list*)
specifies the list of marker symbols for the graph data. Provide a space-separated list of symbols enclosed in parentheses. See marker symbols on page 1834.

Default The symbols that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

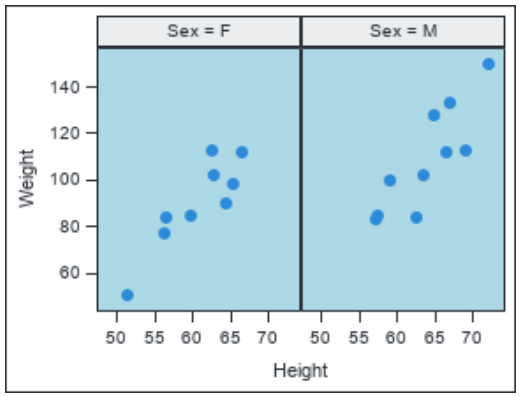
Interaction Where applicable, the SYMBOL= suboption of the MARKERATTRS= option overrides the DATASYMBOLS= option.

Note When this option is specified, the symbols cycle through *marker-symbol-list* rather than the symbols that are defined in the GraphData1–GraphData*n* style elements. When the symbols in *marker-symbol-list* are exhausted, the symbols repeat.

Example datasymbols=(circle square triangle star)

WALLCOLOR=*color*
specifies the color of the plot wall area. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

The following figure shows a light blue wall color.



Default The Color attribute of the GraphWalls style element

Examples wallcolor=CXFF0000

wallcolor=light_blue

Details

When the Style Options Take Effect

The following style options apply only when the GraphData1–GraphData*n* style elements are used: DATACOLORS=, DATACONTRASTCOLORS=, DATAFILLPATTERNS=, DATALINEPATTERNS=, and DATASYMBOLS=.

The corresponding attributes for the style options take effect when any one of the following is true:

- a GROUP= option is specified in at least one plot statement.
- the procedure contains at least two plot statements that cycle the respective attribute.
- a plot statement specifies a GraphData*n* style reference for the respective attribute (COLOR=). Example:

```
proc sgplot data=sashelp.class;
   styleattrs datacolors=(orange);
   vbar age / fillattrs=GraphData1;
run;
```

  In this case, DATACOLORS= overrides GraphData1:Color.

For more information, see the following blogs:

- Getting Started with SGPLOT: Style Attributes
- Attrs, attrs, everywhere: The interaction between ATTRPRIORITY, CYCLEATTRS, and STYLEATTRS in ODS Graphics

How the Attributes Are Cycled

The STYLEATTRS feature cycles the attributes in the order specified in your attribute list until the group values are exhausted. If the specified attribute list is exhausted first, then the list is repeated. If you specify multiple options, the STYLEATTRS feature combines the options as it cycles through.

In the following example, the DATACONTRASTCOLORS= option specifies three contrast colors used for lines. The DATALINEPATTERNS= option specifies two line patterns, and the DATASYMBOLS= option specifies two marker symbols.

```
proc sort data=sashelp.class out=class;
   by age;
run;
proc sgplot data=class;
   styleattrs
      datacontrastcolors=(red green blue)
      datalinepatterns=(dot solid)
      datasymbols=(circlefilled squarefilled);
   needle x=height y=weight / group=age markers;
   keylegend / across=3;
run;
```

In this example, the Illuminate style is used, so the procedure uses a color-priority rotation pattern.

With the color-priority rotation pattern, marker symbols and line patterns are held constant while each color in the list is applied to the marker symbol and line.

In the example, the dotted line pattern is held constant while the procedure applies red, green, and blue colors to the dotted lines and circlefilled marker symbols for the consecutive group values. If there are more group values, the options apply the red, green, and blue colors to solid lines and squarefilled markers.

The following figure shows an example rotation for an age grouping.

You can change this rotation pattern by using the ATTRPRIORITY= option in the ODS GRAPHICS statement. The ATTRPRIORITY= option controls the rotation pattern for the attributes derived from the GraphData1–GraphData*n* style elements for the style that is in effect.

Here is the general syntax for the option.

ATTRPRIORITY=COLOR | NONE

The default behavior for this option depends on the ODS style that is in use. For styles such as Illuminate and HTMLBlue, ATTRPRIORITY in the style is set to COLOR. For some of the other styles, ATTRPRIORITY in the style is set to NONE.

You can use the ATTRPRIORITY= option to control the rotation pattern for the attributes that you specify with the STYLEATTRS feature.

To change the rotation pattern in the previous example, specify the following before you invoke the procedure:

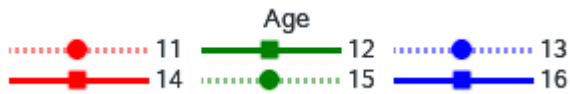
```
ods graphics / attrpriority=none;
```

Now, the following options in the STYLEATTRS statement are applied in alternating order:

```
proc sgplot data=class;
   styleattrs
      datacontrastcolors=(red green blue)
      datalinepatterns=(dot solid)
      datasymbols=(circlefilled squarefilled);
   needle x=height y=weight / group=age markers;
   keylegend / across=3;
run;
```

In the output, a red dotted line pattern and a red circlefilled marker are applied for the first group crossing, a solid green line pattern and a green squarefilled marker are applied for the second, a blue dotted line pattern and a blue circlefilled marker are applied for the third, and so on.

The following figure shows the rotation for the age grouping with no priority rotation.



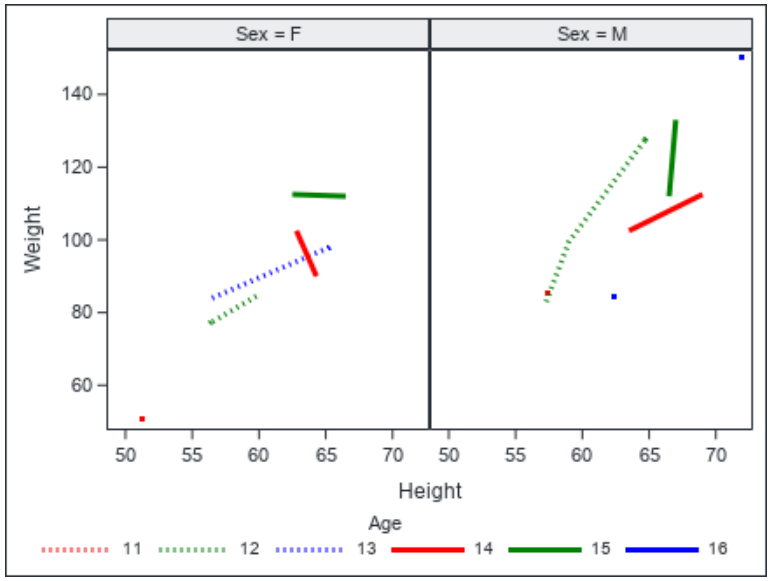
See Also

"ODS GRAPHICS Statement" on page 102

Example: Sorted Data with Contrast Colors and Line Patterns Specified

This example has the following features:

- The input data is sorted by the group variable.
- The ODS style is Illuminate. For the Illuminate style, attribute priority is color by default. Therefore, the procedure uses a color-priority rotation pattern to determine the output colors and line patterns.




```
/* Sort the data set by the group variable */
proc sort data=sashelp.class out=class;
  by age;
run;

/* Generate the graph using the sorted data */
proc sgpanel data=class;
  panelby sex;
  styleattrs
     datacontrastcolors=(red green blue)
     datalinepatterns=(dot solid);
  series x=height y=weight / group=age;
  keylegend;
run;
```

BAND Statement

Creates a band that highlights part of a plot.

Restrictions: The axis that the UPPER and LOWER values are placed on cannot be a discrete axis. For example, if you specify a variable for Y, the plot cannot use a discrete horizontal axis.

The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

Note: The input data should be sorted by the X or Y variable. If the data is not sorted, the graph might produce unpredictable results.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Band Plots" on page 17

Syntax

BAND X=*variable* | Y=*variable*
UPPER=*numeric-value* | *numeric-variable*
LOWER=*numeric-value* | *numeric-variable*
</option(s)>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

FILL

NOFILL
: specifies whether the area fill is visible.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

LINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the lines in the plot.

NOEXTEND
when you specify numeric values for UPPER= and LOWER=, specifies that the band does not extend beyond the first and last data points in the plot.

OUTLINE

NOOUTLINE
specifies whether the outlines of the band are visible.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

TYPE=SERIES | STEP
specifies how the data points for the lower and upper band boundaries are connected.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you specify a curve label.

CURVELABELLOWER="*text-string*"
adds a label to the lower edge of the band.

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label.

CURVELABELUPPER="*text-string*"
adds a label to the upper edge of the band.

LEGENDLABEL="*text-string*"
specifies a label that identifies the elements from the band plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text used for curve labels is always split.

SPLITCHARNODROP

specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot reference options

MODELNAME="*plot-name*"
specifies the name of a plot from which to derive the interpolation for the band.

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*variable*
specifies a variable that is used to plot the band along the X axis.

Y=*variable*
specifies a variable that is used to plot the band along the Y axis.

LOWER=*numeric-value* | *numeric-variable*
specifies the lower value for the band. You can specify either a constant numeric value or a numeric variable.

UPPER=*numeric-value* | *numeric-variable*
specifies the upper value for the band. You can specify either a constant numeric value or a numeric variable.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you specify a curve label. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Interaction This option has no effect unless CURVELABELLOWER or CURVELABELUPPER is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family=Arial Size=8
     Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOWER="*text-string*"
adds a label to the lower edge of the band. Specify the label text.

Restriction This option cannot be used with the GROUP= option. The label specified is not drawn when the GROUP= option is in effect.

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label. Specify one of the following values:

MIN
places the label at the part of the curve closest to the minimum X-axis value.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

START
places the curve label at the first point on the curve.

END
places the curve label at the last point on the curve.

Default END

Interaction This option has no effect unless the CURVELABELLOWER= or CURVELABELUPPER= option is also specified.

CURVELABELUPPER="*text-string*"
adds a label to the upper edge of the band. Specify the label text.

Restriction This option cannot be used with the GROUP= option. The label specified is not drawn when the GROUP= option is in effect.

FILL | NOFILL
specifies whether the area fill is visible. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default The default status of the area fill is specified by the DisplayOpts style attribute of the GraphBand style element in the current style.

Interactions Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see “Fill Attributes” on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

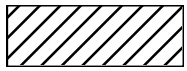
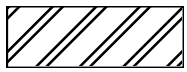
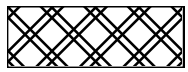
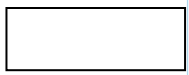
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.4 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions

For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note

This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

GROUP=*variable*

specifies a variable that is used to group the data. A separate band is created for each unique value of the grouping variable.

LEGENDLABEL="*text-string*"
specifies a label that identifies the elements from the band plot in the legend. By default, the label "Band" is used for ungrouped data, and the group values are used for grouped data.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor and LineStyle, and LineThickness attributes of the GraphConfidence style element in the current style.

For grouped data, the lineThickness attribute of the GraphConfidence style element, and the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless you also specify the OUTLINES option.

MODELNAME="*plot-name*"
specifies the name of a plot from which to derive the interpolation for the band.

Default If you do not specify this option, then the band is interpolated in the same way as a series plot.

Requirement *plot-name* must be the name that has been assigned with the associated plot's NAME= option.

See "Details" on page 198

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOEXTEND
when you specify numeric values for UPPER= and LOWER=, specifies that the band does not extend beyond the first and last data points in the plot. By default, the band extends to the edges of the plot area.

Interaction This option has no effect if you do not specify numeric values for the UPPER= and LOWER= options.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

OUTLINE | NOOUTLINE
specifies whether the outlines of the band are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default The default status of the band outlines is specified by the DisplayOpts attribute of the GraphBand style element in the current style.

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text used for curve labels is always split. The text is split at every occurrence of the specified split character or characters. This option affects both the upper and lower curve labels if they are displayed.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Interactions This option has no effect unless either CURVELABELLOWER or CURVELABELUPPER is also specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

TYPE=SERIES | STEP
specifies how the data points for the lower and upper band boundaries are connected. You can specify one of the following:

SERIES
the data points are connected directly using line segments, as in a series plot.

STEP
the data points are connected using a step function, as in a step plot.

Default SERIES

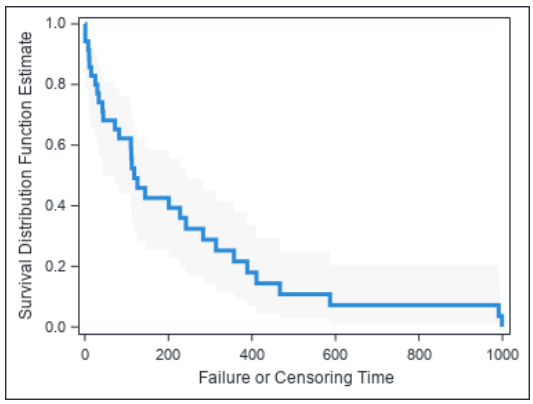
Details

The MODELNAME= option fits a band to another plot. This is particularly useful for plots that use a special interpolation such as step plots.

The following code fragment fits a band to a step plot:

```
band x=SurvTime upper=sdf_ucl lower=sdf_lcl / modelname="myname"
transparency=.5;
step x=SurvTime y=survival / name="myname";
```

Figure 8.4 *Fitted Band Plot Example*



BLOCK Statement

Creates one or more rectangular blocks containing text values. The block widths are determined by a specified block variable.

Notes:

Each block represents a block variable value and spans the data that has the same block variable value.

When data labels are used, the axis offsets are extended to accommodate the length of the longest data label. In that case, the blocks at each end of the block plot are extended to include the additional axis offset. This behavior is expected.

The BLOCK statement is typically used with another plot statement, which establishes the axis type for the Y axis. However, the BLOCK statement can be

used by itself. In that case, the block plot has no Y axis. See "Plot Compatibility" on page 1780.

Example: "About Block Plots" on page 18

Syntax

BLOCK X=*category-variable* BLOCK=*block-variable* </option(s)>;

Summary of Optional Arguments

Appearance options

ALTFILLATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of alternate fills for the blocks.

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

FILL

NOFILL
: specifies whether the blocks are filled.

FILLATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the fill for the blocks.

FILLTYPE=MULTICOLOR | ALTERNATE
: specifies how the blocks are filled.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the block outlines.

OUTLINE

NOOUTLINE
: specifies whether the blocks have outlines.

TRANSPARENCY=*numeric-value*
: specifies the degree of transparency for the blocks.

Block text options

BLOCKLABEL=*variable*
: specifies a column to use for alternative text in the blocks.

NOVALUES

VALUES
: specifies whether the block values are displayed or hidden.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the block text values are split according to the fit policy that is in effect.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the block text values.

VALUEFITPOLICY=NONE | SHRINK | SPLIT | SPLITALWAYS | TRUNCATE

specifies how text values are adjusted to fit within the containing block.

VALUEHALIGN=LEFT | CENTER | RIGHT | START
specifies the horizontal alignment of the value text within the blocks.

VALUEVALIGN=TOP | CENTER | BOTTOM
specifies the vertical alignment of the value text within the blocks.

Label options

LABEL

LABEL="*text-string*"

NOLABEL
specifies an external label for a single block plot.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the color and font attributes of the external block labels.

LABELPOS=BOTTOM | LEFT | RIGHT | TOP
specifies the position for the block label for a single block plot.

Plot options

CLASS=*variable*
creates a stack of block plots, with one block plot for each unique value of the specified variable.

EXTENDMISSING
extends the previous block value if the current value is missing.

NOMISSINGCLASS
suppresses blocks that correspond to missing values of the CLASS= value.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*category-variable*
specifies X-axis positions. When the X axis is numeric and the specified variable is numeric, values are expected to be in sorted, ascending order. If the X axis is discrete and the specified column is numeric, values are treated as numeric-discrete.

BLOCK=*block-variable*
specifies the variable that classifies the observations into distinct subsets.

Optional Arguments

ALTFILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of alternate fills for the blocks. This option in conjunction with the FILLATTRS= option controls fill appearance when FILLTYPE=ALTERNATE. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Requirement FILLTYPE=ALTERNATE must be specified for this option to have any effect.

Interaction This option has no effect if you specify the NOFILL option.

Note The TRANSPARENCY= attribute affects only the alternate fill blocks. To set the same transparency for both the fill blocks and the alternate fill blocks, use the TRANSPARENCY= option in the BLOCK statement.

Tips The FILLATTRS= option controls the fill color of non-alternate blocks.

To make all block fill areas the same color, set the FILLATTRS= and ALTFILLATTRS= options to the same value.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

BLOCKLABEL=*variable*
specifies a column to use for alternative text in the blocks.

Interaction This option overrides the default text for the blocks.

Tip The font and color attributes for the alternative text are specified by the VALUEATTRS= option.

CLASS=*variable*
creates a stack of block plots, with one block plot for each unique value of the specified variable.

Interaction To label the blocks by the class values, specify the BLOCKLABEL= option using the same class variable.

Note For local data, the visual attributes for each class value are assigned in data order. However, if you are using data that is stored in a table, the visual attributes for each class value are assigned in ascending order of the class variable character values or of unformatted numeric values.

EXTENDMISSING
extends the previous block value if the current value is missing.

FILL | NOFILL
specifies whether the blocks are filled.

Default FILL

Interaction The NOFILL option can be used with the NOOUTLINE option to hide both the fill and the outline.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fill for the blocks. This option in conjunction with the ALTFILLATTRS= option controls fill appearance when FILLTYPE=ALTERNATE. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Requirement FILLTYPE=ALTERNATE must be specified for this option to have any effect.

Interaction This option has no effect if you specify the NOFILL option.

Note The TRANSPARENCY= attribute affects only the fill blocks. To set the same transparency for both the fill blocks and the alternate fill blocks, use the TRANSPARENCY= option in the BLOCK statement.

Tips The ALTFILLATTRS= option controls the alternate fill color.

To make all block fill areas the same color, set the FILLATTRS= and ALTFILLATTRS= options to the same value.

FILLTYPE=MULTICOLOR | ALTERNATE
specifies how the blocks are filled.

MULTICOLOR
Blocks are filled with the COLOR attribute of the GraphData1– GraphData*n* style elements.

ALTERNATE
Blocks are filled alternating between the colors specified by the FILLATTRS= and ALTFILLATTRS= options.

Default MULTICOLOR

Interaction This option has no effect if NOFILL is also specified.

LABEL
LABEL="*text-string*"
NOLABEL
specifies an external label for a single block plot. If you specify the LABEL option, the procedure displays the name of the block variable. You can override that label by providing your own text.

Default The label text is the variable label of the BLOCK= variable. If there is no variable label, the variable name is used.

Interactions By default, the label appears to the left of the plot. You can specify a different position for the label using the LABELPOS= option.

You can specify the font and color attributes for the label using the LABELATTRS= option.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the color and font attributes of the external block labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
     Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LABELPOS=BOTTOM | LEFT | RIGHT | TOP
specifies the position for the block label for a single block plot.

Default LEFT

Interaction This option has no effect if NOLABEL is also specified.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the block outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interactions This option has no effect if the NOOUTLINE option is also specified.

The external labels are also outlined when LABEL= is TOP or BOTTOM. In that case, the LINEATTRS= option also controls the appearance of the external label outlines.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGCLASS
suppresses blocks that correspond to missing values of the CLASS= value.

OUTLINE | NOOUTLINE
specifies whether the blocks have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interaction The NOOUTLINE option can be used with the NOFILL option to hide both the outline and the fill.

NOVALUES | VALUES
specifies whether the block values are displayed or hidden.

Default VALUES

Tip The font and color attributes for the text values are specified by the VALUEATTRS= option.

SPLITCHAR="*character-list*"
specifies one or more characters on which the block text values are split according to the fit policy that is in effect. When VALUEFITPOLICY=SPLIT is in effect, the block text values are split on the specified character or characters where required to fit the block text within the containing block. When VALUEFITPOLICY=SPLITALWAYS is in effect, the block text values are split on every occurrence of the specified character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur. In that case, if the value does not fit the available space, it might collide with the values in the adjacent blocks.

Default The default split character is a space.

Interactions This option has no effect unless VALUEFITPOLICY= specifies SPLIT or SPLITALWAYS. The default value for VALUEFITPOLICY= is SPLIT.

When the text value is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the blocks. Transparency affects both the fill and the alternate fills, if enabled. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default 0.0

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the block text values. This option affects the default values that are associated with the BLOCK= argument. If BLOCKLABEL= is specified, then the option affects those values instead.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

VALUEFITPOLICY=NONE | SHRINK | SPLIT | SPLITALWAYS | TRUNCATE
specifies how text values are adjusted to fit within the containing block. Select one of the following values:

NONE
No attempt is made to fit values that collide with the text values in adjacent blocks.

SHRINK
All values are reduced in font size until they all fit.

SPLIT
If a value does not fit within the containing block, it is split at a split character.

No split occurs at split characters where a split is not needed. In that case, the split character is displayed with the text value.

If the value does not contain any of the specified split characters, a split does not occur. In that case, if the value does not fit the available space, it might collide with the adjoining values.

Default The default split character is a space.

Tip Use the SPLITCHAR= option to specify a split character.

SPLITALWAYS
Text values are split at a split character in all blocks.

Default The default split character is a space.

Tip Use the SPLITCHAR= option to specify a split character.

TRUNCATE
Any value that does not fit is truncated.

For a numeric column, an asterisk (*) is substituted for the entire value whenever truncation occurs. For a character column, the truncated portion of the text is replaced by an ellipsis (...).

Default SPLIT

VALUEHALIGN=LEFT | CENTER | RIGHT | START
specifies the horizontal alignment of the value text within the blocks. This option changes the text alignment regardless of whether you split values or specify the VALUEFITPOLICY= option.

Select one of the following values:

LEFT
left-aligned within the block

CENTER
center-aligned within the block

RIGHT
right-aligned within the block

START
center-aligned at the starting value of the block

Default CENTER

Interaction This option has no effect if you also specify the NOVALUES option.

VALUEVALIGN=TOP | CENTER | BOTTOM
specifies the vertical alignment of the value text within the blocks.

Default TOP

Interaction This option has no effect if you also specify the NOVALUES option.

Details

The BLOCK statement provides several options for changing the appearance of the blocks and the text values inside the blocks.

The syntax for these display attributes is almost identical between the SGPANEL and SGPLOT procedures, and is described in the following topics in the SGPLOT chapter:

- "Changing the Appearance of Block Text Values" on page 928
- "Changing the Appearance of Block Fills" on page 929

BUBBLE Statement

Creates a bubble plot in which two variables determine the location of the bubble centers and a third variable controls the size of the bubble.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Bubble Plots" on page 19

Syntax

BUBBLE X=*variable* Y=*variable* SIZE=*numeric-variable* </ option(s)>;

Summary of Optional Arguments

Appearance options

ABSSCALE
: specifies that the SIZE= column values are interpreted in the same units as the axes rather than as relative values.

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BRADIUSMAX=*numeric-value<units>*
: specifies the size of the radius of the largest bubble.

BRADIUSMIN=*numeric-value<units>*
: specifies the size of the radius of the smallest bubble.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DRAWORDER=SIZE | DATA
: specifies whether the bubbles are drawn according to bubble size or according to data order.

FILL

NOFILL
: specifies whether the bubbles are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the outlines for the bubbles.

OUTLINE

NOOUTLINE
: specifies whether the outlines of the bubbles are visible.

PROPORTIONAL
: specifies that proportional scaling be applied to the SIZE= column values.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
: specifies a variable that is used to group the data.

NOMISSINGGROUP
: specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL

DATALABEL=*variable*
: displays a label for each data point.

DATALABELATTRS=*style-element*<(**options**)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
: specifies the location of the data label with respect to the plot.

LEGENDLABEL="*text-string*"
: specifies the label that identifies the bubble plot in the legend.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

Plot options

URL=*character-variable*
: specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
: specifies a name for the plot.

Required Arguments

X=*variable*
: specifies the variable for the X axis.

Y=*variable*
: specifies the variable for the Y axis.

SIZE=*numeric-variable*
: specifies the variable that controls the size of the bubbles. The minimum and maximum values automatically provide the range that is used to determine bubble sizes. You can control this range manually by using the BRADIUSMAX and BRADIUSMIN options.

  Tip You can use the ABSSCALE option to interpret the bubble size in the same units as the axes rather than as relative values.

Optional Arguments

ABSSCALE
: specifies that the SIZE= column values are interpreted in the same units as the axes rather than as relative values. By default, the bubble sizes are scaled to represent the value range of the SIZE= column.

For example, suppose a graph contains only two bubbles, and their sizes are 2 and 4. These two bubbles appear the same as they would if their sizes were 4000 and 8000, respectively. By contrast, when ABSSCALE is specified, the size values are interpreted in the same units as the axes.

Restriction This option is ignored if the X or Y axis is discrete.

Interaction When this option is used, the BRADIUSMAX= and BRADIUSMIN= options are ignored.

Note The bubbles might be drawn as ellipses if the X and Y axes are scaled differently.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BRADIUSMAX=*numeric-value<units>*
specifies the size of the radius of the largest bubble. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Restriction The BRADIUSMAX= value must be greater than the BRADIUSMIN= value if one is specified. If BRADIUSMAX is not greater, an error is generated and a message is written to the SAS log.

Note If you specify the maximum size as a percentage, this is interpreted as a percentage of the graph's height.

BRADIUSMIN=*numeric-value<units>*
specifies the size of the radius of the smallest bubble. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Restriction The BRADIUSMIN= value must be smaller than the BRADIUSMAX= value if one is specified. If BRADIUSMIN is not smaller, an error is generated and a message is written to the SAS log.

Note If you specify the minimum size as a percentage, this is interpreted as a percentage of the graph's height.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element for filled bubbles. The ThreeColorAltRamp style element for unfilled bubbles.

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all bubbles. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 8.5 *DATASKIN Options for Bubbles*

NONE	CRISP	GLOSS

MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DRAWORDER=SIZE | DATA

specifies whether the bubbles are drawn according to bubble size or according to data order.

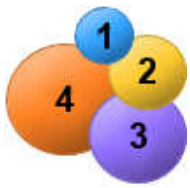
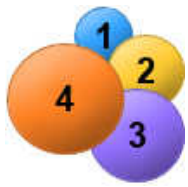
SIZE

draws the bubbles according to bubble size, from the largest to the smallest.

DATA

draws the bubbles according to data order.

The following figure shows the effect of SIZE and DATA on four bubbles. The bubble labels indicate the data order, and the bubble sizes increase linearly starting with 1.

DRAWORDER=SIZE	DRAWORDER=DATA
	

Default SIZE

FILL | NOFILL

specifies whether the bubbles are filled. The FILL option shows the fill color. The NOFILL option hides the fill color.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults: For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions: This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction: If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips: Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

COLOR=*color*
specifies a color to use for the fill lines.

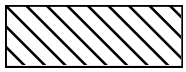
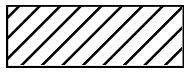
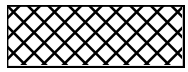
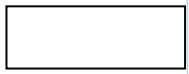
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*

specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.6 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions

For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

GROUP=*variable*
specifies a variable that is used to group the data. The bubbles for each group value are automatically distinguished by different colors.

When this option is used, the bubble colors are taken from the GraphData1–GraphData*n* style elements. If the bubbles are filled, then the COLOR attribute is used for bubble fill and CONTRASTCOLOR is used for the bubble outline. If the bubbles are not filled, then the CONTRASTCOLOR and PATTERN are used for the bubble outlines.

When this option is used and the value is a variable associated with an ATTRID= option, the attribute mapping defined by the associated attribute map is used.

Interaction: This option is ignored if the COLORRESPONSE= option is also used.

Tip: When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bubble plot in the legend.

Default: By default, the label for the Y variable is used for ungrouped data, and the group values are used for grouped data.

Interaction: The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the outlines for the bubbles. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults: For ungrouped data, the GraphOutlines style element in the current style.

For grouped data:

- if only outlines are displayed, LineThickness from GraphOutlines, and ContrastColor and LineStyle from a GraphData1–GraphData*n* style element in the current style
- if outlines and fill are displayed, LineThickness and LineStyle from GraphOutlines, and ContrastColor from a GraphData1–GraphData*n* style element in the current style

Interactions This option has no effect if you also specify the NOOUTLINE option.

This option is ignored if the DATASKIN= option is also specified.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OUTLINE | NOOUTLINE
specifies whether the outlines of the bubbles are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

PROPORTIONAL
specifies that proportional scaling be applied to the SIZE= column values. By default, scaling is linear, which increases the size of the bubbles in linear proportion to the range of the SIZE= column values. For example, if only two bubbles are drawn with sizes of 2 and 4, they appear the same as only two bubbles with sizes of 4000 and 8000. When PROPORTIONAL is specified, each bubble is sized in direct proportion to its corresponding SIZE= column value. For example, if only two bubbles are drawn with sizes of 50 and 100, then the SIZE=50 bubble is drawn to half the size of the SIZE=100 bubble.

Default linear

Interactions If all the values for the SIZE= column are negative, then PROPORTIONAL is ignored and the scaling is linear.

When PROPORTIONAL and BRADIUSMIN= are specified, the BRADIUSMIN= option specifies the minimum bubble size. In that case, when a SIZE= column value results in a bubble of a size that is less than the BRADIUSMIN= value, the bubble size for that value is changed to the BRADIUSMIN= value.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, “Using Range Attribute Maps,” on page 1921

“Overview of Attribute Maps” on page 1897

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See “Fit Policies for Axis Tick Values, Curve Labels, and Data Labels” on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See “Fit Policies for Axis Tick Values, Curve Labels, and Data Labels” on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted "text strings" enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

DENSITY Statement

Creates a density curve that shows the distribution of values in your data.

Restriction: The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

Interaction: The DENSITY statement can be combined only with the DENSITY and HISTOGRAM statements in the SGPANEL procedure. See "Plot Compatibility" on page 1780.

Examples: "About Density Plots" on page 49

"Example 1: Creating a Panel of Graph Cells with Histograms and Density Plots" on page 832

Syntax

DENSITY *response-variable* </option(s)>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

LINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the density line.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

SCALE=COUNT | DENSITY | PERCENT | PROPORTION
specifies the scaling that is used for the response axis.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

Label options

CURVELABEL="*text-string*"
specifies a label for the density curve.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS=END | MAX | MIN | START
specifies the location of the curve label.

LEGENDLABEL="*text-string*"
specifies a label that identifies the density plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

TYPE=NORMAL <(*normal-opts*)> | KERNEL <(*kernel-opts*)>
specifies the type of distribution curve that is used for the density plot.

WEIGHT=*numeric-variable*
specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Argument

response-variable
specifies the variable for the X axis. The variable must be numeric.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CURVELABEL="*text-string*"
specifies a label for the density curve.

Restriction This option is ignored when the GROUP= option is specified.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELPOS=END | MAX | MIN | START
specifies the location of the curve label. Specify one of the following values:

END
places the curve label at the last point on the curve.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

MIN
places the label at the part of the curve closest to the minimum X-axis value.

START
places the curve label at the first point on the curve.

Default END

Interactions This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

LEGENDLABEL="*text-string*"
specifies a label that identifies the density plot in the legend. By default, the label identifies the type of density curve. If you specify TYPE=NORMAL, then the default label is “Normal.” If you specify TYPE=KERNEL, then the default label is “Kernel.”

Note User-specified parameters from the TYPE= option are included in the label by default.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the density line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see “Line Attributes and Patterns” on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphFit style element in the current style.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

SCALE=COUNT | DENSITY | PERCENT | PROPORTION
specifies the scaling that is used for the response axis. Specify one of the following values:

COUNT
the axis displays the frequency count.

DENSITY
the axis displays the density estimate values.

PERCENT
the axis displays values as a percentage of the total.

PROPORTION
the axis displays values in proportion to the total.

Defaults When the DENSITY statement follows a HISTOGRAM statement, the density plot uses the histogram's response-axis scaling.

When the DENSITY statement does not follow a HISTOGRAM statement, the density plot uses DENSITY response-axis scaling.

Interaction This option is ignored when the DENSITY statement follows a HISTOGRAM statement. In that case, the density plot uses the histogram's response-axis scaling instead of the scaling specified in this option. See SCALE= on page 406 for the HISTOGRAM statement.

Note The PROPORTION scale can be used only when you combine a density plot and a histogram.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless CURVELABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

TYPE=NORMAL <(*normal-opts*)> | KERNEL <(*kernel-opts*)>
specifies the type of distribution curve that is used for the density plot. Specify one of the following keywords:

NORMAL <(*normal-opts*)>
specifies a normal density estimate, with a mean and a standard deviation.

normal-opts can be one or more of the following values:

MU=*numeric-value*
specifies the mean value that is used in the density function equation.

By default, the mean value is calculated from the data.

SIGMA=*numeric-value*
specifies the standard deviation value that is used in the density function equation.

The value that you specify for the SIGMA= suboption must be a positive number. By default, the standard deviation value is calculated from the data.

KERNEL <(*kernel-opts*)>
specifies a nonparametric kernel density estimate.

kernel-opts can be:

C=*numeric-value*
specifies the standardized bandwidth for a number that is greater than 0 and less than or equal to 100.

The value that you specify for the C= suboption affects the value of λ as shown in the following equation:

$$\lambda = cQn^{-\frac{1}{5}}$$

In this equation *c* is the standardized bandwidth, *Q* is the interquartile range, and *n* is the sample size.

WEIGHT=NORMAL | QUADRATIC | TRIANGULAR
specifies the weight function.

You can specify either normal, quadratic, or triangular weight function.

Default NORMAL

Note The kernel density function uses the same estimation that is used by the UNIVARIATE procedure. This estimation minimizes the Asymptotic Mean Integrated Square Error (AMISE) for the kernel density function.

Default NORMAL

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Details

Normal Density Function

When the type of the density curve is NORMAL, the fitted density function equation is as follows.

$$p(x) = \frac{100h\,\%}{\sigma\sqrt{2\,\pi}}\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \text{ for } -\infty < x < \infty$$

In the equation, μ is the mean, and σ is the standard deviation. You can specify μ by using the MU= suboption and σ by using the SIGMA= suboption.

Kernel Density Function

When the TYPE of the density curve is KERNEL, the general form of the kernel density estimator is as follows.

$$\widehat{f_\lambda}(x) = \frac{100h\,\%}{n\lambda} \sum_{i=1}^{n} K_0\left(\frac{x - x_i}{\lambda}\right)$$

In the equation, $K_0(\,\cdot\,)$ is the weight function, λ is the bandwidth, *n* is the sample size, and x_i is the *i*th observation. You can use the C= suboption to specify the bandwidth and the WEIGHT= suboption to specify the weight function $K_0(\,\cdot\,)$.

Kernel Density Weight Functions

The formulas for the weight functions are as follows.

NORMAL

$$K_0(t) = \frac{1}{\sqrt{2\,\pi}} \exp\left(-\frac{1}{2}t^2\right) \text{ for } -\infty < t < \infty$$

QUADRATIC

$$K_0(t) = \frac{3}{4}\left(1 - t^2\right) \text{ for } |t| \le 1$$

TRIANGULAR

$$K_0(t) = 1 - |t| \text{ for } -|t| \le 1$$

DOT Statement

Creates a dot plot that summarizes the values of a category variable.

Restrictions: The DOT statement can be combined only with other horizontal categorization plot statements. See "Plot Compatibility" on page 1780.

When category plots are overlaid, all of the plots must use the same category variable.

See: "Marker Fills and Outlines" on page 1805

Example: "About Dot Plots" on page 60

Syntax

DOT *category-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all dots from discrete category values.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
: specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
: specifies how to display grouped dots.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
: displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
: specifies the location of the data label with respect to the plot.

LEGENDLABEL="*text-string*"

specifies the label that identifies the dot plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

STATLABEL

NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels.

Limit options

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITS=UPPER | LOWER | BOTH
specifies which limit lines to display.

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines.

NOLIMITCAPS
suppresses the serif cap on limit lines, if limit lines are displayed.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT=STDERR.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

Plot options

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

MISSING

accepts a missing value as a valid category value.

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

WEIGHT=*numeric-variable*
specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Statistics options

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors when the COLORRESPONSE= option is specified.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the horizontal axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the dots.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default .05

Interactions This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged. By default, the plot is sorted in ascending order based on the category values. You can change the order so that it is based on the response values.

Specify one of the following values:

RESPASC
sorts by the response values in ascending order.

RESPDESC
sorts by the response values in descending order.

Default The plot is sorted in ascending order based on the category values.

Restriction This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Requirement This option requires that you configure the panel to use either one column or one row, depending on the orientation of your charts. Use the ROWS= option or the COLUMNS= option in the PANELBY statement. If you do not use this option and your graph contains multiple cells, the specified sort order is not correctly applied to all cells.

Interactions When a group variable is used with the CATEGORYORDER= option, the category order is not affected by the value of the groups. The categories are always sorted by the response statistic at a category level.

When this option and the GROUPORDER= option are both specified, the GROUPORDER= option is ignored.

Notes If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

When two or more observations have the same response values, there is no guarantee that the tied values will be sub-sorted.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interaction This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR specifies the color for the smallest data value of the COLORRESPONSE= column.

NEUTRALCOLOR specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.

ENDCOLOR specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorAltRamp style element

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified or COLORSTAT= must be FREQ or PCT.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors when the COLORRESPONSE= option is specified. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

SUM
sum values for the color response

MEAN
mean values for the color response

MEDIAN
median values for the color response

Default SUM when option COLORRESPONSE= is also specified.

Requirement The COLORRESPONSE= option must also be specified. If COLORRESPONSE= is not specified, COLORSTAT= is ignored.

Note This option is independent of the STAT= and RESPONSE= options.

DATALABEL
DATALABEL=*variable*
displays a label for each data point.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITS or LIMITSTAT, then the limit labels are also shown. Limit labels are shown when you use DATALABEL (without a variable).

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element. When

DATALABELPOS=DATA is in effect, the marker color. Otherwise, the Color attribute of the GraphDataText style element.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all marker symbols. If the marker symbol is not filled, then the data skin is applied to the outlines. Specify one of the following:

Table 8.7 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded

for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interaction You can use the MARKERATTRS= option to specify a filled marker symbol.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all dots from discrete category values. Specify a value from –0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the category axis is discrete.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interaction Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interactions If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

When FREQ= and MISSING are both specified, the MISSING option has no effect.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped dots.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interaction This option is ignored unless GROUP= is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default ASCENDING

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the dot plot in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is “Frequency”.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see “Line Attributes and Patterns” on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

For grouped data, the LineStyle and LineThickness attributes attributes of the GraphError style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITS= or LIMITSTAT= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITS=UPPER | LOWER | BOTH
specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending horizontally from each dot. Upper limits extend to the right of the dot and lower limits extend to the left of the dot. By default, no limits are displayed unless you specify either the LIMITS= or LIMITSTAT= option. Specify one of the following values:

BOTH
adds lower and upper limit lines to the plot.

LOWER
adds lower limit lines to the plot.

UPPER
adds upper limit lines to the plot.

Interaction Limit lines are displayed only when you specify STAT= MEAN.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines. Specify one of the following statistics:

CLM
confidence limits

STDDEV
standard deviation

STDERR
standard error

Default If you use the LIMITS= option, then the default for LIMITSTAT= is CLM. Otherwise, there is no default.

Interaction If you specify the LIMITSTAT= option only, then the default value for the LIMITS= option is BOTH. Limits lines are displayed only when you specify STAT=MEAN.

Tip To display limit labels, specify DATALABEL (without a variable).

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see “Marker Attributes and Symbols” on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style

element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interaction Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOLIMITCAPS
suppresses the serif cap on limit lines, if limit lines are displayed.

Interaction The appearance of the limit lines is controlled by the LIMITATTRS= option.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT= STDDEV or LIMITSTAT=STDERR. You can specify any positive number, including decimals.

Default 1

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable for each category value are displayed on the horizontal axis.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the horizontal axis. Specify one of the following:

FREQ
frequencies of the response or category variable.

The frequencies are calculated as follows:

- If you specify the RESPONSE= option, FREQ calculates the frequency of the response variable.

- If you do not specify the RESPONSE= option, FREQ calculates the frequency of the category variable.

MEAN
the mean of the response variable.

Interaction For STAT=MEAN to take effect, you must also specify the RESPONSE= option.

MEDIAN
the median of the response variable.

Interaction For STAT=MEDIAN to take effect, you must also specify the RESPONSE= option.

PERCENT
the percentage of the response or category variable.

The percentage is calculated as follows:

- If you specify the RESPONSE= option, PERCENT calculates the percentage of the sum of the response variable.
- If you do not specify the RESPONSE= option, PERCENT calculates the percentage of the frequency of the category variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the PCTLEVEL= option in the PROC SGPANEL statement.

You can use the PCTNDEC= option in the SGPANEL procedure statement to control the number of decimals to be used when calculating the percent values.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level will be zero.

SUM
the sum of the response variable.

This is the default value when you specify the RESPONSE= option.

Interaction For this value to take effect, you must also specify the RESPONSE= option.

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Restriction If you do not also specify the RESPONSE= option, then only the FREQ or PERCENT statistic is calculated (FREQ is the default). If you specify RESPONSE=, then you can use any of the statistics.

Interaction When the graph is generated, the statistic is appended to the variable name in the axis label and the legend (if it is created). However, if a label has been assigned to the variable, then the label appears in the axis label and legend instead of the statistic.

STATLABEL | NOSTATLABEL

specifies whether the response variable statistic is displayed in the axis and legend labels. STATLABEL forces the statistic to be displayed. NOSTATLABEL removes the statistic from the axis and legend labels.

Normally, the procedure displays the statistic along with the name of the response variable. However, when a custom label is assigned to the response variable, the procedure does not display the statistic. In each case, you can control whether the statistic is displayed.

Defaults The statistic is displayed for the response variable.

When a custom label is assigned to the response variable, the statistic is not displayed.

Interactions This option has no effect unless the RESPONSE= option is specified.

This option has no effect if you specify the axis label using the LABEL= option in an AXIS statement.

TIP=(*variable-list*) | NONE

specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)

a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE

suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

If your categorical chart is overlaid with other categorical charts, each chart can specify its own URL variable. The URL variable specified for each chart overlay is honored. This capability affects the following charts: DOT, HBAR, HLINE, VBAR, VLINE.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction If your plot is overlaid with other categorization plots that also specify WEIGHT=, then the first WEIGHT variable that you specified is used for all of the plots.

DROPLINE Statement

Creates one or more drop lines from data points to one or both axes. The line or lines can be horizontal, vertical, or both.

Requirement: The DROPLINE statement must be overlaid with another plot that is derived from data values that provide boundaries for the axis area.

Interaction: When the DROPLINE variable is the same as the response variable of a categorical chart that is specified in the procedure, the DROPLINE statement is ignored.

See: "About Drop Lines" on page 26

"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Syntax

DROPLINE X=*variable* | *x-axis-value*
Y=*variable* | *y-axis-value* </*options*>;

Summary of Optional Arguments

Appearance options

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all drop lines from discrete X or Y values.

LINEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the drop line.

TRANSPARENCY=*value*
specifies the degree of transparency for the drop line.

Axis options

DROPTO=BOTH | X | Y
specifies the axis to which the line is dropped.

NOCLIP
extends the plot axes to contain the drop lines.

Label options

LABEL=*variable* | *"text-string"*
creates labels for each drop line.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels.

LEGENDLABEL="*text-string*"
specifies a label that identifies the plot in the legend.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*variable* | *x-axis-value*
specifies the X coordinate of the drop line or drop lines. If you specify an *x-axis-value* that is a text string, enclose the string in quotation marks.

Requirements Values must agree in type with the X-axis data type. For example, you should use numeric SAS date or time values (or SAS date/time constants) for a time axis.

Unformatted numeric values do not map to a formatted discrete axis. If a drop line is drawn to a point on a formatted discrete X axis, the DROPLINE X= value must be a formatted value that appears on the X axis.

Y=*variable* | *y-axis-value*
specifies the Y coordinate of the drop line or drop lines. If you specify a *y-axis-value* that is a text string, enclose the string in quotation marks.

Requirements Values must agree in type with the Y-axis data type.

Unformatted numeric values do not map to a formatted discrete axis. If a drop line is drawn to a point on a formatted discrete Y axis, the DROPLINE Y= value must be a formatted value that appears on the Y axis.

Optional Arguments

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 8.8 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all drop lines from discrete X or Y values.

Default 0.0 (no offset)

Range −0.5 (left offset) to +0.5 (right offset) where 0.5 represents half the distance between discrete ticks.

Requirement This option is applicable only when the X or Y axis is discrete.

DROPTO=BOTH | X | Y
specifies the axis to which the line is dropped.

BOTH
draws droplines to both axes.

X
draws droplines to the X axis.

Y
draws droplines to the Y axis.

Default X

LABEL=*variable* | "*text-string*"
creates labels for each drop line.

variable
a variable for the label value.

Requirement In order to use a variable to label your drop lines, you must specify a variable in X=, Y=, or both in the DROPLINE statement. If you specify scalar values in X= and Y=, the specified label variable is ignored and a message is written to the log.

"text-string"
a text string for the label value.

Requirement In order to use a text string to label your drop line, you must specify scalar values in X= and Y=. If you specify a variable in X=, Y=, or both in the DROPLINE statement, the specified text string is ignored and a message is written to the log.

Default No label is displayed.

Note Space is reserved at the maximum end of the X axis to accommodate the length of the labels regardless of where the labels appear in the plot.

Tip You can use the OFFSETMAX= axis option to adjust the amount of space that is reserved on the X axis for the labels.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the LABEL option is also specified.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LEGENDLABEL="*text-string*"
specifies a label that identifies the plot in the legend. By default, the label "drop" is used.

Interaction This option has no effect unless you also specify the NAME= option.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the drop line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphReference style element in the current style.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOCLIP
extends the plot axes to contain the drop lines.

By default, if a drop line extends beyond the data range, the drop line is not drawn. When NOCLIP is in effect, the axis ranges are extended, if necessary, to accommodate the longest drop lines.

TRANSPARENCY=*value*
specifies the degree of transparency for the drop line.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Note This option does not affect the drop-line label.

Details

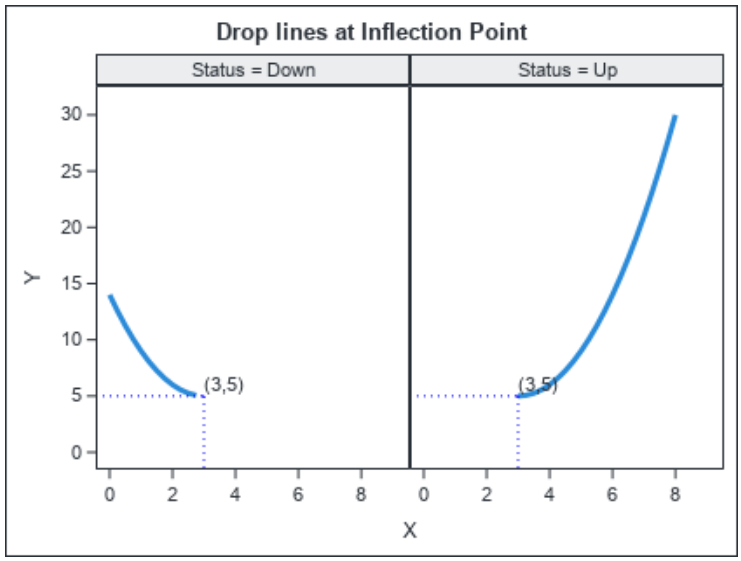
A drop line is always drawn perpendicular from the specified point to the X or Y axis. Axis offsets do not apply to drop lines, so they always meet the axis line.

The DROPTO= option controls whether a horizontal or vertical drop line is created. DROPTO=X specifies the horizontal axis for a vertical drop line, and DROPTO=Y specifies the vertical axis for a horizontal drop line. DROPTO=BOTH specifies both axes.

A DROPLINE statement must be used with another plot statement that establishes the axis scale for the DROPTO= points. For example, a drop line can be used with a scatter plot or a histogram. You can generate multiple drop lines by specifying a column for X and Y. The column type (numeric or string) must agree with the type of data presented on the axis.

Example

This example shows a DROPLINE statement originating from the point (X=3, Y=5) and dropping to both axes. The data form a parabola, which is divided in the panel depending on whether the curve is going down or up.




```
/* Create data set with X and Y variables
   that plot a parabola.*/
data test;
  length Status $4;
  do X=0 to 8 by 0.25;
    Y=(x-3)*(x-3) + 5;
    if X >= 3 then Status='Up';
    else Status='Down';
      output;
  end;
run;

/* Create the plot and drop lines.
   Specify a label and line attributes for the drop lines.*/
title "Drop lines at Inflection Point";
proc sgpanel data=test;
  panelby Status;
  series x=x y=y;
  dropline x=3 y=5 / dropto=both label="(3,5)"
     lineattrs=(color=blue pattern=dot);
  rowaxis min=0;
run;
title;
```

ELLIPSE Statement

Adds a confidence or prediction ellipse to a primary plot.

Restriction: The ELLIPSE statement must be used with another plot statement that uses numeric axes. The ellipse is typically used with a scatter plot. See "Plot Compatibility" on page 1780.

Interaction: The data for the ellipse is ignored when establishing axis scales. Each axis scale is determined by the other plots in the graph. This might result in the ellipse not being entirely displayed if its data range is not within the data ranges of the other plots.

This behavior differs from the ellipse in the SGPLOT procedure, which includes a CLIP option to specify that the ellipse data is ignored.

Example: "About Ellipse Plots" on page 42

Syntax

ELLIPSE X=*numeric-variable* Y=*numeric-variable* </option(s)>;

Summary of Optional Arguments

Appearance options

FILL

NOFILL
: specifies whether the area fill is visible.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

LEGENDLABEL="*text-string*"
: specifies a label that identifies the ellipse in the legend.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the outlines for the ellipse.

OUTLINE

NOOUTLINE
: specifies whether the outlines of the ellipse are visible.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Group options

GROUP=*variable*
: specifies a variable that is used to group the data.

NOMISSINGGROUP
: specifies that missing values of the group variable are not included in the plot.

Plot options

ALPHA=*numeric-value*
: specifies the confidence level for the ellipse.

FREQ=*numeric-variable*
: specifies a variable for the frequency count for each observation in the input data.

TYPE=MEAN | PREDICTED

specifies the type of ellipse.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*numeric-variable*
specifies a numeric variable for the X axis.

Y=*numeric-variable*
specifies a numeric variable for the Y axis.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the ellipse.

Default .05

Range 0.01 (complete confidence) to 0.99 (no confidence)

FILL | NOFILL
specifies whether the area fill is visible. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default The default status of the area fill is specified by the DisplayOpts attribute of the GraphEllipse style element in the current style.

Interactions Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction
If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note
This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips
Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

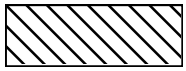
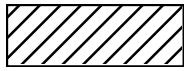
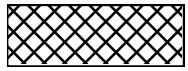
COLOR=*color*
specifies a color to use for the fill lines.

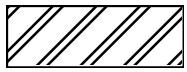
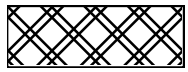
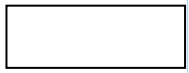
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.9 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

You can use the NOMISSINGGROUP option to omit missing values of the group variable from the plot.

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the ellipses, but the PATTERN= suboption of the OUTLINEATTRS= option could be used to assign the same line pattern to all ellipse outlines, letting outline color distinguish group values.

LEGENDLABEL="*text-string*"
specifies a label that identifies the ellipse in the legend. By default, the label describes the confidence value of the ellipse and the type of ellipse. For example, "95% Prediction Ellipse."

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the outlines for the ellipse. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OUTLINE | NOOUTLINE
specifies whether the outlines of the ellipse are visible. The OUTLINE option shows the outlines and hides the fill. The NOOUTLINE option hides the outlines.

Default The default status of the outlines is specified by the DisplayOpts attribute of the GraphEllipse style element in the current style.

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

TYPE=MEAN | PREDICTED
specifies the type of ellipse. MEAN specifies a confidence ellipse for the population mean. PREDICTED specifies a prediction ellipse for a new observation. Both ellipse types assume a bivariate normal distribution.

Default PREDICTED

Details

Statement Description

The ELLIPSE statement plots an ellipse with specified semimajor and semiminor axis lengths, a specified slope for the major axis, and the ellipse center. The ellipse center is the point of intersection of the semimajor and semiminor axes. The ELLIPSE statement computes an ellipse for a set of points that is specified by the X= on page 980 and Y= on page 980 columns and a confidence level that is specified by the ALPHA= option.

You can generate one or more ellipses by specifying a numeric column for any or all of options SEMIMAJOR=, SEMIMINOR=, XORIGIN=, and YORIGIN=. If any of the specified columns contain a missing value, then no ellipse is drawn. To request a vertical major axis, specify SLOPE=. (missing value) as a column value.

Confidence and Prediction Ellipses

Two types of ellipses can be computed for the input data (where observations correspond to points in a scatter plot). One is a confidence ellipse for the population mean (TYPE=MEAN), and the other is a prediction ellipse for a new observation (TYPE=PREDICT). Both assume a bivariate normal distribution.

Let $\overline{Z}$ and S be the sample mean and sample covariance matrix of a random sample of size *n* from a bivariate normal distribution with mean μ and covariance matrix Σ. The variable $\overline{Z} - \mu$ is distributed as a bivariate normal variate with mean zero and covariance $(\frac{1}{n})\Sigma$, and it is independent of S. Using Hotelling's T^2 statistic, which is defined as follows:

$$T^2 = n(\overline{Z} - \mu)' S^{-1}(\overline{Z} - \mu)$$

A $100(1-\alpha)\,\%$ confidence ellipse for μ is computed from the following equation:

$$\frac{n}{n-1}(\overline{Z} - \mu)' S^{-1}(\overline{Z} - \mu) = \frac{2}{n-2} F_{2,n-2}(1-\alpha)$$

where $F_{2,n-2}(1-\alpha)$ is the $(1-\alpha)$ critical value of an F distribution with degrees of freedom 2 and $n-2$.

A prediction ellipse is a region for predicting a new observation in the population. It also approximates a region containing a specified percentage of the population.

Denote a new observation as the bivariate random variable Z_{new}. The following variable:

$$Z_{new} - \overline{Z} = (Z_{new} - \mu) - (\overline{Z} - \mu)$$

is distributed as a bivariate normal variate with mean zero (the zero vector) and covariance $(1 + \frac{1}{n})\Sigma$, and it is independent of S. A $100(1-\alpha)\,\%$ prediction ellipse is then given by the following equation:

$$\frac{n}{n-1}(\overline{Z} - \mu)' S^{-1}(\overline{Z} - \mu) = \frac{2(n+1)}{n-2} F_{2,n-2}(1-\alpha)$$

The family of ellipses generated by different critical values of the F distribution has a common center (the sample mean) and common major and minor axis directions.

The shape of an ellipse depends on the aspect ratio of the plot. The ellipse indicates the correlation between the two variables if the variables are standardized (by dividing the variables by their respective standard deviations). In this situation, the ratio between the major and minor axis lengths is:

$$\sqrt{\frac{1+|r|}{1-|r|}}$$

In particular, if $r = 0$, then the ratio is 1, which corresponds to a circular confidence contour and indicates that the variables are uncorrelated. A larger value of the ratio indicates a larger positive or negative correlation between the variables.

ELLIPSE Default Data Tip

An ellipse generated by the ELLIPSE statement can display a default data tip in HTML output. Data tips are not enabled by default. To enable data tips, specify the IMAGEMAP option in an ODS GRAPHICS statement. The data tip displays the following roles:

- GROUP= (if specified)
- XORIGIN=

- YORIGIN=
- SEMIMAJOR=
- SEMIMINOR=
- SLOPE=

The ELLIPSE statement does not support data tip options that can be used to customize the data tip.

ELLIPSEPARM Statement

Adds an ellipse to the primary plot, specified by axis, slope, and origin parameters. The axis parameters are required.

Restriction: The ELLIPSEPARM statement must be overlaid with another plot that is derived from data values that provide boundaries for the axis area. It is typically overlaid with a scatter plot. See "Plot Compatibility" on page 1780.

Interaction: The data for the ellipse is ignored when establishing axis scales. Each axis scale is determined by the other plots in the graph. This might result in the ellipse not being entirely displayed if its data range is not within the data ranges of the other plots. This behavior differs from the ellipse in the SGPLOT procedure, which includes a CLIP option to specify that the ellipse data is ignored.

Syntax

ELLIPSEPARM SEMIMAJOR=*numeric-variable* | *positive-number* SEMIMINOR= | *numeric-variable* | *positive-number* </option(s)>;

Summary of Optional Arguments

Appearance options

FILL

NOFILL
: specifies whether to display a filled ellipse.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

LEGENDLABEL="*text-string*"
: specifies a label that identifies the ellipse in the legend.

LINEATTRS=*style-element*<(***options***)> | (*options*)

specifies the appearance of the outlines for the ellipse.

OUTLINE

NOOUTLINE
specifies whether the outlines of the ellipse are visible.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Plot options

SLOPE=*numeric-value* | *numeric-variable*
specifies the slope of the major axis for the ellipse.

XORIGIN=*numeric-value* | *numeric-variable*
specifies the X coordinate of the center of the ellipse.

YORIGIN=*numeric-value* | *numeric-variable*
specifies the Y coordinate of the center of the ellipse.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

SEMIMAJOR=*numeric-variable* | *positive-numeric-value*
specifies half the length of the major axis for the ellipse. Values are in the same units as the data. The SEMIMAJOR value can be greater than, smaller than, or equal to the SEMIMINOR value.

Restrictions The semimajor value must be positive. If a numeric variable is specified and the variable contains negative values, then no ellipses are drawn.

If the variable contains a missing value, the ellipse for that observation is not drawn. However, the remaining ellipses that have nonmissing values for this argument and for SEMIMINOR= are drawn.

SEMIMINOR=*numeric-variable* | *positive-numeric-value*
specifies half the length of the minor axis for the ellipse. Values are in the same units as the data. The SEMIMINOR value can be greater than, smaller than, or equal to the SEMIMAJOR value.

Restrictions The semiminor value must be positive. If a numeric variable is specified and the variable contains negative values, then no ellipses are drawn.

If the variable contains a missing value, the ellipse for that observation is not drawn. However, the remaining ellipses that have nonmissing values for this argument and for SEMIMAJOR= are drawn.

Optional Arguments

FILL | NOFILL

specifies whether to display a filled ellipse. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default: The default is specified by the DisplayOpts attribute of the GraphEllipse style element in the current style.

Interactions: Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)

specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults: For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interaction: This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN

enables the display of line fill patterns for the plot.

Interaction: If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips: Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

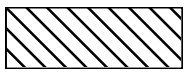
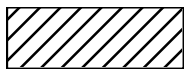
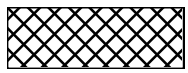
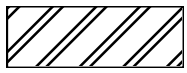
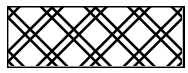
COLOR=*color*
specifies a color to use for the fill lines.

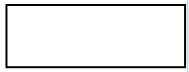
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.10 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

GROUP=*variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction The group value can be a character or numeric column. The GROUP= option can be used only when a numeric column or value is specified for the SEMIMAJOR= and SEMIMINOR= arguments, and when used, the SLOPE=, XORGIN= and YORIGIN= options.

Note If a constant value is specified for SEMIMAJOR=, SEMIMINOR= XORIGIN=, YORIGIN=, and SLOPE=, and GROUP= is also specified, an ellipse is drawn for each group value, but all of the ellipses overlap.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

You can use the NOMISSINGGROUP option to omit missing values of the group variable from the plot.

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the ellipses, but the PATTERN= suboption of the OUTLINEATTRS= option could be used to assign the same line pattern to all ellipse outlines, letting outline color distinguish group values.

LEGENDLABEL="*text-string*"
specifies a label that identifies the ellipse in the legend.

Default EllipseParm

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the outlines for the ellipse. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OUTLINE | NOOUTLINE
specifies whether the outlines of the ellipse are visible. The OUTLINE option shows the outlines and hides the fill. The NOOUTLINE option hides the outlines.

Default The default status of the outlines is specified by the DisplayOpts attribute of the GraphEllipse style element in the current style.

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

SLOPE=*numeric-value* | *numeric-variable*
specifies the slope of the major axis for the ellipse. The slope can be positive or negative.

Default 0

Note The slope value is in the data space and might not always be maintained in the screen space. Thus, setting SLOPE=1 does not always generate a 45-degree line on the screen.

Tips SLOPE=0 creates a major axis parallel to the X axis.

SLOPE= . (missing value) creates a major axis parallel to the Y axis.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

XORIGIN=*numeric-value* | *numeric-variable*
specifies the X coordinate of the center of the ellipse. Values are in the units of the data.

Default 0

Tip You can use the YORIGIN= option to specify the Y coordinate of the center of the ellipse.

YORIGIN=*numeric-value* | *numeric-variable*
specifies the Y coordinate of the center of the ellipse. Values are in the units of the data.

Default 0

Tip You can use the XORIGIN= option to specify the X coordinate of the center of the ellipse.

Details

Statement Description

The ELLIPSEPARM statement plots an ellipse with specified semimajor and semiminor axis lengths, a specified slope for the major axis, and the ellipse center. The ellipse center is the point of intersection of the semimajor and semiminor axes. The ELLIPSEPARM statement does not perform computations on input data to derive the location and shape of the ellipse. Another plot that is derived from data values that provide boundaries for the axis area must be included in the layout. The ellipse is typically overlaid with a scatter plot. See ELLIPSE Statement for information about computed ellipses.

You can generate a single ellipse by specifying a constant for each of the following arguments: SEMIMAJOR=, SEMIMINOR=, SLOPE=, XORIGIN=, YORIGIN=. The SLOPE=, XORIGIN=, and YORIGIN= arguments are set to a constant by default.

You can generate multiple ellipses by specifying a numeric column for any or all of those arguments. If any of the SEMIMAJOR=, SEMIMINOR=, XORIGIN=, or YORIGIN= constants or columns contains a missing value, then no ellipse is drawn. To request a vertical major axis, specify SLOPE=. (missing value) as a constant or column value.

ELLIPSEPARM Default Data Tip

An ellipse generated by the ELLIPSEPARM statement can display a default data tip in HTML output. Data tips are not enabled by default. To enable data tips, specify the IMAGEMAP option in an ODS GRAPHICS statement. The data tip displays the following roles:

- GROUP= (if specified)
- XORIGIN=
- YORIGIN=
- SEMIMAJOR=
- SEMIMINOR=
- SLOPE=

The ELLIPSEPARM statement does not support data tip options that can be used to customize the data tip.

FRINGE Statement

Creates a fringe plot on the X axis of an X-Y plot.

Interaction: Fringe plots can be overlaid with all plots except with box plots and categorical charts (bar charts, line plots, and dot plots). See "Plot Compatibility" on page 1780.

Example: "About Fringe Plots" on page 20

Syntax

FRINGE *numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

HEIGHT=*dimension*<**units**>
: specifies the height of the fringe lines.

LINEATTRS=*style-element*<(***options***)> | (*options*)

specifies the appearance of the fringe lines.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

LEGENDLABEL="*text-string*"
specifies a label that identifies the fringe plot in the legend.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Argument

numeric-variable
specifies the variable that provides the X coordinates of the data values.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes. Each distinct group value is represented in the graph by a different line color. Line patterns are not changed across groups.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

You can use the NOMISSINGGROUP option to omit missing values of the group variable from the plot.

HEIGHT=*dimension*<units>
specifies the height of the fringe lines. You can also specify the unit of measurement. The default unit is pixels.

The following table contains the units that are available:

Table 8.11 *Measurement Units*

Unit	Description
CM	centimeters
IN	inches
MM	millimeters
PCT or %	percentage
PT	point size, calculated at 72 dots per inch
PX	pixels

Default 10px

LEGENDLABEL="*text-string*"
specifies a label that identifies the fringe plot in the legend.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fringe lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role,

using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)

applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*

specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

GRADLEGEND Statement

Generates a gradient legend that maps the data range of a response variable to a range of colors. You can use up to four GRADLEGEND statements in a procedure. The GRADLEGEND statement is used in conjunction with the COLORRESPONSE= option. (The COLORRESPONSE= option is specified in the plot statement and indicates the response variable that is used to map the colors.)

See: "Using Gradient Color Legends" on page 1793

Syntax

GRADLEGEND <"*name*"> </*options*>;

Summary of Optional Arguments

Appearance options

BORDER

NOBORDER
: specifies whether a border is visible around the legend.

INTEGER
: specifies that integers are used for the gradient legend.

NOTITLE
: removes the default title. The default title is the name of the response variable.

OUTERPAD=*numeric-value*<*units*> | (*pad-options*)
: specifies the amount of extra space that is added outside the legend border.

POSITION=TOP | BOTTOM | LEFT | RIGHT
: specifies the position of the legend within the graph.

TITLE="*text-string*"
: specifies a title for the legend.

TITLEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the title.

Legend options

"name"
: specifies the name of the plot that you want to include in the legend.

Scale options

EXTRACTSCALE

EXTRACTSCALE=DEFAULT | SCIENTIFIC

extracts a scale factor from the tick values and uses it to reduce the tick value width.

Optional Arguments

"name"

specifies the name of the plot that you want to include in the legend. The name that you specify must correspond to a value that you entered for the NAME= option in a plot statement. The plot statement must also specify the COLORRESPONSE= option.

Default If no name is specified, the legend references whichever plot statement specifies the COLORRESPONSE= option. If the procedure contains more than one plot with a COLORRESPONSE= option and you do not specify a plot name, then the legend attempts to reference both or all of these plots. The resulting legend might be hard to read.

Restriction Only one name can be specified. If you want a continuous legend for more than one plot, you can use multiple GRADLEGEND statements. You can use up to four GRADLEGEND statements in a procedure.

BORDER | NOBORDER

specifies whether a border is visible around the legend.

Default NOBORDER

EXTRACTSCALE
EXTRACTSCALE=DEFAULT | SCIENTIFIC

extracts a scale factor from the tick values and uses it to reduce the tick value width. The scale used is appended to the legend title as shown in the following example.

```
Total Sales (millions)
```

For long legend titles, if the scale does not fit the available space, then the title is truncated, and the scale is appended to the truncated title. Ellipses indicate that the label was truncated as shown in the following example.

```
Total Sales for the Fourth Quarter Of ... (millions)
```

In extreme cases where the title does not fit even with truncation, the title is dropped.

You can also specify whether to use a named scale or a scientific-notation scale.

DEFAULT

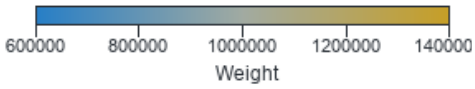
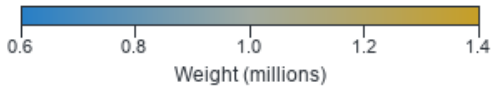
extracts a named scale.

A named scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as 10^*n*) for values over 999 trillion. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of 1/10 (10^–*n*) for values less than 1 trillionth.

SCIENTIFIC
extracts a scientific-notation scale.

A scientific-notation scale is a multiple of 10 expressed as 10^*n* for values greater than 1, or a multiple of 1/10 expressed as 10^–*n* for values less than 1.

The following examples show a gradient legend before and after EXTRACTSCALE= is specified:

Default Scale	EXTRACTSCALE Specified
	

Default DEFAULT

Restriction The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale.

INTEGER
specifies that integers are used for the gradient legend.

NOTITLE
removes the default title. The default title is the name of the response variable.

Default The default title is displayed.

OUTERPAD=*numeric-value<units>* | (*pad-options*)
specifies the amount of extra space that is added outside the legend border. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

numeric-value<units>
specifies a dimension to use for the extra space at the left, right, top, and bottom of the legend border.

(pad-options)
specifies a space-separated list of pad options, enclosed in parentheses.

Specify one or more of the following name-value pair options:

LEFT=*numeric-value<units>*	specifies the amount of extra space added to the left side.
RIGHT=*numeric-value<units>*	specifies the amount of extra space added to the right side.
TOP=*numeric-value<units>*	specifies the amount of extra space added to the top.
BOTTOM=*numeric-value<units>*	specifies the amount of extra space added to the bottom.

Note Sides that are not assigned padding are padded with the default amount.

Tip Use *pad-options* to create non-uniform padding.

Default No padding

Note The default units for *dimension* are pixels. If you want to specify values in other units, then you must specify the units with the value. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Example "Example: Gradient Legend That Uses the OUTERPAD= Option" on page 276

POSITION=TOP | BOTTOM | LEFT | RIGHT
specifies the position of the legend within the graph.

Default RIGHT

Notes By default, if you use more than one GRADLEGEND statement, then each legend is placed in a different position.

If you specify more than one legend with the same position, then those legends are placed at that position.

TITLE="*text-string*"
specifies a title for the legend.

Default If you do not specify this option, then the name of the COLORRESPONSE= variable is displayed as the title.

TITLEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the title. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

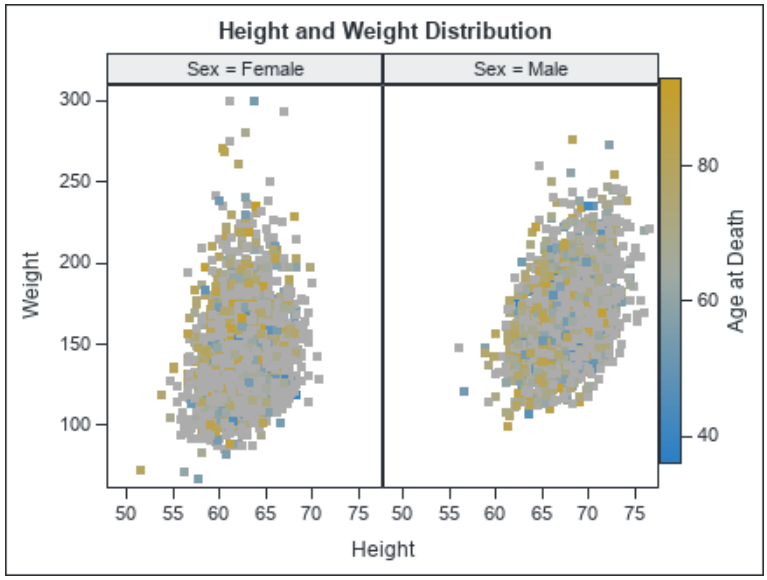
For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
TITLEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TITLEATTRS=GraphTitleText
```

Example: Gradient Legend That Uses the OUTERPAD= Option

The following example shows a gradient legend that maps an age scale to a color gradient. The OUTERPAD= on page 434 option adds padding to the top and bottom of the gradient.




```
title "Height and Weight Distribution";
proc sgpanel data=sashelp.heart;
  panelby sex;
  scatter x=height y=weight /
       colorresponse=ageatdeath  name="scatter"
       markerattrs=(symbol=squarefilled size=6px);
  gradlegend "scatter" /
      outerpad=(top=20px bottom=20px);
run;
title;
```

HBAR Statement

Creates a bar chart that summarizes the values of a category variable.

Restrictions: The HBAR statement can be combined only with other categorization plot statements in the SGPANEL procedure. See "Plot Compatibility" on page 1780.

When category plots are overlaid, all of the plots must use the same category variable.

Tips: The HBAR statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

Bar charts can be combined with basic plot types using the HBARBASIC and VBARBASIC statements.

Examples: "About Bar Charts" on page 53

"Example 3: Creating a Panel of Bar Charts" on page 835

Syntax

HBAR *category-variable* </option(s)>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
: specifies whether the baseline is displayed.

FILL

NOFILL
: specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLENDCOLOR=*color*
: specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
: specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
: specifies the thickness of the bars when the category data is numeric.

NOZEROBARS
: suppresses zero-length bars.

OUTLINE

NOOUTLINE
specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each bar.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
specifies the fit policy for the bar labels.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend.

SEGLABEL
displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

STATLABEL

NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels.

Limit options

LIMITATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITS=BOTH | LOWER | UPPER
specifies which limit lines to display.

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR.

Plot options

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

MISSING
accepts a missing value as a valid category value.

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

WEIGHT=*numeric-variable*
specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Statistics options

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN

specifies the statistic to use for computing the response colors when the COLORRESPONSE= option is specified.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the horizontal axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the bars. The variable generates the midpoints to which each observation in the data set contributes.

Interval bar charts are supported when the category axis is set to TYPE=LINEAR.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default .05

Interactions This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Defaults .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

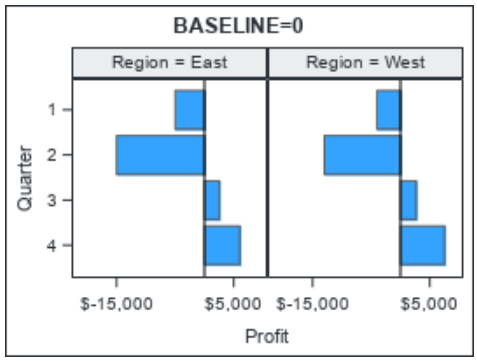
Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*

specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interactions If GROUPDISPLAY=STACK is specified, this option is ignored.

When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CATEGORYORDER=RESPASC | RESPDESC

specifies the order in which the categories are arranged. By default, the plot is sorted in ascending order based on the category values. You can change the order so that it is based on the response values.

Specify one of the following values:

RESPASC
sorts by the response values in ascending order.

RESPDESC
sorts by the response values in descending order.

Default The plot is sorted in ascending order based on the category values.

Restriction This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Requirement This option requires that you configure the panel to use either one column or one row, depending on the orientation of your charts. Use the ROWS= option or the COLUMNS= option in the PANELBY statement. If you do not use this option and your graph contains multiple cells, the specified sort order is not correctly applied to all cells.

Interactions When a group variable is used with the CATEGORYORDER= option, the category order is not affected by the value of the groups. The categories are always sorted by the response statistic at a category level.

When this option and the GROUPORDER= option are both specified, the GROUPORDER= option is ignored.

Notes If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

When two or more observations have the same response values, there is no guarantee that the tied values will be sub-sorted.

CLUSTERWIDTH=*numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the

maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)

specifies a color ramp that is to be used to determine response colors.

style-element

specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)

specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see “Color-Naming Schemes” on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified or COLORSTAT= must be FREQ or PCT.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors when the COLORRESPONSE= option is specified. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

SUM
sum values for the color response

MEAN
mean values for the color response

MEDIAN
median values for the color response

Default SUM when option COLORRESPONSE= is also specified.

Requirement The COLORRESPONSE= option must also be specified. If COLORRESPONSE= is not specified, COLORSTAT= is ignored.

Note This option is independent of the STAT= and RESPONSE= options.

DATALABEL
DATALABEL=*variable*
displays a label for each bar.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITS or LIMITSTAT, then the limit labels are also shown. Limit labels are shown when you use DATALABEL by itself without a variable.

This option is supported with the GROUP= option only when GROUPDISPLAY=CLUSTER is specified. Otherwise, the label is not drawn. The labels are shown if you also specify the DATALABELPOS= option.

By default, the data label fit policy is to show the labels unless they collide. As a result, the labels sometimes might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
specifies the fit policy for the bar labels. Specify one of the following values:

NONE
specifies that no fit policy is implemented for the bar labels.

By default, the fit policy is to show the labels unless they collide. As a result, the labels might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

INSIDEPREFERRED
places bar labels inside the bars, when possible.

Labels are placed outside the bars in the following cases:

- If a label cannot fit within the length of its respective bar, then that label is moved outside the bar.
- If the font height for any label exceeds the bar width, then all labels are positioned outside the bars.
- If SEGLABEL is specified, the INSIDEPREFERRED policy is ignored and the labels are positioned outside the bars.

Default Show the labels unless they collide.

Interaction This option has no effect unless DATALABEL= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 8.12 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.

Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL
specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

COLOR=*color*
specifies a color to use for the fill lines.

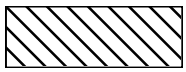
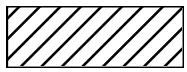
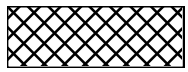
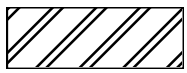
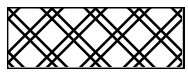
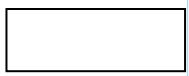
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With

grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.13 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions
For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note
This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interactions If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

When FREQ= and MISSING are both specified, the MISSING option has no effect.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP=is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default ASCENDING

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

INTERVALBARWIDTH=*numeric-value<units>*
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is “Frequency”.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes attributes of the GraphError style element in the current style.

Interaction This option has no effect unless you also specify either the LIMITS= or LIMITSTAT= option.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITS= or LIMITSTAT= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITS=BOTH | LOWER | UPPER
specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending from each bar. Upper limits extend to the right of the bar and lower limits extend to the left of the bar. By default, no limits are displayed unless you specify either the LIMITS= or LIMITSTAT= option. If you specify the LIMITSTAT= option only, then LIMITS=BOTH is the default. Specify one of the following values:

BOTH
adds lower and upper limit lines to the plot.

LOWER
adds lower limit lines to the plot.

UPPER
adds upper limit lines to the plot.

Default By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interactions Limit lines are displayed only when you specify STAT= MEAN.

If you use the GROUP= option in the plot statement, the LIMITS= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines. Specify one of the following statistics:

CLM
confidence limits

STDDEV
standard deviation

STDERR
standard error

Default If you specify the LIMITS= option, then the default for LIMITSTAT= is CLM. Otherwise, there is no default.

Interactions If you specify the LIMITSTAT= option only, then the default value for the LIMITS= option is BOTH.

Limits lines are displayed only when you specify STAT=MEAN.

If you use the GROUP= option in the plot statement, the LIMITSTAT= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

Tip To display limit labels, specify DATALABEL (without a variable).

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

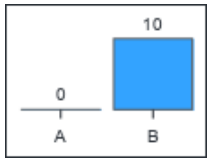
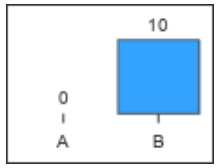
Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS
suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR. You can specify any positive number, including decimals.

Default 1

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes

of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphOutlines style element

- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

This option is ignored if the DATASKIN= option is also specified.

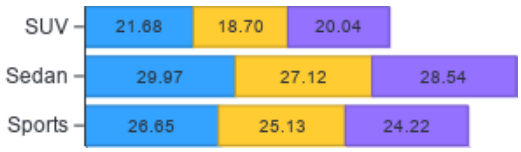
RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the horizontal axis.

SEGLABEL
displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element in the current style.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

NONE
no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP
does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN
drops any bar segment label that does not fit within its segment.

The label text height must not exceed the bar width, and the label length must not exceed the segment length.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM

specifies the statistic for the horizontal axis. Specify one of the following:

FREQ

frequencies of the response or category variable.

The frequencies are calculated as follows:

- If you specify the RESPONSE= option, FREQ calculates the frequency of the response variable.
- If you do not specify the RESPONSE= option, FREQ calculates the frequency of the category variable.

MEAN

the mean of the response variable.

Interaction For STAT=MEAN to take effect, you must also specify the RESPONSE= option.

MEDIAN

the median of the response variable.

Interaction For STAT=MEDIAN to take effect, you must also specify the RESPONSE= option.

PERCENT

the percentage of the response or category variable.

The percentage is calculated as follows:

- If you specify the RESPONSE= option, PERCENT calculates the percentage of the sum of the response variable.
- If you do not specify the RESPONSE= option, PERCENT calculates the percentage of the frequency of the category variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the PCTLEVEL= option in the PROC SGPANEL statement.

You can use the PCTNDEC= option in the SGPANEL procedure statement to control the number of decimals to be used when calculating the percent values.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level will be zero.

SUM

the sum of the response variable.

This is the default value when you specify the RESPONSE= option.

Interaction For this value to take effect, you must also specify the RESPONSE= option.

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Restriction If you do not also specify the RESPONSE= option, then only the FREQ or PERCENT statistic is calculated (FREQ is the default). If you specify RESPONSE=, then you can use any of the statistics.

Interaction When the graph is generated, the statistic is appended to the variable name in the axis label and the legend (if it is created). However, if a label has been assigned to the variable, then the label appears in the axis label and legend instead of the statistic.

STATLABEL | NOSTATLABEL

specifies whether the response variable statistic is displayed in the axis and legend labels. STATLABEL forces the statistic to be displayed. NOSTATLABEL removes the statistic from the axis and legend labels.

Normally, the procedure displays the statistic along with the name of the response variable. However, when a custom label is assigned to the response variable, the procedure does not display the statistic. In each case, you can control whether the statistic is displayed.

Defaults The statistic is displayed for the response variable.

When a custom label is assigned to the response variable, the statistic is not displayed.

Interactions This option has no effect unless the RESPONSE= option is specified.

This option has no effect if you specify the axis label using the LABEL= option in an AXIS statement.

TIP=(*variable-list*) | NONE

specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)

a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE

suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces the information that is displayed by default. However, if RESPONSE= is not specified and TIP= is other than NONE, the computed frequency is added to the specified tips.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

If your categorical chart is overlaid with other categorical charts, each chart can specify its own URL variable. The URL variable specified for each chart overlay is honored. This capability affects the following charts: DOT, HBAR, HLINE, VBAR, VLINE.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction If your plot is overlaid with other categorization plots that also specify WEIGHT=, then the first WEIGHT variable that you specified is used for all of the plots.

HBARBASIC Statement

Creates a horizontal bar chart that is compatible with other categorization charts as well as basic plots, such as scatter and series plots, and box plots.

Notes: When using the HBARBASIC statement, axes are not guaranteed to be uniform across BY groups.

The bars are sorted in data order based on the category values. The sort order cannot be changed. This arrangement of the bars differs from the HBAR statement, which by default sorts bars in alphabetical order based on the category values.

Tip: The HBARBASIC statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Bar Charts" on page 53

Syntax

HBARBASIC *category-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed.

FILL

NOFILL
specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
specifies the thickness of the bars when the category data is numeric.

NOZEROBARS
suppresses zero-length bars.

OUTLINE

NOOUTLINE
specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

Data tip options

TIP=(*role-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL
displays the bar statistic value for each bar.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL option.

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
specifies the fit policy for the bar labels.

DATALABELFORMAT=*format*
specifies the text format used to display the bar label.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend.

SEGLABEL
displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Plot options

MISSING
accepts a missing value as a valid category value.

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Statistics options

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors.

STAT=FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic for the horizontal axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the bars. The variable generates the midpoints to which each observation in the data set contributes.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Defaults .8

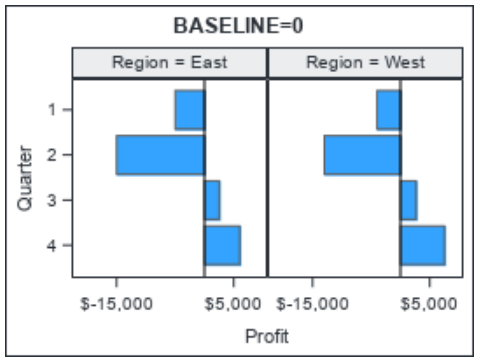
1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interaction When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CLUSTERWIDTH=*numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified or COLORSTAT= must be FREQ or PCT.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Interaction This option is ignored if COLORSTAT=FREQ or COLORSTAT=PCT.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

When COLORRESPONSE= is specified, the following values are valid:

SUM
sum values for the color response

MEAN
mean values for the color response

MEDIAN
median values for the color response

Default SUM when option COLORRESPONSE= is also specified.

Requirement The COLORRESPONSE= option must also be specified. If COLORRESPONSE= is not specified, COLORSTAT= is ignored.

Note This option is independent of the STAT= and RESPONSE= options.

DATALABEL
displays the bar statistic value for each bar. For grouped clustered bars, each bar is labeled with the summarized value of the bar. For grouped stacked bars, the segmented bar is labeled with the accumulated, summarized value of all the bar segments.

Default No label is shown

Interaction By default, the data label fit policy is to show the labels unless they collide. As a result, the labels sometimes might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

Tip The font and color attributes for the label are specified by the DATALABELATTRS= option. The text format is specified by the DATALABELFORMAT= option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
specifies the fit policy for the bar labels. Specify one of the following values:

NONE
specifies that no fit policy is implemented for the bar labels.

By default, the fit policy is to show the labels unless they collide. As a result, the labels might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

INSIDEPREFERRED
places bar labels inside the bars, when possible.

Labels are placed outside the bars in the following cases:

- If a label cannot fit within the length of its respective bar, then that label is moved outside the bar.
- If the font height for any label exceeds the bar width, then all labels are positioned outside the bars.
- If SEGLABEL is specified, the INSIDEPREFERRED policy is ignored and the labels are positioned outside the bars.

Default Show the labels unless they collide.

Interaction This option has no effect unless DATALABEL is also specified.

DATALABELFORMAT=*format*
specifies the text format used to display the bar label.

Default The column format assigned to the RESPONSE= column, or BEST6 if no format is assigned.

Interaction This option has no effect unless DATALABEL= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 8.14 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	**PRESSED**	**SHEEN**

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.

Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL
specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interaction This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction
If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note
This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips
Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

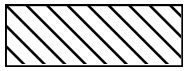
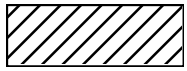
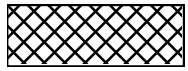
COLOR=*color*
specifies a color to use for the fill lines.

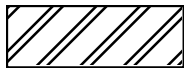
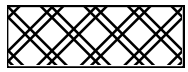
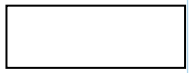
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.15 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, “Using Attribute Maps to Control Visual Attributes,” on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP=is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted

values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INTERVALBARWIDTH=*numeric-value<units>*
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is "Frequency".

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

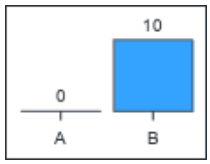
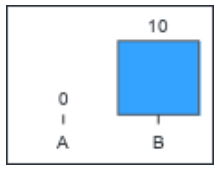
MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS
suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphOutlines style element
- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

This option is ignored if the DATASKIN= option is also specified.

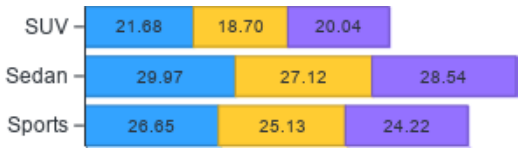
RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed for each value on the horizontal axis.

SEGLABEL
displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element in the current style.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

NONE
no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP
does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN
drops any bar segment label that does not fit within its segment.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

STAT=FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic for the horizontal axis.

For bar charts with no RESPONSE= variable, the following values are valid:

FREQ
frequency count

PERCENT
percentages between 0 and 100

Alias PCT

PROPORTION
proportions between 0 and 1

For bar charts with a RESPONSE= variable, the following values are valid:

SUM
sum values for the response

MEAN
mean values for the response

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Interaction When this option is used with the GROUP=group option, the specified statistic is computed for each segment that is created for the unique group values.

TIP=(*role-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*role-list*)
a space-separated list of unique chart roles enclosed in parentheses.

The available roles for TIP are CATEGORY, GROUP, and RESPONSE. Specify one or more of these role names in *role-list*. Data tips are displayed using the data obtained from the specified roles.

Note: You must specify the GROUP and RESPONSE roles for the chart in order to use those roles for data tips.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP=ON;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(category response)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

HBARPARM Statement

Creates a horizontal bar chart based on a pre-summarized response value for each unique value of the category variable. You can also assign variables to the upper and lower limits.

Requirements: Parameterized bar charts require a response variable in addition to the category variable. The response variable contains pre-summarized computed values such as a sum or a mean for each unique value of the category variable. This is the key distinction between HBARPARM and HBAR.

The data must contain only one response value per unique category variable. If more than one value is found, a warning is written to the SAS log, and the graph might produce unpredictable results.

Interaction: The HBARPARM statement can be combined only with other basic plot statements in the SGPANEL procedure. See "Plot Compatibility" on page 1780.

Note: The bars are sorted in data order based on the category values. The sort order cannot be changed. This arrangement of the bars differs from the HBAR statement, which by default sorts bars in alphabetical order based on the category values.

Tip: The HBARPARM statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

Example: "About Bar Charts" on page 53

Syntax

HBARPARM CATEGORY=*category-variable* RESPONSE=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
specifies the thickness of the bars when the category data is numeric.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend.

NOZEROBARS
suppresses zero-length bars.

OUTLINE

NOOUTLINE
specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

Bar options

FILL

NOFILL
specifies whether the bars are filled.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
specifies the fit policy for the bar labels.

SEGLABEL
displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Limit options

LIMITATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*

specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITLOWER=*numeric-variable*
specifies values for the lower endpoints on the limit lines.

LIMITUPPER=*numeric-variable*
specifies values for the upper endpoints on the limit lines.

Plot options

MISSING
accepts a missing value as a valid category value.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME=*"text-string"*
specifies a name for the plot.

Required Arguments

CATEGORY=*category-variable*
specifies the variable that categorizes the data. All values are treated as discrete values. The input data for this variable should contain unique values. When the category values are not unique, a warning is logged, and multiple bars are superimposed at the duplicated category values. The CATEGORY axis is always discrete.

Interval bar charts are supported when the category axis is set to TYPE=LINEAR.

RESPONSE=*numeric-variable*
specifies a numeric response variable. The input data is expected to be pre-summarized computed values (sum, mean, and so on).

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Defaults .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

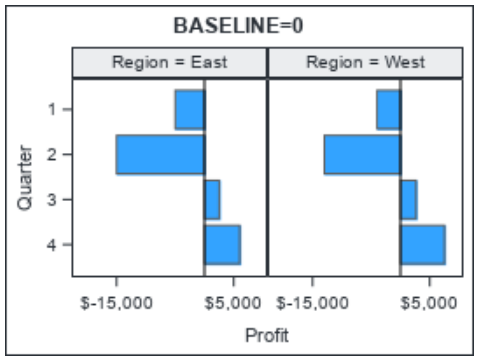
Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*

specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interaction When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style

element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CLUSTERWIDTH=*numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)

specifies a color ramp that is to be used to determine response colors.

style-element

specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

DATALABEL
DATALABEL=*variable*
displays a label for each data point.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

This option is supported with the GROUP= option only when GROUPDISPLAY=CLUSTER is specified. Otherwise, the label is not drawn. The labels are shown if you also specify the DATALABELPOS= option.

If you also specify LIMITLOWER or LIMITUPPER, then the limit labels are also shown. Limit labels are shown when you use DATALABEL (without a variable).

Note By default, the data label fit policy is to show the labels unless they collide. As a result, the labels sometimes might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
specifies the fit policy for the bar labels. Specify one of the following values:

NONE
specifies that no fit policy is implemented for the bar labels.

By default, the fit policy is to show the labels unless they collide. As a result, the labels might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

INSIDEPREFERRED
places bar labels inside the bars, when possible.

Labels are placed outside the bars in the following cases:

- If a label cannot fit within the length of its respective bar, then that label is moved outside the bar.
- If the font height for any label exceeds the bar width, then all labels are positioned outside the bars.

- If SEGLABEL is specified, the INSIDEPREFERRED policy is ignored and the labels are positioned outside the bars.

Default Show the labels unless they collide.

Interaction This option has no effect unless DATALABEL= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 8.16 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.

Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL
specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

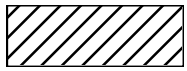
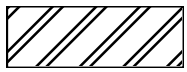
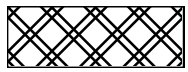
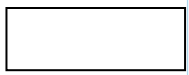
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.17 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions

For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note

This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT

specifies the fill type that is applied to the chart.

SOLID

each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP=is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

INTERVALBARWIDTH=*numeric-value*<*units*>
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITLOWER= or LIMITUPPER= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITLOWER=*numeric-variable*
specifies values for the lower endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default The lower segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions If LIMITUPPER= is also specified, then the plot displays the lower and upper segments of the limit lines.

If you use the GROUP= option in the plot statement, the LIMITLOWER= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITUPPER=*numeric-variable*
specifies values for the upper endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default The upper segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions If LIMITLOWER= is also specified, then the plot displays the lower and upper segments of the limit lines.

If you use the GROUP= option in the plot statement, the LIMITUPPER= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

Tip To display limit labels, specify DATALABEL (without a variable).

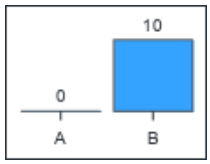
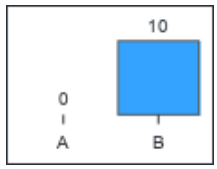
MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS
suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style

element, and the LineThickness attribute of the GraphOutlines style element

- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

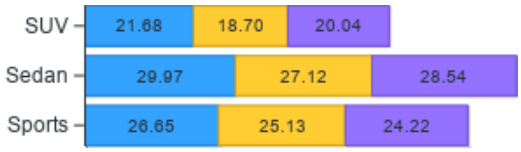
This option is ignored if the DATASKIN= option is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SEGLABEL
displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element in the current style.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

NONE
no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP
does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN
drops any bar segment label that does not fit within its segment.

The label text height must not exceed the bar width, and the label length must not exceed the segment length.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Notes When TIP= is not specified, the CATEGORY, RESPONSE, COLORRESPONSE, and GROUP variables are displayed, if applicable, when you hover over a bar. The ERRORUPPER or ERRORLOWER variable is displayed when you hover over an error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a bar. No variables are displayed when you hover over an error bar or error bar cap.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see “Using the ODS GRAPHICS Statement” on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

HBOX Statement

Creates a horizontal box plot that shows the distribution of your data.

Restriction: This plot has compatibility restrictions when used with other plots. See Table 13.2 on page 1781.

See: "Visual Description of Box Plot Percentile Boundaries" on page 360

Example: "About Box Plots" on page 47

Syntax

HBOX *numeric-analysis-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BOXWIDTH=*numeric-value*
: specifies the width of the box.

CAPSCALE=*numeric-value*
: specifies a positive number to use as a multiplier for changing the default width of the whisker caps.

CAPSHAPE=BRACKET | LINE | SERIF NONE
: specifies the shape of the whisker cap lines.

CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
: specifies that a connect line joins a statistic from box to box.

CONNECTATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the lines that connect multiple boxes.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all boxes from the discrete tick marks.

EXTREME
: specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified.

FILL

NOFILL

specifies whether the boxes are filled with color.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency.

FILLPATTERN
enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern.

INTERVALBOXWIDTH=*numeric-value*
specifies the box width when an interval category (Y) variable is specified.

LINEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the box outlines.

MEANATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the marker that represents the mean in the box.

MEDIANATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the line that represents the median.

NOCAPS
hides the cap lines for the whiskers.

NOMEAN
hides the mean marker.

NOMEDIAN
hides the median line.

NOOUTLIERS
hides the outliers from the plot.

NOTCHES
specifies that the boxes be notched.

OUTLIERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the marker that represents the outliers.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

WHISKERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the whisker and cap lines.

WHISKERPCT=*number*
specifies the whisker length, in percentile units.

Data tip options

TIP=(*role-list*) | NONE
specifies the information to display when the cursor is positioned over a box or whisker in the box plot.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the cluster width as a ratio of the midpoint spacing.

GROUP=*variable*
: specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
: specifies how to display grouped boxes.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
: adds data labels for the outlier markers.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

LABELFAR
: specifies that only the far outliers have data labels.

LEGENDLABEL="*text-string*"
: specifies a label that identifies the box plot in the legend.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

Plot options

CATEGORY=*category-variable*
: specifies the category variable for the plot.

DISPLAYSTATS=ALL | (*statistics-list*)
: specifies statistics to be displayed for each box.

FREQ=*numeric-variable*
: specifies a variable for the frequency count for each observation in the input data.

MISSING
: accepts a missing value as a valid category value.

PERCENTILE=1 | 2 | 3 | 4 | 5
: specifies a method for computing the percentiles for the plot.

SPREAD
: relocates outlier points that have identical values to prevent overlapping.

WEIGHT=*numeric-variable*
: specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
: specifies a name for the plot.

Required Argument

numeric-analysis-variable
specifies the analysis variable for the plot. If you do not specify the CATEGORY= option, then one box is created for the analysis variable.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

BOXWIDTH=*numeric-value*
specifies the width of the box. Specify a value between 0.0 (0% of the available width) and 1.0 (100% of the available width).

Defaults 0.4

When GROUP is specified, the default box width is 0.6.

CAPSCALE=*numeric-value*
specifies a positive number to use as a multiplier for changing the default width of the whisker caps. For example, CAPSCALE=2 doubles the default cap width and CAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction This option has no effect if NOCAPS is specified or if CAPSHAPE=NONE.

Tips The thickness (height) of a SERIF, LINE, or BRACKET cap honors the line thickness specified in the WHISKERATTRS= option. The cap pattern is always solid.

Use the CAPSHAPE= option to specify the line shape of the caps.

CAPSHAPE=BRACKET | LINE | SERIF NONE
specifies the shape of the whisker cap lines. Specify one of the following values:

BRACKET
displays a straight line with brackets.

LINE
displays a straight line.

SERIF
displays a short straight line.

NONE
does not display a cap.

Default SERIF

CATEGORY=*category-variable*
specifies the category variable for the plot. A box plot is created for each distinct value of the category variable.

If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete.

The default ordering of the tick values is ascending formatted order of the *category-variable*. To change this ordering, you can use the DISCRETEORDER=DATA option in the category axis statement.

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the midpoint spacing. Specify a value from 0.1 (narrowest) to 1.0 (widest).

Default 0.7

Interaction This option is applicable only when a GROUP is in effect and the category axis is discrete.

CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
specifies that a connect line joins a statistic from box to box.

Interaction This option applies only when the CATEGORY option is used to generate multiple boxes.

Tip You can use the CONNECTATTRS option to specify attributes for the connect line.

CONNECTATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the lines that connect multiple boxes. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphConnectLine style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interactions This option takes effect only if the CONNECT= option is also specified.

This option is ignored if the GROUP= option is also specified.

Examples
```
CONNECTATTRS=(Color="light green" Pattern=MediumDash Thickness=4)
```

This example specifies a style element:
```
CONNECTATTRS=GraphData3
```

DATALABEL
DATALABEL=*variable*
adds data labels for the outlier markers. If you specified a variable, then the values for that variable are used for the data labels. If you did not specify a variable, then the values of the analysis variable are used.

Note This option has no effect unless the plot contains outlier points.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

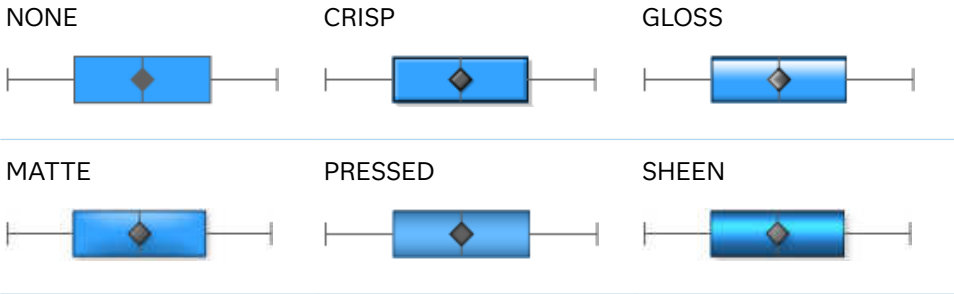
Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled boxes. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 8.18 *DATASKIN Options for Box Plots*



Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interaction If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all boxes from the discrete tick marks.

Specify a value from -0.5 (left offset) to +0.5 (right offset). If you specify a value outside of this range, an error message appears in the SAS log and the graph is not produced.

Default 0.0 (no offset)

DISPLAYSTATS=ALL | (*statistics-list*)
specifies statistics to be displayed for each box.

ALL
displays all available statistics.

See the *statistics-list*.

Note: When you choose this option, there might not be enough space for the boxes to be drawn. For horizontal box plots, you might have better results if you specify DISPLAYSTATS by itself or specify a limited set of statistics using DISPLAYSTATS=(*statistics-list*).

statistics-list
specifies a space-separated list of one or more of the following statistics, enclosed in parentheses:

DATAMAX	maximum data value that includes not only the maximum whisker values but also the maximum outlier values. The DATAMAX value is greater than or equal to the MAX value and is always represented in the axis range.
DATAMIN	minimum data value that includes not only the minimum whisker values but also the minimum outlier values. The DATAMIN value is less than or equal to the MIN value and is always represented in the axis range.
IQR	interquartile range (Q3–Q1).
MAX	maximum data value below the box upper fence.
MEAN	mean data value for the box.
MEDIAN	median data value for the box.

MIN	minimum data value above the box lower fence.
N	number of observations for the box.
Q1	lower quartile (25th percentile) for the box.
Q3	upper quartile (75th percentile) for the box.
RANGE	range of the data (MAX–MIN).
STD	standard deviation of the data for the box.
SUMWGT	sum of the weights for the box.

Example `displaystats=(mean median)`

Restriction This option is ignored if the GROUP= option is also specified.

EXTREME
specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified. When you do not specify the EXTREME option, the whiskers cannot be longer than 1.5 times the length of the box.

FILL | NOFILL
specifies whether the boxes are filled with color. The FILL option shows the fill color. The NOFILL option hides the fill color.

Default FILL

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by

their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note

This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips

Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)

specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element

specifies the name of a style element.

options

specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

COLOR=*color*

specifies a color to use for the fill lines.

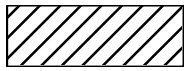
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

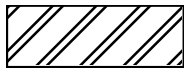
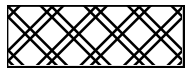
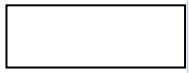
PATTERN=*line-pattern*

specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.19 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interaction If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map.

For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped boxes.

CLUSTER
the boxes are drawn adjacent to each other.

OVERLAY
all the boxes for a given group value are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Defaults CLUSTER for a discrete category axis

OVERLAY for a linear axis

Interaction This option is ignored unless GROUP=is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use

default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INTERVALBOXWIDTH=*numeric-value*
specifies the box width when an interval category (Y) variable is specified.

Alias INTBOXWIDTH=

Restriction The axis type for the category axis must be LINEAR, and the variable must be numeric.

Example
```
proc sgpanel data=sashelp.class;
  panelby sex;
  hbox weight / category=height intboxwidth=50 ;
  rowaxis type=linear;
run;
```

LABELFAR
specifies that only the far outliers have data labels. Far outliers are points whose distance from the box is more than three times the length of the box.

Note This option has no effect if you do not specify the DATALABEL option, or if there are no far outliers.

LEGENDLABEL="*text-string*"
specifies a label that identifies the box plot in the legend. By default, the label of the analysis variable is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the box outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

Interaction This option is ignored if the DATASKIN= option is also specified.

MEANATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker that represents the mean in the box. You can specify the appearance by using a style element or by specifying specific

options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphBoxMean style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

MEDIANATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the line that represents the median. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphBoxMedian style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interaction This option is ignored if the NOMEDIAN option is also specified.

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOCAPS
hides the cap lines for the whiskers.

Interaction Using several options that hide box features can cause the NOCAPS option to be ignored. For example, if you use NOCAPS, NOFILL, NOMEAN, NOMEDIAN, and NOOOUTLIERS in the same statement, the NOCAPS option might be ignored.

NOMEAN
hides the mean marker.

NOMEDIAN
hides the median line.

NOOUTLIERS
hides the outliers from the plot.

NOTCHES
specifies that the boxes be notched. See "Notches" on page 361.

OUTLIERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker that represents the outliers. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphOutlier style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

PERCENTILE=1 | 2 | 3 | 4 | 5
specifies a method for computing the percentiles for the plot.

Default 5 (empirical distribution function with averaging)

See "Calculating Percentiles" on page 362

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPREAD
relocates outlier points that have identical values to prevent overlapping.

Note This option has no effect if your data does not contain two or more outliers with identical values for the analysis variable.

TIP=(*role-list*) | NONE
specifies the information to display when the cursor is positioned over a box or whisker in the box plot.

(*role-list*)
a space-separated list of unique box plot roles enclosed in parentheses.

The box plot roles for TIP are CATEGORY, N, STD, MIN, MAX, MEAN, MEDIAN, Q1, and Q3. Specify one or more of these role names in *role-list*. Data tips are displayed using the data obtained from the specified roles.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(mean median)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

WHISKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the whisker and cap lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

options can be one or more of the following:

Default For ungrouped data, the GraphBoxWhisker style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

WHISKERPCT=*number*
specifies the whisker length, in percentile units. When this option is specified, *number* is used as the low percentile, and 100–*number* is used as the high percentile.

Here are some examples of values and their effect:

0 specifies the high and low extremes

10 specifies the 10th percentile low and the 90th percentile high

25 specifies the 25th percentile low and the 75th percentile high

Default The whiskers are drawn from the box to the most extreme point that is less than or equal to 1.5 times the IQR

Range 0–25

Notes When this option is specified, fences and far outliers are not drawn.

When this option is set to 25, no whiskers are drawn because the box extends from the 25th to the 75th percentile.

Details

Statement Summary

The plot displays a single box if only the analysis variable is provided. The plot displays multiple boxes if a category variable is also provided and that variable has more than one unique value.

The ANALYSIS variable is displayed on the horizontal axis. The axis for the analysis column is always LINEAR.

By default for numeric or character columns, the CATEGORY= axis is TYPE=DISCRETE. You can override the default and set TYPE=LINEAR in the axis statement, provided that the category column is numeric.

If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete. For the interval case, you can use the INTBOXWIDTH= option to specify the box width.

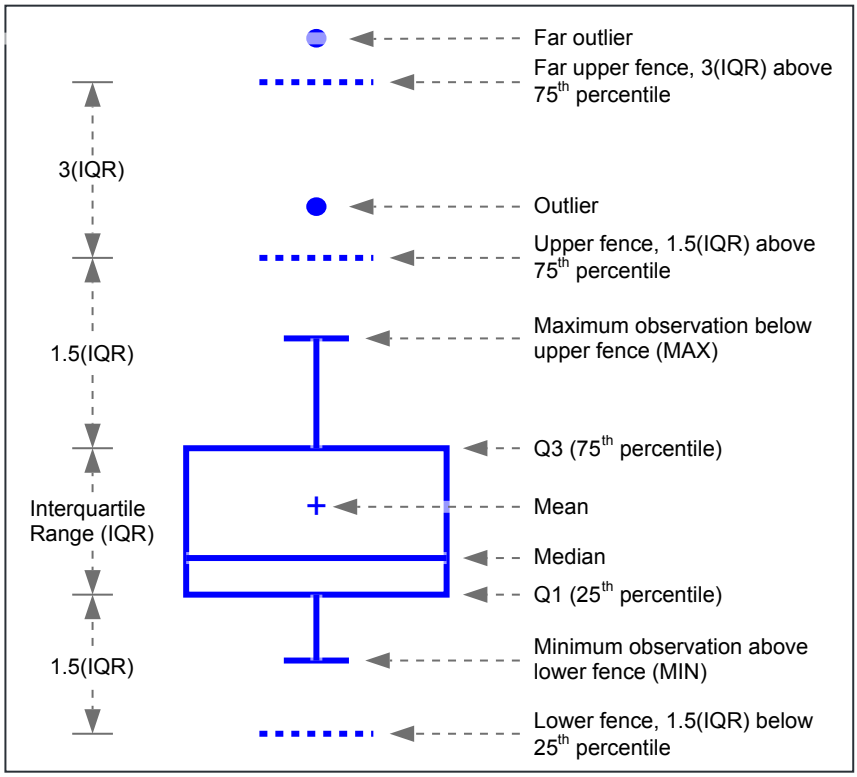
Two basic box plot representations can be drawn: a schematic (Tukey) box plot and a skeletal box plot. See the EXTREME option for details.

Visual Description of Box Plot Percentile Boundaries

Box plots display the distribution of data by using a rectangular box and whiskers. Whiskers are lines that indicate a data range outside of the box.

Note: Although the following figure shows a vertical box plot, the basic concepts apply to horizontal box plots.

Figure 8.5 *Parts of a Box Plot*



In the previous figure, the bottom and top edges of the box indicate the intra-quartile range (IQR). That is the range of values between the first and third quartiles (the 25th and 75th percentiles). The marker inside the box indicates the mean value. The line inside the box indicates the median value.

The elements that are outside the box are dependent on your options. By default, the whiskers that extend from each box indicate the range of values that are outside of the intra-quartile range. However, they are close enough not to be considered outliers (a distance less than or equal to 1.5*IQR). If you specify the EXTREME option, then the whiskers indicate the entire range of values, including outliers.

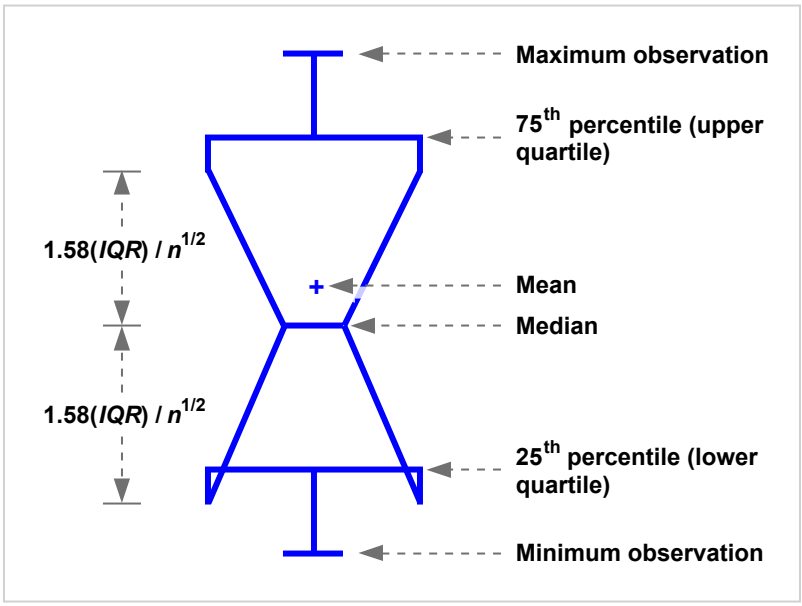
Outliers are observations that are more extreme than the upper and lower fences ($\pm\ 1.5\ IQR$). Outliers that are beyond upper and lower far fences ($\pm\ 3\ \ IQR$) are called FAR OUTLIERS. By default, outliers are indicated by markers. If you specify the DATALABEL= option, then the outlier points have data labels. If you also specify the LABELFAR option, then only outliers that are 3*IQR from the box have data labels.

Notches

In a notched box plot, the endpoints of the notches are at the following computed locations:

$$median\ \pm\ 1.58\ (IQR/\sqrt{N})$$

In the equation, IQR (IQR=Q3-Q1) is the interquartile range and *n* is the sample size. The medians (central lines) of the two boxes are significantly different at approximately the 0.05 level if the corresponding notches do not overlap.



Note: In some cases, the notches in the box plot can extend beyond DATAMIN and DATAMAX.

Calculating Percentiles

You can specify one of five definitions for computing the percentiles with the PERCENTILE= option. Let n be the number of nonmissing values for a variable, and let $X_1, X_2, ..., X_n$ represent the ordered values of the variable. X_1 is the smallest value, X_2 is the next smallest, and X_n is the largest value. Let the tth percentile be y, set:

$$p = \frac{t}{100}$$

and let:

$$np = j + g$$

when PERCENTILE=1, 2, 3, or 5, or let:

$$(n+1)p = j + g$$

when PERCENTILE=4, where j is the integer part of np, and g is the fractional part of np. Then the PERCENTILE= option defines the tth percentile, y, as described in the following table:

Percentile	Description	Equation	Notes
1	Weighted average at X_{np}	$y = (1-g)x_j + gx_{j+1}$	x_0 is taken to be x_1

Percentile	Description	Equation	Notes
2	Observation numbered closest to *np*	$y = x_j$	Used when $g < \frac{1}{2}$ or when $g = \frac{1}{2}$ and *j* is even
		$y = x_{j+1}$	Used when $g = \frac{1}{2}$ and *j* is odd or when $g > \frac{1}{2}$
3	Empirical distribution function	$y = x_j$	Used when $g = 0$
		$y = x_{j+1}$	Used when $g > 0$
4	Weighted average aimed at $x_{(n+1)p}$	$y = (1-g)x_j + gx_{j+1}$	x_{n+1} is taken to be x_n
5	Empirical distribution function with averaging	$y = \frac{1}{2}(x_j + x_{j+1})$	Used when $g = 0$
		$y = x_{j+1}$	Used when $g > 0$

HEATMAP Statement

Creates a plot of color-coded rectangles for the response variable of a pair of X and Y variables after it bins the data in two dimensions.

Interaction: The number of bins is limited to 100,000. If the number of bins exceeds the limit, the heat map is not drawn. A warning message is written to the SAS log in that case. The limit is specified by the ODS GRAPHICS statement NXYBINSMAX= option, which defaults to 100,000. See NXYBINSMAX= on page 115.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Heat Maps" on page 21

Syntax

HEATMAP X=*variable* Y=*variable* <*/options*>;

Summary of Optional Arguments

Appearance options

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-variable*
: specifies a numeric variable that is used to color the regions of the heat map.

COLORSTAT=FREQ | PCT | SUM | MEAN
: specifies the statistic to use for the COLORRESPONSE= variable.

FILLATTRS=(TRANSPARENCY=*number*)
: specifies the transparency of the area fill in the rectangles.

OUTLINE
: displays an outline around each colored region.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the rectangle outlines.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

SHOWXBINS
: specifies that bins be used as the basis for the X-axis tick marks.

SHOWYBINS
: specifies that bins be used as the basis for the Y-axis tick marks.

XENDLABELS
: specifies that axis ticks and value labels are drawn at the endpoints of the bins for the X axis.

YENDLABELS
: specifies that axis ticks and value labels are drawn at the endpoints of the bins for the Y axis.

Binning options

XBINSIZE=*positive-number*
: specifies the horizontal size of bins in X-axis coordinate system units.

XBINSTART=*number*
: specifies the data value for the first bin of the X role.

YBINSIZE=*positive-number*
: specifies the vertical size of bins in Y-axis coordinate system units.

YBINSTART=*number*
: specifies the data value for the first bin of the Y role.

Data options

DISCRETEX
: forces the X axis to be discrete when the X= variable is numeric.

DISCRETEY
: forces the Y axis to be discrete when the Y= variable is numeric.

Data tip options

TIP=(*role-list*) | NONE
: specifies the information to display when the cursor is positioned over a rectangle.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip roles that you specify in the TIP= option.

Plot options

FREQ=*numeric-variable*
: specifies a variable for the frequency count for each observation in the input data.

WEIGHT=*numeric-variable*
: specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
: specifies a name for the plot.

Required Arguments

X=*variable*

specifies the variable for the X axis.

Requirement If you specify a numeric variable and the X-axis type is discrete, then you must also specify DISCRETEX in the HEATMAP statement. Otherwise, the heat map might not be drawn correctly.

Y=*variable*

specifies the variable for the Y axis.

Requirement If you specify a numeric variable and the Y-axis type is discrete, then you must also specify DISCRETEY in the HEATMAP statement. Otherwise, the heat map might not be drawn correctly.

Optional Arguments

COLORMODEL=*style-element* | (*color-list*)

specifies a color ramp that is to be used to determine response colors.

style-element

specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is

not required when you specify a two-color ramp model.

ENDCOLOR — specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element

COLORRESPONSE=*numeric-variable*
specifies a numeric variable that is used to color the regions of the heat map.

Tip The color ramp is specified by the COLORMODEL= option. The color ramp represents the range of unique response values.

COLORSTAT=FREQ | PCT | SUM | MEAN
specifies the statistic to use for the COLORRESPONSE= variable. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

When COLORRESPONSE= is specified, the following values are valid:

SUM
sum values for the color response

MEAN
mean values for the color response

Defaults FREQ if COLORRESPONSE= is not specified

SUM if COLORRESPONSE= is specified

DISCRETEX
forces the X axis to be discrete when the X= variable is numeric.

Requirement If X= specifies a numeric variable and the X-axis type is discrete, then you must specify DISCRETEX. Otherwise, the heat map might not be drawn correctly.

Interactions If X= specifies a character variable, then this option is ignored, and the X axis is considered to be discrete.

When this option is used, the XENDLABELS= option is ignored.

DISCRETEY

forces the Y axis to be discrete when the Y= variable is numeric.

Requirement If Y= specifies a numeric variable and the Y-axis type is discrete, then you must specify DISCRETEY. Otherwise, the heat map might not be drawn correctly.

Interactions If Y= specifies a character variable, then this option is ignored, and the Y axis is considered to be discrete.

When this option is used, the YENDLABELS= option is ignored.

FILLATTRS=(TRANSPARENCY=*number*)

specifies the transparency of the area fill in the rectangles.

Default The TRANSPARENCY= option value.

Range 0 (opaque) to 1 (entirely transparent)

Interaction This option overrides the TRANSPARENCY= option for the colored region only. It does not affect the region outlines.

Tip You can combine this option with TRANSPARENCY= to set one transparency for the outline but a different transparency for the fill. Example:

```
outline transparency=0.2 fillattrs=(transparency=0.6)
```

FREQ=*numeric-variable*

specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

NAME="*text-string*"

specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

OUTLINE
displays an outline around each colored region.

Default No outline is displayed

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the rectangle outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphOutlines style element in the current style.

Interaction This option has no effect unless OUTLINE is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SHOWXBINS
specifies that bins be used as the basis for the X-axis tick marks.

Default Without this option, a standard axis is used, ignoring bin boundaries and midpoints.

Interaction When this option is used, the XENDLABELS= option determines how the axis ticks and value labels are displayed. When this option is not specified, the XENDLABELS= option is ignored.

SHOWYBINS
specifies that bins be used as the basis for the Y-axis tick marks.

Default Without this option, a standard axis is used, ignoring bin boundaries and midpoints.

Interaction When this option is used, the YENDLABELS= option determines how the axis ticks and value labels are displayed. When this option is not specified, the YENDLABELS= option is ignored.

TIP=(*role-list*) | NONE
specifies the information to display when the cursor is positioned over a rectangle.

(*role-list*)
a space-separated list of unique heat map roles enclosed in parentheses.

The roles for TIP are X, Y, and COLORRESPONSE. Specify one or more of these role names in *role-list*. Data tips are displayed using the data obtained from the specified roles.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(x y)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation’s weight is zero, negative, or missing, the observation is deleted from the analysis.

XBINSIZE=*positive-number*
specifies the horizontal size of bins in X-axis coordinate system units. The bins always span the X data range.

Default Determined by the system.

See “Binning Options” on page 1106

XBINSTART=*number*
specifies the data value for the first bin of the X role. The bins always span the X data range.

Default Determined by the system.

See “Binning Options” on page 1106

XENDLABELS
specifies that axis ticks and value labels are drawn at the endpoints of the bins for the X axis.

Default If this option is not used, then the axis ticks and value labels are drawn at the bin midpoints.

Interactions This option has no effect unless the SHOWXBINS option is also used.

This option is ignored when the X axis is discrete.

YBINSIZE=*positive-number*
specifies the vertical size of bins in Y-axis coordinate system units. The bins always span the Y data range.

Default Determined by the system.

See "Binning Options" on page 1106

YBINSTART=*number*
specifies the data value for the first bin of the Y role. The bins always span the Y data range.

Default Determined by the system.

See "Binning Options" on page 1106

YENDLABELS
specifies that axis ticks and value labels are drawn at the endpoints of the bins for the Y axis.

Default If this option is not used, then the axis ticks and value labels are drawn at the bin midpoints.

Interactions This option has no effect unless the SHOWYBINS option is also used.

This option is ignored when the Y axis is discrete.

HEATMAPPARM Statement

Creates a plot that represents the values of three variables. Generating an X, Y grid of rectangles from the values of two independent variables, it colors the rectangles to represent the values of a third variable, which can be a response variable or a group variable.

Requirements: The COLORGROUP= role or the COLORRESPONSE= role must be specified.

The data must have at least one observation that has a value for both the X and Y variables. Otherwise, the heat map is not drawn.

Interaction: The number of bins is limited to 100,000. If the number of bins exceeds the limit, the heat map is not drawn. A warning message is written to the SAS log in that case. The limit is specified by the ODS GRAPHICS statement NXYBINSMAX= option, which defaults to 100,000. See NXYBINSMAX= on page 115.

Note: The data for a parameterized heat map should contain only one observation for each X and Y value pair.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Heat Maps" on page 21

Syntax

HEATMAPPARM X=*variable* Y=*variable*
COLORGROUP=*variable* </*options*>;

HEATMAPPARM X=*variable* Y=*variable*
COLORRESPONSE=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

FILLATTRS=(TRANSPARENCY=*number*)
: specifies the transparency of the area fill in the rectangles.

NOMISSINGCOLOR
: excludes missing values of the COLORGROUP= variable or of the COLORRESPONSE= variable from the heat map.

OUTLINE
: displays an outline around each colored region.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the rectangle outlines.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
: specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins.

YVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
: specifies whether the input Y values represent midpoints, lower endpoints, or upper endpoints of the bins.

Axis options

SHOWXBINS
: specifies that bins be used as the basis for the X-axis tick marks.

SHOWYBINS

specifies that bins be used as the basis for the Y-axis tick marks.

XENDLABELS
specifies that axis ticks and value labels are drawn at the endpoints of the bins for the X axis.

YENDLABELS
specifies that axis ticks and value labels are drawn at the endpoints of the bins for the Y axis.

Data options

DISCRETEX
forces the X axis to be discrete when the X= variable is numeric.

DISCRETEY
forces the Y axis to be discrete when the Y= variable is numeric.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Plot options

MISSING
accepts a missing value as a valid category value.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Requirement If you specify a numeric variable and the X-axis type is discrete, then you must also specify DISCRETEX in the HEATMAPPARM statement. Otherwise, the heat map might not be drawn correctly.

Y=*variable*
specifies the variable for the Y axis.

Requirement If you specify a numeric variable and the Y-axis type is discrete, then you must also specify DISCRETEY in the HEATMAPPARM statement. Otherwise, the heat map might not be drawn correctly.

COLORGROUP=*variable*
specifies a variable that is used to color the regions of the heat map.

Requirement This argument is required when the response variable is of type discrete.

COLORRESPONSE=*numeric-variable*
specifies a numeric variable that is used to color the regions of the heat map.

Requirement This argument is required when the response variable is of type interval.

Tip The color ramp is specified by the COLORMODEL= option. The color ramp represents the range of unique response values.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

Note: This feature applies to the COLORGROUP variable in the heat map.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example
```
colormodel=TwoColorRamp
```

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorRamp style element

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

DISCRETEX
forces the X axis to be discrete when the X= variable is numeric.

Requirement If X= specifies a numeric variable and the X-axis type is discrete, then you must specify DISCRETEX. Otherwise, the heat map might not be drawn correctly.

Interactions If X= specifies a character variable, then this option is ignored, and the X axis is considered to be discrete.

When this option is used, the XENDLABELS= option is ignored.

DISCRETEY
forces the Y axis to be discrete when the Y= variable is numeric.

Requirement If Y= specifies a numeric variable and the Y-axis type is discrete, then you must specify DISCRETEY. Otherwise, the heat map might not be drawn correctly.

Interactions If Y= specifies a character variable, then this option is ignored, and the Y axis is considered to be discrete.

When this option is used, the YENDLABELS= option is ignored.

FILLATTRS=(TRANSPARENCY=*number*)
specifies the transparency of the area fill in the rectangles.

Default The TRANSPARENCY= option value.

Range 0 (opaque) to 1 (entirely transparent)

Interaction This option overrides the TRANSPARENCY= option for the colored region only. It does not affect the region outlines.

Note The fill colors are determined by the COLORRESPONSE= or COLORGROUP= variable.

Tip You can combine this option with TRANSPARENCY= to set one transparency for the outline but a different transparency for the fill. Example:

```
outline transparency=0.2 fillattrs=(transparency=0.6)
```

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Requirement The option enables missing values to show up along the X or Y axis provided that the axis is discrete. Either the values must be character, or you can specify the DISCRETEX or DISCRETEY option.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGCOLOR
excludes missing values of the COLORGROUP= variable or of the COLORRESPONSE= variable from the heat map. If missing color values are present, observations with missing COLORGROUP= or COLORRESPONSE= values are not displayed in the heat map. However, their values are still represented on the axis.

Default Without this option, missing values are included in the heat map. The attributes of the missing value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphData*n* style element instead of by the GraphMissing style element.

OUTLINE
displays an outline around each colored region.

Default No outline is displayed

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the rectangle outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphOutlines style element in the current style.

Interaction This option has no effect unless OUTLINE is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SHOWXBINS
specifies that bins be used as the basis for the X-axis tick marks.

Default Without this option, a standard axis is used, ignoring bin boundaries and midpoints.

Interaction When this option is used, the XENDLABELS= option determines how the axis ticks and value labels are displayed. When this option is not specified, the XENDLABELS= option is ignored.

SHOWYBINS
specifies that bins be used as the basis for the Y-axis tick marks.

Default Without this option, a standard axis is used, ignoring bin boundaries and midpoints.

Interaction When this option is used, the YENDLABELS= option determines how the axis ticks and value labels are displayed. When this option is not specified, the YENDLABELS= option is ignored.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

XENDLABELS
specifies that axis ticks and value labels are drawn at the endpoints of the bins for the X axis.

Default If this option is not used, then the axis ticks and value labels are drawn at the bin midpoints. This is true regardless of whether the XVALUES= option identifies the X data as endpoint values or midpoint values.

Interactions This option has no effect unless the SHOWXBINS option is also used.

This option is ignored when the X axis is discrete.

XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins.

Default MIDPOINTS

YENDLABELS
specifies that axis ticks and value labels are drawn at the endpoints of the bins for the Y axis.

Default If this option is not used, then the axis ticks and value labels are drawn at the bin midpoints. This is true regardless of whether the YVALUES= option identifies the Y data as endpoint values or midpoint values.

Interactions This option has no effect unless the SHOWYBINS option is also used.

This option is ignored when the Y axis is discrete.

YVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the input Y values represent midpoints, lower endpoints, or upper endpoints of the bins.

Default MIDPOINTS

HIGHLOW Statement

Creates a display of floating vertical or horizontal lines or bars that represent high and low values. The statement also gives you the option to display open and close values as tick marks and to specify a variety of plot attributes.

Note: This plot does not summarize data. If multiple observations have the same X or Y value, the observations are all plotted separately based on their values.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About High-Low Plots" on page 23

Syntax

HIGHLOW X=*variable* | Y=*variable*
HIGH=*numeric-variable* LOW=*numeric-variable* </option(s)>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

CLIPCAP
: displays a clip indicator cap at the end of a bar or line when the bar or line extends beyond the axis range.

CLIPCAPSHAPE=DEFAULT | SERIF | BARBEDARROW | CLOSEDARROW | OPENARROW
: specifies the type of cap used for the clip cap.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all lines or bars from the category midpoints.

FILL

NOFILL
: specifies whether the area fill is visible for bars.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

HIGHCAP=*character-variable* | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW
: specifies the type of cap used at the high end of the bar or line.

INTERVALBARWIDTH=*numeric-value*<*units*>
: specifies the thickness of the bar when the X (or Y) data is numeric.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the outlines for the band.

LOWCAP=*character-variable* | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW
: specifies the type of cap used at the low end of the bar or line.

OUTLINE

NOOUTLINE
: specifies whether the outlines of the bars are visible.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE

specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=OVERLAY | CLUSTER
specifies how to display grouped data.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

HIGHLABEL=*variable*
specifies the label to be shown at the high end of the line or bar.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the HIGHLABEL= option, the LOWLABEL= option, or both options.

LEGENDLABEL="*text-string*"
specifies a label that identifies the elements from the band plot in the legend.

LOWLABEL=*variable*
specifies the label to be shown at the low end of the line or bar.

Plot options

CLOSE=*numeric-variable*
specifies the data for the CLOSE tick on the bar or line.

OPEN=*numeric-variable*
specifies the data for the OPEN tick on the bar or line.

TYPE=BAR | LINE
specifies how the data is to be represented.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*variable*
Y=*variable*
specifies a variable that is used to plot the values along the N or Y axis.

Note If you specify X=*variable*, then the statement creates vertical lines or bars on the X axis, and the HIGH and LOW values are plotted along the Y axis. Conversely, if you specify Y=*variable*, then the statement creates horizontal lines or bars on the Y axis, and the HIGH and LOW values are plotted along the X axis.

HIGH=*numeric-variable*
specifies the upper value for the floating lines or bars.

LOW=*numeric-variable*
specifies the lower value for the floating lines or bars.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of `1`, then there is no distance between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

Default 0.85

Restriction This option applies only when the X or Y axis is discrete.

Interaction This option has no effect unless TYPE= BAR.

Tip If the category axis is numeric, use the INTERVALBARWIDTH= option to specify the bar width.

CLIPCAP
displays a clip indicator cap at the end of a bar or line when the bar or line extends beyond the axis range. The cap indicates where clipping has occurred. When the MIN= and MAX= axis options are specified for an axis and a data value exceeds the specified axis range, the bar or line for that value is clipped. If the bar or line already has a high or low cap, it is replaced by the clip cap.

For vertical bars, the clip cap is added to the end of the bar that is clipped by the Y-axis range. The cap is a vertical arrowhead that points toward the clip edge (or).

For horizontal bars, the clip cap is added to the end of the bar that is clipped by the X-axis range. The cap is a horizontal arrowhead that points toward the clip edge (or).

If an entire bar or line is clipped, a clip cap is displayed at the high or low side where the bar or line was clipped.

Interaction Clip indicators appear only when CLIPCAP is specified and the data values exceed the axis range that is specified by the MIN= and MAX= options. When the MIN= and MAX= options are not specified, the axis range is adjusted to accommodate the data values, and clipping does not occur.

Note When the high-low TYPE=LINE, you can change the appearance of the clip cap using the CLIPCAPSHAPE= option.

Tip If you specify the HIGHLABEL or LOWLABEL option, and the bar or line is clipped, the label value is still drawn outside the tip of the clip cap. If the entire bar or line is clipped, no labels are shown.

CLIPCAPSHAPE=DEFAULT | SERIF | BARBEDARROW | CLOSEDARROW | OPENARROW

specifies the type of cap used for the clip cap.

The following table shows each clip-cap shape.

Table 8.20 *Clip-Cap Shapes*

Clip-Cap Shape	TYPE=LINE	TYPE=BAR
DEFAULT		
SERIF		Not applicable.[1]
BARBEDARROW		Not applicable.[1]
CLOSEDARROW		
OPENARROW		Not applicable.[1]

1 When specified for bars, DEFAULT is used instead.

Default DEFAULT

CLOSE=*numeric-variable*

specifies the data for the CLOSE tick on the bar or line. For a vertical plot, the tick value is represented by an indicator on the side of the bar or line that has

higher X values. For a horizontal plot, the value is represented by an indicator on the side with higher Y values.

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Requirement This option is applicable only when the X or Y axis is discrete.

Interactions This option is applicable only when TYPE=BAR.

This option is applicable only when the GROUP option is specified and GROUPDISPLAY=CLUSTER.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for lines or for outline-only bars.

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 8.21 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

High-low plots can specify lines rather than bars.

Table 8.22 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit

is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines or bars from the category midpoints. Specify a value from –0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset from the category midpoints)

Requirement This option is applicable only when the X or Y axis is discrete.

FILL | NOFILL
specifies whether the area fill is visible for bars. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default The default status of the area fill is specified by the DisplayOpts attribute of the GraphBand style element in the current style.

Interactions This option has no effect unless TYPE=BAR.

Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataConfidence style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

0.0 transparency

Interaction This option has no effect if you specify the NOFILL option.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interactions

If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

TYPE=BAR must be specified for the fill pattern to have any effect.

Accessibility note

This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips

Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)

specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element

specifies the name of a style element.

options

specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

COLOR=*color*

specifies a color to use for the fill lines.

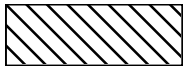
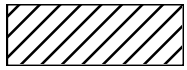
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

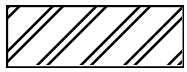
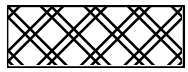
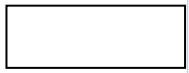
PATTERN=*line-pattern*

specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.23 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions

For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note

This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

GROUP=*variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction

When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, “Using Attribute Maps to Control Visual Attributes,” on page 1897.

Tip

ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=OVERLAY | CLUSTER
specifies how to display grouped data.

OVERLAY
groups are overlaid without any clustering.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

CLUSTER
observations with different group values are displayed in adjacent clusters around the category value.

Each set of group values is centered at the midpoint tick mark for the category.

Default OVERLAY

Interactions GROUPDISPLAY=CLUSTER is applicable only when TYPE=BAR.

This option is ignored unless GROUP= is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

HIGHCAP=*character-variable* | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW

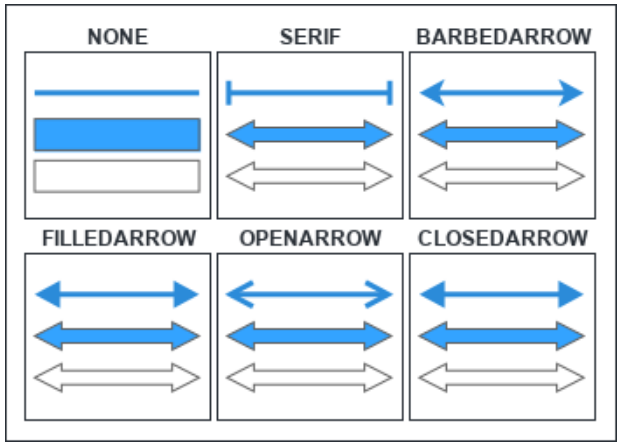
specifies the type of cap used at the high end of the bar or line. You can specify one of the keywords, or you can specify a character variable that contains one of the keywords.

All of the keywords can be specified for any high-low chart. However, the effect of each keyword depends on the setting for the TYPE= option and also the fill state of the bars, when displayed:

- When TYPE=BAR and the bars are filled, FILLEDARROW is used for all settings other than NONE.
- When TYPE=BAR and the bars are not filled, CLOSEDARROW is used for all settings other than NONE.
- When TYPE=LINE and CLOSEDARROW is specified, FILLEDARROW is used instead.

The following figure shows the effect of each cap value on horizontal lines, filled bars, and unfilled bars.

Figure 8.6 *Horizontal High and Low Cap Shapes for Lines, Filled Bars, and Unfilled Bars*



Default NONE

Restriction Caps are not displayed for very short bars. Bar height must be at least twice the size of the cap in order for the cap to appear.

Interaction When TYPE=BAR, the caps are drawn to fit within the bar width. The width of the bar itself might be reduced.

HIGHLABEL=*variable*

specifies the label to be shown at the high end of the line or bar.

INTERVALBARWIDTH=*numeric-value*<*units*>
specifies the thickness of the bar when the X (or Y) data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along X or Y.

Requirement This option is applicable only when the X or Y axis is a linear axis or a TIME axis.

Interaction This option has no effect unless TYPE=BAR.

Tip If the category axis is discrete, use the BARWIDTH= option to specify the bar width.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the HIGHLABEL= option, the LOWLABEL= option, or both options. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LEGENDLABEL="*text-string*"
specifies a label that identifies the elements from the band plot in the legend. By default, the label "band" is used for ungrouped data, and the group values are used for grouped data.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the outlines for the band. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the GraphOutlines style element in the current style.

For grouped data:

- if only outlines are displayed, LineThickness from GraphOutlines, and ContrastColor and LineStyle from a GraphData1–GraphData*n* style element in the current style
- if outlines and fill are displayed, LineThickness and LineStyle from GraphOutlines, and ContrastColor from a GraphData1–GraphData*n* style element in the current style

Interactions This option has no effect unless you also specify the OUTLINES option.

This option is ignored if the DATASKIN= option is also specified.

LOWCAP=*character-variable* | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW

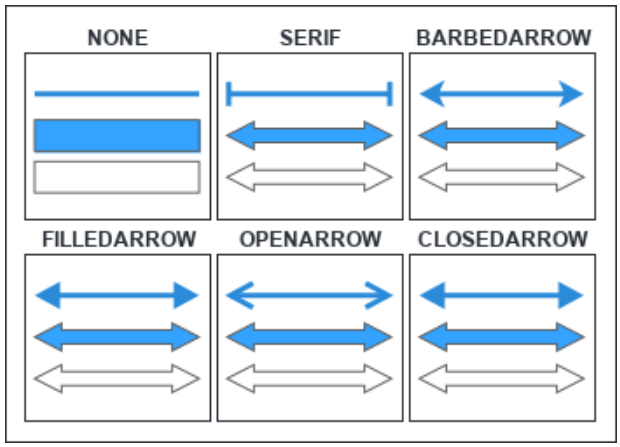
specifies the type of cap used at the low end of the bar or line. You can specify one of the keywords, or you can specify a character variable that contains one of the keywords.

All of the keywords can be specified for any high-low chart. However, the effect of each keyword depends on the setting for the TYPE= option and also the fill state of the bars, when displayed:

- When TYPE=BAR and the bars are filled, FILLEDARROW is used for all settings other than NONE.
- When TYPE=BAR and the bars are not filled, CLOSEDARROW is used for all settings other than NONE.
- When TYPE=LINE and CLOSEDARROW is specified, FILLEDARROW is used instead.

The following figure shows the effect of each cap value on horizontal lines, filled bars, and unfilled bars.

Figure 8.7 *Horizontal High and Low Cap Shapes for Lines, Filled Bars, and Unfilled Bars*



Default NONE

Restriction Caps are not displayed for very short bars. Bar height must be at least twice the size of the cap in order for the cap to appear.

Interaction When TYPE=BAR, the caps are drawn to fit within the bar width. The width of the bar itself might be reduced.

LOWLABEL=*variable*
specifies the label to be shown at the low end of the line or bar.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OPEN=*numeric-variable*
specifies the data for the OPEN tick on the bar or line.

For a vertical plot, the tick value is represented by an indicator on the side of the bar or line that has lower X values. For a horizontal plot, the value is represented by an indicator on the side with lower Y values.

OUTLINE | NOOUTLINE
specifies whether the outlines of the bars are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions This option has no effect unless TYPE=BAR.

Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

Tips For ungrouped data, the outline appearance is controlled by the GraphOutlines style element. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data, if only outlines are displayed, the outline thickness is controlled by the GraphOutlines style element, and the color and line style are controlled by a GraphData1–GraphData*n* style element. If outlines and fill are displayed, the outline thickness and line style are controlled by the GraphOutlines style element, and the color is controlled by a GraphData1–GraphData*n* style element.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, “Using Range Attribute Maps,” on page 1921

“Overview of Attribute Maps” on page 1897

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

TYPE=BAR | LINE
specifies how the data is to be represented. BAR uses fill and outline attributes. LINE uses line attributes.

Default LINE

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

HISTOGRAM Statement

Creates a histogram that displays the frequency distribution of a numeric value.

Interactions: The HISTOGRAM statement can be combined only with DENSITY statements in the SGPANEL procedure. See "Plot Compatibility" on page 1780.

The number of bins is limited to 10,000. If the number of bins exceeds the limit, the histogram is not drawn. A warning message is written to the SAS log in that case. The limit is specified by the ODS GRAPHICS statement NBINSMAX= option, which defaults to 10,000. See NBINSMAX= on page 114.

Note: The range of the response variable is automatically divided into an appropriate number of bins.

Examples: "About Histograms" on page 51

"Example 1: Creating a Panel of Graph Cells with Histograms and Density Plots" on page 832

Syntax

HISTOGRAM *response-variable* </option(s)>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

FILL

NOFILL
: specifies whether the area fill is visible.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLENDCOLOR=*color*
: specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
: specifies the fill type that is applied to the chart.

OUTLINE

NOOUTLINE
: specifies whether outlines are displayed for the bars.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Group options

GROUP=*variable*
: specifies a variable that is used to group the data.

Label options

DATALABEL

DATALABEL=NONE | AUTO | COUNT | DENSITY | PERCENT | PROPORTION
: specifies the statistic to display at the end of each bin.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)

specifies the appearance of the labels in the plot when you use the DATALABEL= option.

LEGENDLABEL="*text-string*"
specifies a label that identifies the histogram in the legend.

Plot options

BINSTART=*numeric-value*
specifies the X coordinate of the first bin.

BINWIDTH=*numeric-value*
specifies the bin width.

BOUNDARY=LOWER | UPPER
specifies how boundary values are assigned to bins.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

NBINS=*numeric-value*
specifies the number of bins.

SCALE=COUNT | PERCENT | PROPORTION
specifies the scaling that is applied to the vertical axis.

WEIGHT=*numeric-variable*
specifies a variable in the input data set that contains values to be used as weights for the bin height.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Argument

response-variable
specifies a response variable for the histogram.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

BINSTART=*numeric-value*
specifies the X coordinate of the first bin. Use this option in conjunction with the BINWIDTH= or NBINS= options to specify bins. If neither BINWIDTH= nor the NBINS= option is specified, the system determines the number of bins. If the BINSTART value results in excluding the entire range of data, it is ignored and the default BINSTART value is used.

Default The default value is determined by the system.

BINWIDTH=*numeric-value*
specifies the bin width. The system determines the number of bins. The bins always span the range of the data.

Default The default value is determined by the system.

Restriction The maximum number of bins is limited to approximately 10,000. If the number of bins computed from the data and the BINWIDTH= value exceeds 10,000, SAS computes a new bin-width value that yields approximately 10,000 bins. A warning of the change is written to the SAS log.

Interaction This option is ignored if the NBINS= option is also specified.

BOUNDARY=LOWER | UPPER
specifies how boundary values are assigned to bins.

LOWER
specifies that boundary values are assigned to the lower bin.

UPPER
specifies that boundary values are assigned to the upper bin.

Default UPPER

DATALABEL
DATALABEL=NONE | AUTO | COUNT | DENSITY | PERCENT | PROPORTION
specifies the statistic to display at the end of each bin. You can specify one of the following:

NONE
suppresses the data labels.

AUTO
uses the SCALE= option value. By default, SCALE=PERCENT.

COUNT
DENSITY
PERCENT
PROPORTION
specifies that the count, density, percentage, or proportion statistic is to be displayed at the end of each bin.

Default If DATALABEL is specified without a value, then the default is AUTO.

Interaction When DATALABEL=AUTO, the SCALE= option determines the statistic that is displayed at the end of each bin.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bins. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 8.24 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data"

on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

FILL | NOFILL
specifies whether the area fill is visible. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default The default status of the area fill is specified by the DisplayOpts style attribute of the GraphHistogram style element in the current style.

Interactions Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction
: If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note
: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips
: Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

: Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

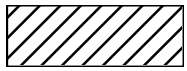
COLOR=*color*
specifies a color to use for the fill lines.

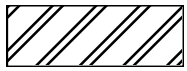
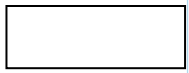
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.25 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bin is filled with the color that is assigned to the bin fill area.

GRADIENT
displays the color from the applied style at the end of the bin gradually changing to the color of the wall at the baseline of the bin.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bin is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bin to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bins.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

LEGENDLABEL="*text-string*"
specifies a label that identifies the histogram in the legend. By default, the label of the response variable is used.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NBINS=*numeric-value*
specifies the number of bins. The system determines the BINWIDTH= value. The bins always span the range of the data.

The procedure attempts to produce tick values that are easily interpreted (for example, 5, 10, 15, 20). The procedure sometimes adjusts the location of the first bin and the bin width accordingly. As a result, the number of bins shown in the plot might not exactly match the number specified with the NBINS= option.

Default The default number of bins is determined by the system.

Range 2–10,000

Note The limit on the number of bins is specified by the ODS GRAPHICS statement NBINSMAX= option, which defaults to 10,000. See "NBINSMAX=*n*" on page 114.

OUTLINE | NOOUTLINE
specifies whether outlines are displayed for the bars. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default The default status of the bar outlines is specified by the DisplayOpts attribute of the GraphHistogram style element in the current style.

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

SCALE=COUNT | PERCENT | PROPORTION
specifies the scaling that is applied to the vertical axis. Specify one of the following values:

COUNT
the axis displays the frequency count.

PERCENT
the axis displays values as a percentage of the total.

PROPORTION
the axis displays values as proportions (0.0 to 1.0) of the total.

Default PERCENT

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies a variable in the input data set that contains values to be used as weights for the bin height.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

HLINE Statement

Creates a horizontal line chart (the line is vertical). You can use the HLINE statement with the HBAR statement to create a horizontal bar-line chart.

Restrictions: The HLINE statement can be combined only with other categorization plot statements in the SGPANEL procedure. See "Plot Compatibility" on page 1780.

When category plots are overlaid, all of the plots must use the same category variable.

See: "Marker Fills and Outlines" on page 1805

Example: "About Line Charts" on page 61

Syntax

HLINE *category-variable* </option(s)>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

BREAK
breaks the line whenever the computed statistic for a category value is missing.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from the discrete category values.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the lines in the line plot.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
specifies a category variable to divide the values into groups.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot.

LEGENDLABEL="*text-string*"
specifies the label that identifies the line plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

STATLABEL

NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels.

Limit options

LIMITATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITS=BOTH | LOWER | UPPER
adds limit lines to the plot.

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines.

NOLIMITCAPS
suppresses the serif cap on limit lines, if limit lines are displayed.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

MARKERS
adds data point markers to the plot.

Plot options

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

MISSING
accepts a missing value as a valid category value.

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

WEIGHT=*numeric-variable*
specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Statistics options

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the horizontal axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the lines.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default .05

Interactions This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BREAK
breaks the line whenever the computed statistic for a category value is missing. For example, if a response variable is used and it has all missing values for a certain category value, the SUM or MEAN for this category value will be missing. By default in such cases, the response value for the previous category is joined to the response value for the next category value by a line segment. If BREAK is specified, this segment is not drawn.

Interaction This option is ignored when the Y axis is discrete and the MISSING option is specified. To break the line at missing values in that case, remove the MISSING option.

Note The observation is excluded from the graph when there is a missing value for the FREQ variable.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged. By default, the plot is sorted in ascending order based on the category values. You can change the order so that it is based on the response values.

Specify one of the following values:

RESPASC
sorts by the response values in ascending order.

RESPDESC
sorts by the response values in descending order.

Default The plot is sorted in ascending order based on the category values.

Restriction This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Requirement This option requires that you configure the panel to use either one column or one row, depending on the orientation of your charts. Use the ROWS= option or the COLUMNS= option in the PANELBY statement. If you do not use this option and your graph contains multiple cells, the specified sort order is not correctly applied to all cells.

Interactions When a group variable is used with the CATEGORYORDER= option, the category order is not affected by the value of the groups. The categories are always sorted by the response statistic at a category level.

When this option and the GROUPORDER= option are both specified, the GROUPORDER= option is ignored.

Notes If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

When two or more observations have the same response values, there is no guarantee that the tied values will be sub-sorted.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interaction This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:

```
CURVELABELATTRS=GraphTitleText
```

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label. Specify one of the following values:

MIN
places the label at the part of the curve closest to the minimum X-axis value.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

START
places the curve label at the first point on the curve.

END
places the curve label at the last point on the curve.

Default END

Interaction This option has no effect unless the CURVELABEL option is also specified.

DATALABEL
DATALABEL=*variable*
displays a label for each data point.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITS or LIMITSTAT, then the limit labels are also shown. Limit labels are shown when you use DATALABEL (without a variable).

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, for the label font, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style. For the label color, when DATALABELPOS=DATA, the ContrastColor attribute of a GraphData1–GraphData*n* style element. Otherwise, the Color attribute of the GraphDataText style element.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 8.26 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

If you specify markers with the plot, then the data skin affects the markers as well.

Table 8.27 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	**PRESSED**	**SHEEN**

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from the discrete category values. Specify a value from –0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the category axis is discrete.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interactions This option has no effect unless MARKERS is also specified.

Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interactions If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

When FREQ= and MISSING are both specified, the MISSING option has no effect.

GROUP=*variable*
specifies a category variable to divide the values into groups. A separate plot is created for each unique value of the category variable.

Interaction If you specify more than one categorization plot statement, then all of the plots must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the categorization plots, then the GROUP= option has no effect.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interaction This option is ignored unless GROUP= is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default ASCENDING

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the line plot in the legend. By default, the label of the response variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) are used. If you do not specify a response variable, then the legend label is "Frequency".

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

For grouped data, the LineStyle and LineThickness attributes of the GraphError style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITS= or LIMITSTAT= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITS=BOTH | LOWER | UPPER
adds limit lines to the plot. Specify one of the following values:

BOTH
adds lower and upper limit lines to the plot.

LOWER
adds lower limit lines to the plot.

UPPER
adds upper limit lines to the plot.

Default By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interaction Limit lines are displayed only when you specify STAT= MEAN.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines. Specify one of the following

CLM
confidence limits

STDDEV
standard deviation

STDERR
standard error

Default If you specify the LIMITS= option, then the default for LIMITSTAT= is CLM. Otherwise, there is no default.

Interaction If you specify the LIMITSTAT= option, then the default value for the LIMITS= option is BOTH.

Tip To display limit labels, specify DATALABEL (without a variable).

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the lines in the line plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see “Line Attributes and Patterns” on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interactions This option has no effect unless you also specify the MARKERS option.

Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

Suboption COLOR= in option LINEATTRS= overrides the default marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style

element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKERS
adds data point markers to the plot.

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOLIMITCAPS
suppresses the serif cap on limit lines, if limit lines are displayed.

Interaction The appearance of the limit lines is controlled by the LIMITATTRS= option.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR. You can specify any positive number, including decimals.

Default 1

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the horizontal axis.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the horizontal axis. Specify one of the following:

FREQ
frequencies of the response or category variable.

The frequencies are calculated as follows:

- If you specify the RESPONSE= option, FREQ calculates the frequency of the response variable.
- If you do not specify the RESPONSE= option, FREQ calculates the frequency of the category variable.

MEAN
the mean of the response variable.

Interaction For STAT=MEAN to take effect, you must also specify the RESPONSE= option.

MEDIAN
the median of the response variable.

Interaction For STAT=MEDIAN to take effect, you must also specify the RESPONSE= option.

PERCENT
the percentage of the response or category variable.

The percentage is calculated as follows:

- If you specify the RESPONSE= option, PERCENT calculates the percentage of the sum of the response variable.
- If you do not specify the RESPONSE= option, PERCENT calculates the percentage of the frequency of the category variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the PCTLEVEL= option in the PROC SGPANEL statement.

You can use the PCTNDEC= option in the SGPANEL procedure statement to control the number of decimals to be used when calculating the percent values.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level will be zero.

SUM
the sum of the response variable.

This is the default value when you specify the RESPONSE= option.

Interaction For this value to take effect, you must also specify the RESPONSE= option.

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Restriction If you do not also specify the RESPONSE= option, then only the FREQ or PERCENT statistic is calculated (FREQ is the default). If you specify RESPONSE=, then you can use any of the statistics.

Interaction When the graph is generated, the statistic is appended to the variable name in the axis label and the legend (if it is created). However, if a label has been assigned to the variable, then the label appears in the axis label and legend instead of the statistic.

STATLABEL | NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels. STATLABEL forces the statistic to be displayed. NOSTATLABEL removes the statistic from the axis and legend labels.

Normally, the procedure displays the statistic along with the name of the response variable. However, when a custom label is assigned to the response variable, the procedure does not display the statistic. In each case, you can control whether the statistic is displayed.

Defaults The statistic is displayed for the response variable.

When a custom label is assigned to the response variable, the statistic is not displayed.

Interactions This option has no effect unless the RESPONSE= option is specified.

This option has no effect if you specify the axis label using the LABEL= option in an AXIS statement.

TIP=(*variable-list*) | NONE

specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)

a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE

suppresses the data tips from this plot.

Requirement: You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction: This option replaces all of the information that is displayed by default.

Notes: When TIP= is not specified, the category, RESPONSE, and GROUP variables are displayed, if applicable, when you hover over a line or marker. The category and upper limit or lower limit variables are displayed when you hover over an error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a line or marker. No variables are displayed when you hover over an error bar or error bar cap.

Tip: Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example: tip=(age weight)

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default: The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement: A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction: This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

If your categorical chart is overlaid with other categorical charts, each chart can specify its own URL variable. The URL variable specified for each chart overlay is honored. This capability affects the following charts: DOT, HBAR, HLINE, VBAR, VLINE.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction If your plot is overlaid with other categorization plots that also specify WEIGHT=, then the first WEIGHT variable that you specified is used for all of the plots.

INSET Statement

Adds a text box to each panel cell of the graph.

Restriction: Only one INSET statement can be specified in the PROC SGPANEL step. If more than one is specified, the first inset is generated and a message is logged for the additional statements.

See: "About Text Insets" on page 38

Syntax

INSET *variable* <*...variable-n*> </*options*>;

Summary of Optional Arguments

Appearance options
BACKCOLOR=*color*

specifies the background color of the inset.

BORDER

NOBORDER

specifies whether to display a border around the text box.

NOLABEL

suppresses the label.

OPAQUE

forces the inset background to be opaque rather than transparent.

POSITION=*position-value*

specifies the position of the text box within the plot.

SEPARATOR="*string*"

specifies one or more characters to place between the data label and the data value.

TEXTATTRS=*style-element*<(***options***)> | (*options*)

specifies the appearance of the text in the text box.

TITLE="*text-string*"

specifies a title for the text box.

TITLEATTRS=*style-element*<(***options***)> | (*options*)

specifies the appearance of the title.

Required Argument

variable <...variable-n>

specifies one or more variables to use for the data-driven text inside the inset. Typically, the variable is a computed numeric value, such as a mean or a sum. For non-computed variables, the statement displays the value of the first observation for each classification.

The inset labels are derived from the variable labels, or variable names if the labels are not present. (You can suppress the labels using the NOLABEL option.) The inset values come from the variable data.

Tip If the text does not fit in the space provided, the inset might become clipped. Use the TEXTATTRS= option to change the font size or other font characteristics.

Optional Arguments

BACKCOLOR=*color*

specifies the background color of the inset. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default By default, the background is transparent.

Interaction If BACKCOLOR= is set, then the OPAQUE option is also set automatically.

BORDER | NOBORDER

specifies whether to display a border around the text box. The BORDER option displays the border. The NOBORDER option hides the border.

NOLABEL
suppresses the label.

OPAQUE
forces the inset background to be opaque rather than transparent. This option is useful when a transparent background makes the text in the inset difficult to see. For example, if the inset is positioned on top of a histogram, specifying OPAQUE sets the inset off from the histogram.

Default The background is transparent unless BACKCOLOR= is also specified.

Interaction If BACKCOLOR= is specified, the background is always opaque.

POSITION=*position-value*
specifies the position of the text box within the plot. The position values are as follows:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
TOP	TOPLEFT	TOPRIGHT
LEFT	RIGHT	

If you do not specify a position, then a position is determined automatically.

SEPARATOR="*string*"
specifies one or more characters to place between the data label and the data value. With fonts that support Unicode, you can produce specific characters by specifying a hexadecimal value. A trailing x identifies a string as a hexadecimal value. You must also enclose the character specification in a special ODS handler string, in the format (*ESC*){Unicode '*hexadecimal-value*'x}.

For example:

```
separator="(*ESC*){unicode '03B1'x}";
```

This option produces the lowercase Greek letter alpha for the separator.

Interaction This option is ignored if NOLABEL is also specified.

TEXTATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the text in the text box. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Tip The GraphDataText style element makes the text slightly smaller. The GraphLabelText style element makes the text slightly larger.

Examples
```
TEXTATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TEXTATTRS=GraphDataText
```

TITLE="*text-string*"
specifies a title for the text box. The title text is always center-aligned.

TITLEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the title. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

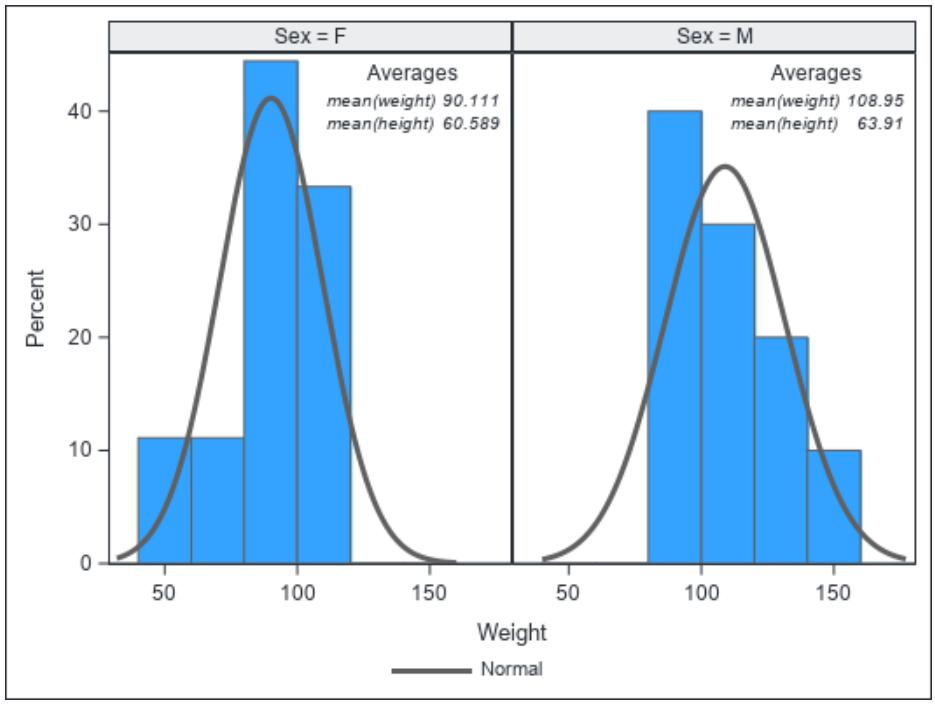
For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

Examples
```
TITLEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TITLEATTRS=GraphTitleText
```

Example: Panel with Insets




```
/* Calculate the mean height and weight for the inset */
proc means data=sashelp.class nway;
  class sex;
  var weight height;
```

```
  output out=stats mean(weight)=mean_weight mean(height)=mean_height;
run;

/* Sort the data */
proc sort data=sashelp.class out=class;
  by sex;
run;

/* Match-merge the inset data with the original data */
data merged;
  merge class stats;
  by sex;
  label mean_height = "mean(height)";
  label mean_weight = "mean(weight)";
run;

proc sgpanel data=merged;
  panelby sex;
  histogram weight;
  density weight;
  inset mean_weight mean_height /
      position=topright textattrs=(style=italic)
      title="Averages";
run;
```

KEYLEGEND Statement

Adds a legend to the plot.

Tip: You can define a legend item that can be included in the legend. For more information, see "LEGENDITEM Statement" on page 438.

See: "Using Discrete Legends" on page 1791

Syntax

KEYLEGEND <"*name-1*" ..."*name-n*"> </*options*>;

Summary of Optional Arguments

Appearance options

ACROSS=*n*
: specifies the number of columns in the legend.

BORDER

NOBORDER
: specifies whether the border around the legend is visible.

DOWN=*n*
: specifies the number of rows in the legend.

LINELENGTH=*numeric-value*<*units*>
: specifies the length of the line glyph for line entries in the legend.

OUTERPAD=*numeric-value*<*units*> | (*pad-options*)
: specifies the amount of extra space that is added outside the legend border.

POSITION=*position-value*
: specifies the position of the legend within the graph outside of the axes.

TITLE="*text-string*"
: adds a title to the legend.

TITLEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the title.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the legend value labels.

Legend options

"*name-1*" ... "*name-n*"
: specifies the names of one or more plots that you want to include in legend.

EXCLUDE=("*item-name1*" <"*item-name2*" <...>>)
: specifies a list of legend entries to exclude from the display.

SORTORDER=ASCENDING | DESCENDING | REVERSEAUTO
: specifies the sort order to use for the legend entry labels.

TYPE=FILL | FILLCOLOR | LINE | LINECOLOR | LINEPATTERN | MARKER | MARKERCOLOR | MARKERSYMBOL
: specifies which visual attributes to display for legend entries in the legend.

Marker options

AUTOITEMSIZE
: specifies that all markers, lines, and filled symbols in the legend are sized in proportion to the font size that is used for the legend entry labels.

AUTOOUTLINE
: specifies that the outline settings in the plot statements determine whether the fill swatches in the legend have outlines.

FILLASPECT=GOLDEN | *positive-number*
: specifies an aspect ratio for the fill swatches based on their height.

FILLHEIGHT=*numeric-valueunits*
: specifies the height of the fill swatches.

SCALE=*positive-number*
: specifies a scale factor that is to be applied to the fill swatch height.

Optional Arguments

"name-1" ... "name-n"
: specifies the names of one or more plots that you want to include in legend. Each name that you specify must correspond to a value that you entered for the NAME= option in a plot statement.

You can specify names to control which plots are included in the legend. In addition, by specifying one or more names, you can use the KEYLEGEND statement to create a legend when no legend is created automatically.

Note: A KEYLEGEND statement without a "*name*" reference affects only the automatic legend.

Default If you do not specify a name, then the legend contains references to all of the plots in the graph.

Note The names specified here determine which plots are included, but not the labels that appear in the legend for those plots. To specify labels, use the LEGENDLABEL= option on the plot statements.

ACROSS=*n*
specifies the number of columns in the legend. By default, the number of columns is determined automatically.

Note Depending on the number of legend entries and the number of columns and rows that you specify, the legend might not fit in your graph. If your legend does not appear, then you might need to specify a different value for the ACROSS= option.

AUTOITEMSIZE
specifies that all markers, lines, and filled symbols in the legend are sized in proportion to the font size that is used for the legend entry labels.

These proportional sizes take effect regardless of the attributes that are used in the plot. The line segments are drawn as deemed appropriate by the legend, regardless of the line thickness reported by the plot. As a result, the thickness of the line segments in the legend might differ from what is shown in the plot.

The following figures show a legend with fairly large labels. In the first figure, the color swatches are small compared to the labels. The second figure uses AUTOITEMSIZE to size the color swatches in proportion to the labels.

Default Swatch Size	AUTOITEMSIZE Used in the Statement

Tip Use the VALUEATTRS= option to control the font size for the legend value labels.

AUTOOUTLINE
specifies that the outline settings in the plot statements determine whether the fill swatches in the legend have outlines.

Default When this option is not specified, the legend fill swatches are always outlined.

Note The outlines are always 1 pixel wide with a solid pattern.

BORDER | NOBORDER
specifies whether the border around the legend is visible. The BORDER option shows the border. The NOBORDER option hides the border.

Default BORDER

DOWN=*n*
specifies the number of rows in the legend. By default, the number of rows is determined automatically.

Note Depending on the number of legend entries and the number of columns and rows that you specify, the legend might not fit in your graph. If your legend does not appear, then you might need to specify a different value for the DOWN= option.

EXCLUDE=(*"item-name1"* <*"item-name2"* <...>>)
specifies a list of legend entries to exclude from the display.

Default No items are excluded.

Requirement Each item name must be enclosed in quotation marks and separated from adjacent names by a space.

Note When the specified names are compared with the legend entry names, leading blanks are honored and trailing blanks are ignored.

Tip For plots with groups, you can exclude specific group values.

Example The following example excludes items Truck and Wagon from the legend.

```
exclude=("Truck" "Wagon")
```

FILLASPECT=GOLDEN | *positive-number*
specifies an aspect ratio for the fill swatches based on their height.

GOLDEN
specifies the golden ratio of 1.618 (width = 1.618 * height).

positive-number
specifies a custom aspect ratio.

Default The system determines the aspect.

Interaction This option is ignored when AUTOITEMSIZE is specified.

Tip Use FILLHEIGHT= to specify the height.

FILLHEIGHT=*numeric-valueunits*
specifies the height of the fill swatches. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The system determines the height.

Interaction This option is ignored when AUTOITEMSIZE is specified.

Tip Use FILLASPECT= to specify the aspect ratio.

LINELENGTH=*numeric-value<units>*
specifies the length of the line glyph for line entries in the legend. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The length is determined automatically by the system.

OUTERPAD=*numeric-value<units>* | (*pad-options*)
specifies the amount of extra space that is added outside the legend border. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

numeric-value<units>
specifies a dimension to use for the extra space at the left, right, top, and bottom of the legend border.

(*pad-options*)
specifies a space-separated list of pad options, enclosed in parentheses.

Specify one or more of the following name-value pair options:

LEFT=*numeric-value<units>*	specifies the amount of extra space added to the left side.
RIGHT=*numeric-value<units>*	specifies the amount of extra space added to the right side.
TOP=*numeric-value<units>*	specifies the amount of extra space added to the top.
BOTTOM=*numeric-value<units>*	specifies the amount of extra space added to the bottom.

Note Sides that are not assigned padding are padded with the default amount.

Tip Use *pad-options* to create non-uniform padding.

Default No padding

Note The default units for *dimension* are pixels. If you want to specify values in other units, then you must specify the units with the value. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

POSITION=*position-value*
specifies the position of the legend within the graph outside of the axes. The positions are as follows:

BOTTOM
places the legend at the bottom of the graph.

LEFT
places the legend at the left side of the graph.

RIGHT
places the legend at the right side of the graph.

TOP
places the legend at the top of the graph.

Default BOTTOM

Note By default, if you use more than one KEYLEGEND statement, then each legend is placed in a different position.

SCALE=*positive-number*
specifies a scale factor that is to be applied to the fill swatch height. Values greater than 1 increase the height while values less than 1 reduce the height.

Default 1

Restriction This option does not apply to markers with pattern fills.

Interaction This option is ignored when AUTOITEMSIZE is specified.

Tips Use FILLHEIGHT= to change the base height.

Use FILLASPECT= to specify the width.

SORTORDER=ASCENDING | DESCENDING | REVERSEAUTO
specifies the sort order to use for the legend entry labels. The REVERSEAUTO value reverses the order of the legend entry labels.

Defaults When the procedure contains more than one plot statement, the default is based on the order of the plot statements.

For grouped data, the default is derived from the data order. If you are using data that is stored in a table, the default is ASCENDING.

Interaction This option overrides the order that is set by any constituent plot statement's GROUPORDER= option.

Note Legend entries are sorted by the formatted values of the data.

TITLE="*text-string*"
adds a title to the legend. The title is placed to the left of the legend body, except in the following cases:

- the legend contains two or more rows of items
- the legend title length exceeds the space that is available on the left side of the legend

In those cases, the title is placed above the legend body.

Defaults In an auto-generated legend, no title is displayed by default unless the legend shows group values. If the legend shows group values,

the group variable label is displayed when a label is assigned, or the group variable name is displayed when a label is not assigned.

In a custom legend created using plot NAME= options and a KEYLEGEND statement, no legend title is displayed by default, even when the legend displays group values.

Requirement *text-string* must be enclosed in quotation marks.

Tips Use this option to add a title to a custom legend.

To suppress the default title in an auto-generated legend, specify TITLE="" in a KEYLEGEND statement as shown in the following example:

```
proc sgpanel data=sashelp.prdsale;
  panelby year;
  vline month / response=actual group=product;
  keylegend / title="";
run;
```

TITLEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the title. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
TITLEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TITLEATTRS=GraphTitleText
```

TYPE=FILL | FILLCOLOR | LINE | LINECOLOR | LINEPATTERN | MARKER | MARKERCOLOR | MARKERSYMBOL

specifies which visual attributes to display for legend entries in the legend. The legend entries can be distinguished as fills, lines, or markers.

FILL
displays the fill colors and filled-area outline colors that are used in the plots as outlined color swatches.

Filled areas include bars, bubbles, and so on.

FILLCOLOR
displays the fill colors that are used in the plots as outlined color swatches.

The swatch outline color is controlled by the style reference GraphOutlines.contrastColor in the current style. The swatch outline thickness is always 1 pixel.

LINE
displays the line colors and line patterns that are used in the plots.

LINECOLOR
displays the line colors and outline colors that are used in the plots as outlined color swatches.

The swatch outline color is controlled by the style reference GraphOutlines.contrastColor in the current style. The swatch outline thickness is always 1 pixel.

LINEPATTERN
displays the line patterns or outline patterns that are used in the plots.

The line color in the legend items is the same as the legend text color, which is controlled by the style reference GraphValueText:contrastColor in the current style.

MARKER
displays the marker symbols and marker colors that are used in the plots.

MARKERCOLOR
displays the marker colors that are used in the plots as outlined color swatches.

The swatch outline color is controlled by the style reference GraphOutlines.contrastColor in the current style. The swatch outline thickness is always 1 pixel.

MARKERSYMBOL
displays the marker symbols that are used in the plots.

The marker color in the legend items is the same as the legend text color, which is controlled by the style reference GraphValueText:contrastColor in the current style.

Default All attributes are displayed (FILL, LINE, and MARKER).

Note If a statement contributing to the legend does not have any visual attributes that match the TYPE specified, then the legend does not display any entries from that statement.

Tip This option can be used as a filter.

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the legend value labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color attribute of the GraphValueText style element in the current style. For grouped data, the ContrastColor

attribute of a GraphData1–GraphData*n* style element in the current style.

GraphValueText style element in the current style. The effective attributes are FontFamily, FontSize, FontStyle, and FontWeight.

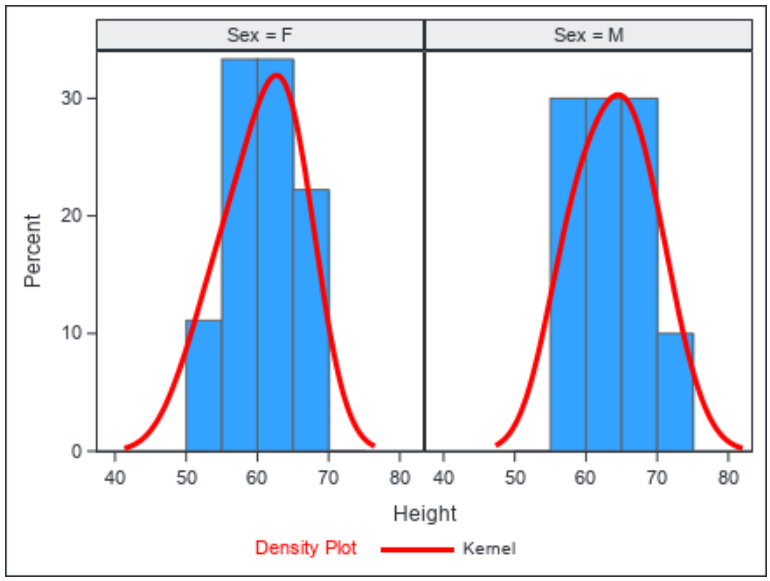
Examples

```
valueattrs=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:

```
valueattrs=GraphTitleText
```

Example




```
proc sgpanel data=sashelp.class noautolegend;
  panelby sex;
  histogram height;
  density height /
    type=kernel
    name="kernel"
    lineattrs=(color = red);
  keylegend "kernel" /
    title="Density Plot"
    titleattrs=(color = red);
run;
```

LEGENDITEM Statement

Creates a legend item that can be included in the legend. The LEGENDITEM statement must be used with one or more plot statements. The procedure can contain multiple LEGENDITEM statements.

Notes: When used with multiple plot statements, the order of the LEGENDITEM statement among the plot statements determines the legend item's position in the automatic legend.

Any of the supported attribute options can be specified for a legend item regardless of its type. However, attribute options that do not apply to the specified TYPE= value are ignored. For example, the MARKERATTRS= option sets the attributes for a marker and is useful if you set TYPE=MARKER or TYPE=MARKERLINE. However, if TYPE= is set to a value that does not display a marker symbol, then the MARKERATTRS= option is ignored. Similarly, if you specify a marker legend item that uses a filled marker type, then specifying NOFILL has no effect.

See: "KEYLEGEND Statement" on page 430

Syntax

LEGENDITEM TYPE=*type* NAME="*text-string*" <*/options*>;

Summary of Optional Arguments

Appearance options

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency when TYPE= is set to FILL.

FILLEDOUTLINEDMARKERS
: specifies that a filled marker also has an outline.

LINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the line when TYPE= is set to LINE or MARKERLINE.

MARKERATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the marker when TYPE= is set to MARKER or MARKERLINE.

NOFILL

NOOUTLINE
: hides the fill or outline of the fill swatch for the legend item

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the outline when TYPE= is set to FILL.

TEXT="*text-string*"
: specifies the text to display when TYPE= is set to TEXT.

TEXTATTRS=*style-element*<(*text-options*)> | (*text-options*)
: specifies the color and font properties of the legend item when TYPE is set to TEXT.

Label options

LABEL="*text-string*"
: specifies a label to be used for the item in the legend.

LABELATTRS=*style-element*<(*options*)> | (*options*)
: specifies the color and font attributes of the legend item's label.

Required Arguments

TYPE=FILL | MARKER | MARKERLINE | LINE | TEXT
specifies the type of legend item.

FILL
specifies a fill (displayed as a filled square).

The appearance can be managed with the FILLATTRS= and OUTLINEATTRS= options.

MARKER
specifies a marker.

The appearance can be managed with the MARKERATTRS= option.

MARKERLINE
specifies a marker and a line.

The appearance can be managed with the MARKERATTRS= and LINEATTRS= options.

LINE
specifies a line.

The appearance can be managed with the LINEATTRS= option.

TEXT
specifies text that can be displayed in the legend area.

The text string is defined with the TEXT= option, and the text appearance can be managed with the TEXTATTRS= option.

NAME="*text-string*"
assigns a name to the legend item for reference in a KEYLEGEND statement.

Note The *text-string* is case-sensitive and must define a unique name within the procedure. The *text-string* must be enclosed in quotation marks.

Optional Arguments

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency when TYPE= is set to FILL. You can specify the color by using a style element or by specifying a specific option. If you specify a style element, you can also specify *options* to override the color.

Note: TRANSPARENCY cannot be derived from the style element, but it can be set in the *options*.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect if you specify the NOFILL option.

FILLEDOUTLINEDMARKERS
specifies that a filled marker also has an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the “Filled” suffix in its name. Otherwise, this option is ignored.

Interactions This option has no effect unless TYPE= is set to MARKER or MARKERLINE.

Use the FILLATTRS= and OUTLINEATTRS= options to specify attributes for the fill and outline.

Note When this option is specified, the default fill color is the color that is in effect for the FILLATTRS= option. The default line color is the color that is in effect for the OUTLINEATTRS= option. The default transparency is the transparency in effect for the MARKERATTRS= option (the transparency in the FILLATTRS= option is not honored).

LABEL="*text-string*"
specifies a label to be used for the item in the legend.

Default No label is displayed

Tip You can specify the font and color attributes for the label with the LABELATTRS= option.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the color and font attributes of the legend item’s label. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see “Text Attributes” on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Interaction For this option to take effect, the LABEL= option must also be specified.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the line when TYPE= is set to LINE or MARKERLINE. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see “Line Attributes and Patterns” on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphDataDefault style element in the current style.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker when TYPE= is set to MARKER or MARKERLINE. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default The ContrastColor, MarkerSize, and MarkerSymbol attributes of the GraphDataDefault style element in the current style.

NOFILL | NOOUTLINE
hides the fill or outline of the fill swatch for the legend item

Default The fill and outline are shown.

Interactions If NOFILL and NOOUTLINE are both specified, then both options are ignored.

If you specify a marker legend item that uses a filled marker type, then specifying NOFILL has no effect.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the outline when TYPE= is set to FILL. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the outline.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness

Interaction This option has no effect if NOOUTLINE is also specified.

TEXT="*text-string*"
specifies the text to display when TYPE= is set to TEXT.

Default If this option is not specified in a legend item with TYPE=TEXT, then a default blank space is used.

Tip You can specify the appearance of the text with the TEXTATTRS= option.

TEXTATTRS=*style-element*<(*text-options*)> | (*text-options*)
specifies the color and font properties of the legend item when TYPE is set to TEXT. You can specify the appearance by using a style element or by specifying

specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Details

The LEGENDITEM statement creates a definition for a legend item that can be included in a legend. The item that you define is independent of the data and enables you to customize the legend to enhance or replace a standard legend. For example, to display annotation text within the legend area, you can define a LEGENDITEM statement with TYPE=TEXT and specify the text in the TEXT= option.

The LEGENDITEM statement must be used with one or more plot statements within the procedure. The legend item is included in the automatically generated legend. The legend contains the legend item along with entries for the other plot statements. Note the following behaviors:

- When used with multiple plot statements, the order of the LEGENDITEM statement among the plot statements determines the legend item's position in the automatic legend.
- If a plot statement includes a group, then only the group plot shows up in the automatic legend by default.

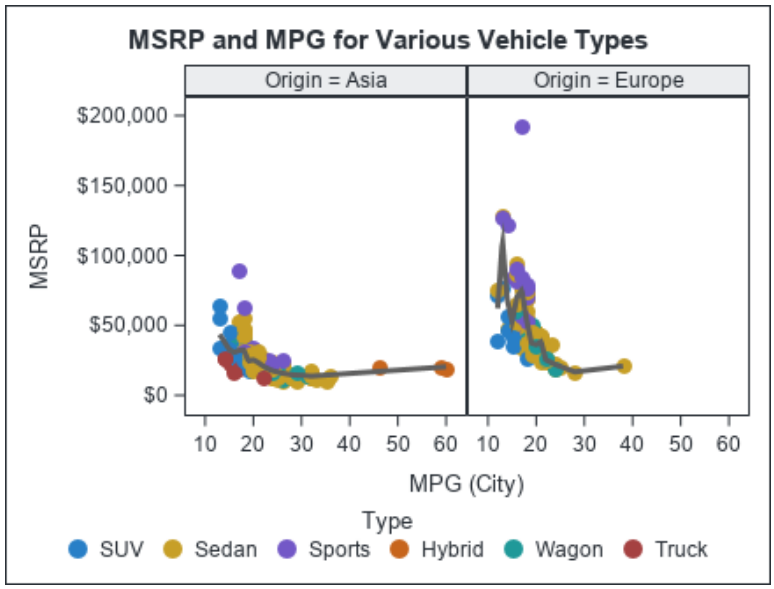
  You can override the automatic legend functionality with the KEYLEGEND statement. A single KEYLEGEND statement can reference one or more legend item names the same way it references plot names. Each legend item name that you specify must correspond to a value that you entered for the NAME= option in the LEGENDITEM statement.

  See also "KEYLEGEND Statement" on page 430.

Example

This example scatter plot shows the city miles per gallon (MPG) and the manufacturer's suggested retail price (MSRP) for a group of vehicles. A Loess fit plot is also included. Without the LEGENDITEM statement, the auto-generated legend shows each of the six group values for vehicle type. The legend does not include the Loess plot.

Figure 8.8 Graph without the LEGENDITEM or KEYLEGEND Statements



In this example, the goal is to present the overall trend of vehicle prices and MPG rather than focus on individual vehicle types. There is no need for the detailed legend shown in the previous figure.

The example shows how to customize the legend to do the following:

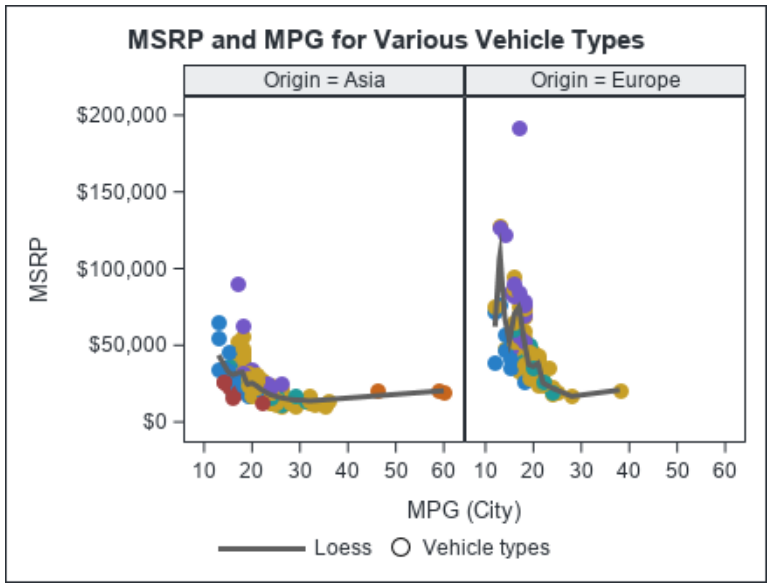
- combine all the vehicle types into a single legend entry
- include the Loess plot

In the following program, the LEGENDITEM statement is used to represent all the vehicle types within a single legend entry. The legend item is a marker that matches the scatter plot that it represents. The marker is given a neutral color so as not to be associated with any one of the vehicle types. The KEYLEGEND on page 430 statement includes both the legend item and the Loess plot.

```
title "MSRP and MPG for Various Vehicle Types";
proc sgpanel data=sashelp.cars;
  where origin in ("Asia" "Europe");
  panelby origin;
  scatter x=mpg_city y=msrp / group=type;
  loess x=mpg_city y=msrp / nomarkers name="fit";
  legenditem type=marker name="car_types" /
        label="Vehicle types"
        markerattrs=(color=black symbol=circle);
        keylegend "fit" "car_types";
run;
title;
```

The result is a legend with a single entry for vehicles and an entry for the Loess plot.

Output 8.1 *Graph with the LEGENDITEM and KEYLEGEND Statements*



LINEPARM Statement

Creates a straight line specified by a point and a slope. You can generate a single line by specifying a constant for each required argument. You can generate multiple lines by specifying a numeric variable for any or all required arguments.

Requirement: The LINEPARM statement must be used with another plot statement that is derived from data values that provide boundaries for the axis area. For example, it can be used with a scatter plot or a histogram. See "Plot Compatibility" on page 1780.

Example: "About Parameterized Lines" on page 27

Syntax

LINEPARM X=*numeric-value* | *numeric-variable*
Y=*numeric-value* | *numeric-variable*
SLOPE=*numeric-value* | *numeric-variable*
</option(s)>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

LINEATTRS=*style-element*<(***options***)> | (*options*)

specifies the appearance of the line.

NOEXTEND
prevents the line from being extended beyond the axis offset.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS=MAX | MIN
specifies the location of the curve label.

LEGENDLABEL="*text-string*"
specifies a label that identifies the plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text used for curve labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

CLIP
specifies that the data for the line is ignored when determining the data ranges for the axes.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*numeric-value* | *numeric-variable*
specifies the X coordinate of a point.

Notes Values are in the units of the data.

If the value specified for the X= option is outside of the data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

If you specify a variable, and the variable contains any missing values, no line is drawn for the respective observation.

Y=*numeric-value* | *numeric-variable*
specifies the Y coordinate of a point.

Notes Values are in the units of the data.

If the value specified for the Y= option is outside of the data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

If you specify a variable, and the variable contains any missing values, no line is drawn for the respective observation.

SLOPE=*numeric-value* | *numeric-variable*
specifies the slope of the line. The slope can be positive or negative.

Tips SLOPE=0 creates a horizontal line (parallel to the X axis).

SLOPE= . (missing value) creates a vertical line (parallel to the Y axis).

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

CLIP
specifies that the data for the line is ignored when determining the data ranges for the axes. Each axis scale is determined by the other plots in the overlay. This might result in the line not being displayed if its data range is not within the data ranges of the other plots. This option ensures that the line is displayed.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the label from the Y variable is used as the curve label. For grouped data, the group values are used as curve labels.

Restriction This option is not valid when a constant value is specified for X=, Y=, and SLOPE= and the GROUP= option is not in effect. To label the curves in that case, use CURVELABEL="*text-string*" instead.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall

outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is valid only when a constant value is specified for X=, Y=, and SLOPE= and the GROUP= option is not in effect. If a column is specified for one or more of X=, Y=, or SLOPE, or if the GROUP= option is in effect, CURVELABEL="*text-string*" is ignored. To label the curves in that case, use CURVELABEL instead.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
```

```
Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:

```
CURVELABELATTRS=GraphTitleText
```

CURVELABELPOS=MAX | MIN
specifies the location of the curve label. Specify one of the following values:

MIN
places the curve label at the minimum value for the X axis.

MAX
places the curve label at the maximum value for the X axis.

Default MAX

Interaction This option has no effect unless CURVELABEL is also specified.

GROUP=*variable*
specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

LEGENDLABEL="*text-string*"
specifies a label that identifies the plot in the legend. By default, the label "LineParm" is used (if you specify a numeric value for Y) or the Y variable name is used (if you specify a variable for Y).

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOEXTEND
prevents the line from being extended beyond the axis offset. When specified, there might be a gap between the line and the axis. The gap is controlled by the axis offset. If the offset is set to 0, then there is no gap.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text used for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless CURVELABEL is also specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See “Fit Policies for Axis Tick Values, Curve Labels, and Data Labels” on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See “Fit Policies for Axis Tick Values, Curve Labels, and Data Labels” on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

LOESS Statement

Creates a fitted loess curve.

Restriction: The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

See: "Marker Fills and Outlines" on page 1805

"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Loess Plots" on page 43

Syntax

LOESS X=*numeric-variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fit curve.

NOMARKERS

removes the scatter markers from the plot.

SMOOTH=*numeric-value*
specifies a smoothing parameter value.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
specifies a classification variable to divide the values into groups.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Legend options

LEGENDLABEL="*text-string*"

specifies a label that identifies the fit line in the legend.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

Limit options

CLM

CLM="*text-string*"
creates confidence limits.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY=*numeric-value*
specifies the degree of transparency for the confidence limits.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

JITTER
specifies that data markers are offset when multiple observations have the same response value.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

Plot options

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

DEGREE=1 | 2
specifies the degree of the local polynomials to use for each local regression.

INTERPOLATION=CUBIC | LINEAR
specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

MAXPOINTS=*n*
specifies the maximum number of predicted points for the loess fit and the corresponding limits.

REWEIGHT=*n*
specifies the number of iterative reweighting steps to apply to the data.

WEIGHT=*numeric-variable*
specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*numeric-variable*
specifies the variable for the X axis.

Y=*numeric-variable*
specifies the variable for the Y axis.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default .05

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect unless you also specify the CLM option.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CLM
CLM="*text-string*"
creates confidence limits. The optional text string overrides the default legend label for the confidence limit. The confidence level is set by the ALPHA= option.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

Note: The CLMATTRS= option has no effect unless you also specify the CLM option.

style-element
specifies the name of a style element.

CLM-attrs-suboptions
specifies one or more CLM suboptions.

Specify one or more of the following suboptions:

CLMFILLATTRS=*style-element* <(*options*)> | (*options*)
Specify the fill attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interactions CLMFILLATTRS= has no effect unless the fill is displayed. The fill is displayed by default. You can manually control this setting with the FILL and NOFILL suboptions.

The transparency specified with this option overrides the CLMTRANSPARENCY= option.

Tip If you want to display only the outlines, you can specify the OUTLINE and NOFILL suboptions.

Example
```
clmattrs=(clmfillattrs=(color=red transparency=.8))
```

CLMLINEATTRS=*style-element* <(*options*)> | (*options*)
Specify the line attributes of the confidence limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can specify the OUTLINE option.

Default For ungrouped data, the GraphConfidence style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interaction Specifying OUTLINE also hides the fill color, unless FILL is also specified.

Example
```
clmattrs=(outline clmlineattrs=(color=red thickness=8))
```

FILL
shows the fill color.

Default FILL

NOFILL
hides the fill color.

Default FILL

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

NOOUTLINE
hides the outlines.

Default NOOUTLINE

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

OUTLINE
shows the outlines

Default NOOUTLINE

Interaction The CLMATTRS= option has no effect unless you also specify the CLM option.

CLMTRANSPARENCY=*numeric-value*
specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default 0.0

Interaction This option takes effect only if you also specify the CLM option.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than

one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label. Specify one of the following values:

MIN
places the label at the part of the curve closest to the minimum X-axis value.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

START
places the curve label at the first point on the curve.

END
places the curve label at the last point on the curve.

Default END

Interaction This option has no effect unless the CURVELABEL option is also specified.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction This option has no effect if you also specify the NOMARKERS option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DEGREE=1 | 2
specifies the degree of the local polynomials to use for each local regression. 1 specifies a linear fit and 2 specifies a quadratic fit.

Default 1

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interaction Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

GROUP=*variable*
specifies a classification variable to divide the values into groups. A separate plot is created for each unique value of the classification variable.

Interactions If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

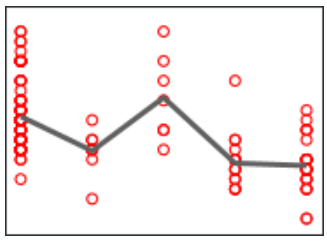
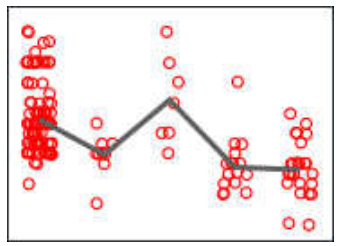
When the GROUP= option is used, the data is rendered in sorted order by the group variable.

INTERPOLATION=CUBIC | LINEAR
specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

Default LINEAR

JITTER
specifies that data markers are offset when multiple observations have the same response value. When the JITTER option is enabled, markers that represent the same response value are offset slightly in order to make all of the markers visible.

The following partial images show the effect of the JITTER option.

Default	JITTER Specified
	

Default Markers that represent the same response value are overlaid, which results in some markers being obscured.

Notes This option affects only how the scatter plot is drawn. It has no effect on the LOESS curve.

By default, the width of the jitter space is 40% of the minimal interval width of the axis. When the minimum data interval is very small, the jitter offset might not be noticeable.

LEGENDLABEL="*text-string*"
specifies a label that identifies the fit line in the legend. By default, the label "Loess" is used, along with the value of the SMOOTH= option if specified.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fit curve. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphFit style element in the current style.

For grouped data, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphFit style element in the current style.

Note If you overlay multiple Loess plots using ungrouped data, the procedure uses GraphFit and GraphFit2 for the first two plots. If three or more plots are requested, the GraphData1–GraphData*n* style elements are used instead for the ContrastColor and LineStyle attributes. In this case, the LineThickness attribute comes from the GraphFit element.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interaction Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MAXPOINTS=*n*
specifies the maximum number of predicted points for the loess fit and the corresponding limits.

Default 201

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

NOMARKERS
removes the scatter markers from the plot.

REWEIGHT=*n*
specifies the number of iterative reweighting steps to apply to the data.

Default 0

SMOOTH=*numeric-value*
specifies a smoothing parameter value.

Default A smoothing value is determined automatically.

Range (0,1]

See What is loess regression? on blogs.sas.com.

The LOESS Procedure in *SAS/STAT User's Guide* for information about the fitting methodology that the LOESS statement uses.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Note The option affects only the scatter plot in this statement.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement except the confidence limits. The CLMTRANSPARENCY option must still be used to control transparency for confidence limits.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

NEEDLE Statement

Creates a plot with needles connecting each point to the baseline.

Restriction: The vertical axis that is used with the NEEDLE statement cannot be a discrete axis.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Needle Plots" on page 29

Syntax

NEEDLE X=*variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all needle lines from discrete X values.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the needle lines.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped needles.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*

specifies the location of the data label with respect to the plot.

LEGENDLABEL="*text-string*"
specifies a label that identifies the needle plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Marker options

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERS
adds markers to the tips of the needles.

Plot options

BASELINE=*numeric-value*
specifies a value on the Y axis for the baseline.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*numeric-variable*
specifies a numeric variable for the Y axis.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BASELINE=*numeric-value*
specifies a value on the Y axis for the baseline. The baseline is always displayed in the plot, even when this option is not specified. In that case, the default value

is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the axis.

Default 0

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

For interval data, when markers are displayed and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 8.28 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS

MATTE	PRESSED	SHEEN

If you specify markers with the plot, then the data skin affects the markers as well.

Table 8.29 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	**PRESSED**	**SHEEN**

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See “Using Data Skins” on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all needle lines from discrete X values. Specify a value from –0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the X axis is discrete.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.

Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped needles.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interactions This option is ignored unless GROUP= is specified.

For interval data, when markers are displayed, and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced. The size could be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

LEGENDLABEL="*text-string*"
specifies a label that identifies the needle plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the needle lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interactions This option has no effect unless you also specify the MARKERS option.

Suboption COLOR= in option LINEATTRS= overrides the default marker color.

MARKERS
adds markers to the tips of the needles.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(variable-list)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

PBSPLINE Statement

Creates a fitted penalized B-spline curve.

Restriction: The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

See: "Marker Fills and Outlines" on page 1805

"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Penalized B-Spline Plots" on page 44

Syntax

PBSPLINE X=*numeric-variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

LINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the fitted curve.

NKNOTS=*n*
: specifies the number of evenly spaced internal knots.

SMOOTH=*numeric-value*
: specifies a smoothing parameter value.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Label options

CURVELABEL
: adds a label to the curve or curves.

CURVELABEL="*text-string*"
: specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS=MIN | MAX | START | END
: specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
: displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*

specifies the location of the data label with respect to the plot.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Legend options

LEGENDLABEL="*text-string*"
specifies a label that identifies the B-spline curve in the legend.

NOLEGCLI
hides the legend entry for the individual value prediction limits.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

Limit options

CLI

CLI="*text-string*"
creates confidence limits for the individual predicted values.

CLIATTRS=*style-element* | (*CLI-attrs-suboptions*)
specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying line and fill attributes.

CLM

CLM="*text-string*"
creates confidence limits.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY=*numeric-value*
specifies the degree of transparency for the confidence limits.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

JITTER
specifies that data markers are offset when multiple observations have the same response value.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

NOMARKERS

removes the scatter markers from the plot.

PBSPLINE options

MAXPOINTS=*n*
: specifies the maximum number of predicted points for the spline curve and for any confidence limits.

Plot options

ALPHA=*numeric-value*
: specifies the confidence level for the confidence limits.

DEGREE=*n*
: specifies the degree of the spline transformation.

FREQ=*numeric-variable*
: specifies a variable for the frequency count for each observation in the input data.

GROUP=*variable*
: specifies a classification variable to divide the values into groups.

WEIGHT=*numeric-variable*
: specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
: specifies a name for the plot.

Required Arguments

X=*numeric-variable*
specifies the variable for the X axis.

Y=*numeric-variable*
specifies the variable for the Y axis.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default .05

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect unless you specify the CLI or CLM option (or both).

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CLI
CLI="*text-string*"
creates confidence limits for the individual predicted values. The optional text string overrides the default legend label for the prediction limits. The confidence level is set by the ALPHA= option.

CLIATTRS=*style-element* | (*CLI-attrs-suboptions*)
specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying line and fill attributes.

Note: The CLIATTRS= option has no effect unless you also specify the CLI option.

CLI-attrs-suboptions can be the following suboptions:

CLIFILLATTRS=*style-element* <(*options*)> | (*options*)
Specify the fill attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Default The Color and Transparency attributes of the GraphConfidence style element in the current style.

Interaction The CLIFILLATTRS= suboption has no effect unless you also specify the FILL suboption.

Example
```
cliattrs=(fill clifillattrs=(color=yellow transparency=.8))
```

CLILINEATTRS=*style-element* <(*options*)> | (*options*)
Specify the line attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphPredictionLimits style element in the current style.

Example
```
cliattrs=(clilineattrs=(color=orange pattern=longdash thickness=8px))
```

FILL
NOFILL
specifies whether the prediction band is filled.

The FILL suboption shows the fill color. The NOFILL suboption hides the fill color.

Default NOFILL

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

OUTLINE
NOOUTLINE
specifies whether the prediction band has outlines.

The OUTLINE suboption shows the outlines. The NOOUTLINE suboption hides the outlines.

Default OUTLINE

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

Default The default style element for CLIATTRS= is GraphPredictionLimits.

Interaction The CLIATTRS= option has no effect unless you also specify the CLI option.

CLM
CLM="*text-string*"
creates confidence limits. The optional text string overrides the default legend label for the confidence limit. The confidence level is set by the ALPHA= option.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

Note: The CLMATTRS= option has no effect unless you also specify the CLM option.

style-element
specifies the name of a style element.

CLM-attrs-suboptions
specifies one or more CLM suboptions.

Specify one or more of the following suboptions:

CLMFILLATTRS=*style-element* <(*options*)> | (*options*)
Specify the fill attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interactions CLMFILLATTRS= has no effect unless the fill is displayed. The fill is displayed by default. You can manually control this setting with the FILL and NOFILL suboptions.

The transparency specified with this option overrides the CLMTRANSPARENCY= option.

Tip If you want to display only the outlines, you can specify the OUTLINE and NOFILL suboptions.

Example
```
clmattrs=(clmfillattrs=(color=red transparency=.8))
```

CLMLINEATTRS=*style-element* <(*options*)> | (*options*)
Specify the line attributes of the confidence limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can specify the OUTLINE option.

Default For ungrouped data, the GraphConfidence style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interaction Specifying OUTLINE also hides the fill color, unless FILL is also specified.

Example
```
clmattrs=(outline clmlineattrs=(color=red thickness=8))
```

FILL
shows the fill color.

Default FILL

NOFILL
hides the fill color.

Default FILL

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

NOOUTLINE
hides the outlines.

Default NOOUTLINE

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

OUTLINE
shows the outlines

Default NOOUTLINE

Interaction The CLMATTRS= option has no effect unless you also specify the CLM option.

CLMTRANSPARENCY=*numeric-value*
specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default 0.0

Interaction This option takes effect only if you also specify the CLM option.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label. Specify one of the following values:

MIN
places the label at the part of the curve closest to the minimum X-axis value.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

START
places the curve label at the first point on the curve.

END
places the curve label at the last point on the curve.

Default END

Interaction This option has no effect unless the CURVELABEL option is also specified.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction This option has no effect if you also specify the NOMARKERS option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DEGREE=*n*
specifies the degree of the spline transformation.

Default 3

Range 0–10

Restriction DEGREE= and NKNOTS= cannot be set to 0 simultaneously. When both are set to 0, an error results.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled"

suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interaction Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*

specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

GROUP=*variable*

specifies a classification variable to divide the values into groups. A separate plot is created for each unique value of the classification variable.

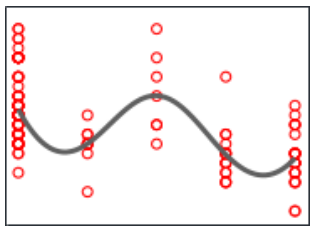
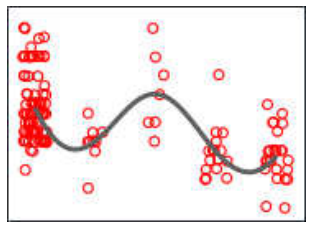
Interactions If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

When the GROUP= option is used, the data is rendered in sorted order by the group variable.

JITTER

specifies that data markers are offset when multiple observations have the same response value. When the JITTER option is specified, markers that represent the same response value are offset slightly in order to make all of the markers visible.

The following partial images show the effect of the JITTER option.

Default	JITTER Specified
	

Default Markers that represent the same response value are overlaid, which results in some markers being obscured.

Notes This option affects only how the scatter plot is drawn. It has no effect on the PBSPLINE curve.

By default, the width of the jitter space is 40% of the minimal interval width of the axis. When the minimum data interval is very small, the jitter offset might not be noticeable.

LEGENDLABEL="*text-string*"
specifies a label that identifies the B-spline curve in the legend. By default, the label "Penalized B-Spline" is used with the SMOOTH= value if specified, or else the group value for each B-spline is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fitted curve. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphFit style element in the current style.

For grouped data, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphFit element in the current style.

Note If you overlay multiple PBSpline plots using ungrouped data, the procedure uses GraphFit and GraphFit2 for the first two plots. If three or more plots are requested, the GraphData1–GraphData*n* style elements are used instead for the ContrastColor and LineStyle attributes. In this case, the LineThickness attribute comes from the GraphFit element.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interaction Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MAXPOINTS=*n*
specifies the maximum number of predicted points for the spline curve and for any confidence limits.

Default 201

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NKNOTS=*n*
specifies the number of evenly spaced internal knots. By default, a large number of knots (100) is specified, which allows for a lack of smoothness in the results. However, the final function is typically much smoother due to the penalty. When SMOOTH=0 is specified, you should typically request fewer knots than the default because there is no penalty for lack of smoothness. For example, ten or fewer knots are usually enough to follow the functional form found in most data.

Default 100

Range 0 to 1000

NOLEGCLI
hides the legend entry for the individual value prediction limits.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

NOMARKERS
removes the scatter markers from the plot.

SMOOTH=*numeric-value*
specifies a smoothing parameter value.

Default A smoothing value that minimizes a lack-of-smoothness penalty is determined automatically.

Range [0,1]

Tip Specify SMOOTH=0 for an ordinary B-spline fit.

See The TRANSREG Procedure in *SAS/STAT User's Guide* for information about the fitting methodology that the PBSPLINE statement uses.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless CURVELABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Note The option affects only the scatter plot in this statement.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable,

using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement except the confidence limits. The CLMTRANSPARENCY option must still be used to control transparency for confidence limits.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction When the CLI= option is used with this option, the confidence band for individual predicted values is displayed as a high-low plot instead of a band.

POLYGON Statement

Draws a polygon from data stored in a data set.

Restrictions: You cannot overlay a polygon with a bar chart using the HBAR or VBAR statements. If you need to overlay a polygon plot on a bar chart, use an HBARPARM or VBARPARM statement instead. See "Plot Compatibility" on page 1780.

The POLYGON statement does not support data that is stored in a table. Only SAS data sets are supported for polygons.

Syntax

POLYGON X=*x-variable* Y=*y-variable* ID=*id-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BACKLIGHT

BACKLIGHT=*number*
: specifies that label text should have a back light of a contrasting color.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

FILL

NOFILL
: specifies whether the polygons are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the polygon outline.

OUTLINE

NOOUTLINE
: specifies whether the polygons have outlines.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

ROTATE=*numeric-column* | *number*
: specifies the angle of rotation for the polygon measured in degrees.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

XOFFSET=*numeric-value* | *numeric-variable*
: specifies an amount to offset all polygon segment starting and ending points from discrete X values.

YOFFSET=*numeric-value* | *numeric-variable*

specifies an amount to offset all polygon segment starting and ending points from discrete Y values.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Group options

GROUP=*variable*
creates a separate polygon color or outline pattern for each unique grouping that is specified.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

LABEL

LABEL=*variable*
specifies a label for the polygon curve.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot.

LABELLOC=INSIDEBBOX | OUTSIDEBBOX
specifies the location of the polygon label.

LABELPOS=CENTER | XMIN | XMAX | YMIN | YMAX
specifies the position of the polygon label.

LEGENDLABEL="*text-string*"
specifies the label that identifies the polygon in the legend.

ROTATELABEL=AUTO | NONE | VERTICAL
specifies the rotation of the polygon label with respect to the rotation of the polygon.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*x-variable*
specifies the variable for the X values.

Y=*y-variable*
specifies the variable for the Y values.

ID=*id-variable*
specifies the variable for the ID values that identify the polygon or polygons. All data rows for a single polygon must have the same ID value. Only unformatted values in the ID= column are used.

Requirement When multiple polygons are defined in the same data set, all rows with the same ID value must be in contiguous rows. The ID values cannot be interspersed.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BACKLIGHT
BACKLIGHT=*number*
specifies that label text should have a back light of a contrasting color. When you specify the BACKLIGHT option, you can also specify a number. *number* specifies the degree of the back-light effect.

The following figures show the effect of applying back light to the label. In these examples, FILL has also been specified.

BACKLIGHT=0	BACKLIGHT=1

The back light is based on text color. For dark colors, a white back-light effect is used. For lighter colors, a black back-light effect is used. The following figures show the back-light effects when full back light is applied (BACKLIGHT=1). In the first two examples, FILL has also been specified. The third example shows green text against a white background (no fill).

Black Text	Gray Text	Green Text

Defaults 0 (no back-light effect) when the BACKLIGHT option is not used.

When no number is specified for the BACKLIGHT option, the default is 0.5 when no GROUP= or COLORRESPONSE= option is used. The default is 0.75 when the GROUP= or COLORRESPONSE= option is used.

Range 0.0–1.0, where 0.0 specifies no effect and 1.0 specifies maximum effect

Note This option is most effective when text color has a low level of contrast with the background. It is also effective when the background is cluttered.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see “Color-Naming Schemes” on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element for filled polygons. The ThreeColorAltRamp style element for unfilled polygons.

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled polygons. Specify one of the following:

Table 8.30 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interaction If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

FILL | NOFILL
specifies whether the polygons are filled. The FILL option shows the fill color for the polygons. The NOFILL option hides the fill color.

Default NOFILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

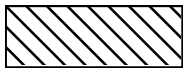
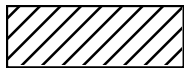
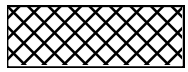
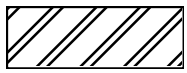
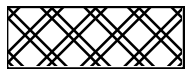
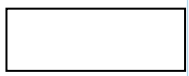
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.31 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

GROUP=*variable*

creates a separate polygon color or outline pattern for each unique grouping that is specified. The plot elements for each group value are automatically distinguished by different visual attributes.

The polygon attributes for each unique group value are derived from the GraphData1–GraphData*n* and GraphMissing style elements. If the polygon is filled, then the COLOR attribute is used for the polygon fill and the CONTRASTCOLOR attribute is used for the polygon outline. If the polygon is not filled, then the CONTRASTCOLOR and PATTERN attributes are used for the polygon outline.

Interactions When this option is used and the value is a variable associated with an ATTRID= option, the attribute mapping defined by the associated attribute map is used.

This option is ignored if the COLORRESPONSE= option is also used.

LABEL
LABEL=*variable*

specifies a label for the polygon curve.

Default When no *variable* is provided, the ID variable is used for labels.

Interaction If a *variable* is supplied, rows that have the same ID value are expected to have the same value. If they do not, the ID value's first row determines the polygon label.

Tip The font and color attributes for the label are specified by the LABELATTRS= option.

LABELATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the labels in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the LABEL= option is also specified.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphValueText
```

LABELLOC=INSIDEBBOX | OUTSIDEBBOX
specifies the location of the polygon label.

INSIDEBBOX
inside the polygon.

OUTSIDEBBOX
outside the polygon but inside the plot area.

Default INSIDEBBOX

Interactions This option has no effect unless the LABEL= option is also specified.

The label's exact position is relative to the polygon's X and Y data ranges and is determined by the combination of this option and the LABELPOS= option.

LABELPOS=CENTER | XMIN | XMAX | YMIN | YMAX
specifies the position of the polygon label. Specify one of the following values:

CENTER
centers the label.

Interaction LABELPOS=CENTER has no effect if you also specify LABELLOC=OUTSIDEBBOX. The default label position is used in this case.

XMIN
places the label at the part of the polygon closest to the minimum X-axis value, centered in the Y-axis range.

XMAX
places the label at the part of the polygon closest to the maximum X-axis value, centered in the Y-axis range.

YMIN
places the label at the part of the polygon closest to the minimum Y-axis value, centered in the X-axis range.

YMAX
places the label at the part of the polygon closest to the maximum Y-axis value, centered in the X-axis range.

Default CENTER

Interactions This option has no effect unless the LABEL= option is also specified.

The label's exact position is relative to the polygon's X and Y data ranges and is determined by the combination of this option and the LABELLOC= option.

When LABELLOC=OUTSIDE, increasing the length of the label might cause the available plot area to decrease. Also, when LABELLOC=OUTSIDE, the label might collide with the axis ticks and tick values.

LEGENDLABEL="*text-string*"
specifies the label that identifies the polygon in the legend.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the polygon outline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the GraphOutlines style element in the current style.

For grouped data, the LineThickness attribute of the GraphOutlines style element, and the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element in the current style.

Interactions This option is ignored if the NOOUTLINE option is also specified.

This option is ignored if the DATASKIN= option is also specified.

Note If this option is used with grouped data, the COLOR= suboption has the effect of holding the polygon outline color constant across all group values.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OUTLINE | NOOUTLINE
specifies whether the polygons have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interaction If NOOUTLINE and NOFILL are both specified, then both options are ignored.

Tip Use the LINEATTRS= option to control the appearance of the outline.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

ROTATE=*numeric-column* | *number*
specifies the angle of rotation for the polygon measured in degrees. Positive angles are measured in the counterclockwise direction, and negative angles are measured in the clockwise direction.

Default 0 (no rotation is performed)

Interaction When this option is specified, the LABELLOC= and LABELPOS= settings are ignored and they are automatically set to INSIDEBBOX and CENTER respectively.

Notes An angle that exceeds 360 degrees in absolute value can be specified.

Rotating a polygon might cause clipping in some cases.

ROTATELABEL=AUTO | NONE | VERTICAL
specifies the rotation of the polygon label with respect to the rotation of the polygon. Specify one of the following:

AUTO
rotates the label with the rotation of the polygon.

NONE
does not rotate the label with the rotation of the polygon.

The label position remains fixed regardless of the polygon rotation.

VERTICAL
rotates the label to a vertical position.

Default AUTO

Interaction If ROTATE= is also specified, then ROTATELABEL=VERTICAL is ignored.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

Default A single space

Interactions This option has no effect unless LABEL= is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see “Using the ODS GRAPHICS Statement” on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

XOFFSET=*numeric-value* | *numeric-variable*
specifies an amount to offset all polygon segment starting and ending points from discrete X values.

Default 0.0 (no offset)

Range –0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between discrete ticks.

Requirement The X axis must be discrete.

YOFFSET=*numeric-value* | *numeric-variable*
specifies an amount to offset all polygon segment starting and ending points from discrete Y values.

Default 0.0 (no offset)

Range –0.5 (downward) to +0.5 (upward), where 0.5 represents half the distance between discrete ticks.

Requirement The Y axis must be discrete.

Details

Overview

The POLYGON statement draws a polygon from a series of X-Y value pairs that are stored in a SAS data set. The first X-Y value pair defines the starting point of the polygon. When the next X-Y pair is encountered in the data set, a line is drawn from the starting point to the second X-Y point. For subsequent X-Y pairs, a line is drawn from the previous X-Y point the current X-Y point. This pattern repeats until all of the segments have been drawn

Using the POLYGON statement, you can draw any data-driven shape on your graph, which enables you to highlight data features, outline data boundaries, and so on. If you later want to modify your polygon, you need only modify the polygon data in the graph data set.

You can specify a “hole” in a polygon. A data row with missing X and Y values indicates that the rows that follow specify the X and Y points for the hole, which is inside the outer polygon.

Here is the sequence:

1. The initial data rows specify the outer polygon.
2. A row with missing X and Y values signals the beginning of a hole.
3. Subsequent data rows specify the X and Y points for the hole.

Multiple holes can be specified. Each hole is specified by a data row with missing X and Y values, followed by a series of rows that specify the X and Y points for the hole.

Requirements for the Polygon Data Set

In the simplest case of a single polygon, your data set must provide an X, Y, and ID column that stores the X-Y values and the ID for your polygon. The X-Y values in the first data set observation must specify the starting point of your polygon. The X-Y values in the subsequent observations must provide the points of each segment in the order in which the polygon is to be drawn. There should be no gaps in the data. In order to close the polygon, you can specify the starting X-Y values in your last observation, although this is not required. If your last observation does not close the polygon, the POLYGON statement automatically draws a segment from your last point back to the starting point in order to close the polygon.

If you want to draw multiple polygons, your ID column must specify a unique identifier string for each polygon. The identifier string associates the observations in the data set with a specific polygon. All of the observations for each individual polygon must be grouped together by ID and must be arranged in the order in which the polygon segments are to be drawn.

Options are available that enable you to customize the polygon and enhance its appearance. For example, you can do the following:

- show or hide the fill and outline, and specify line and fill attributes. You can also apply a data skin as well as transparency to the polygon.
- rotate the polygon.
- offset the polygon from the X or Y axis.
- specify a label, the locations of the label, and label attributes. You can also rotate the label and specify how it fits in the allotted space when there is not enough room to display the text normally.
- specify an HTML page to display when the selectable polygon is clicked.

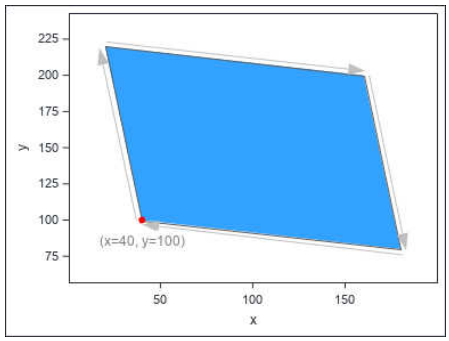
Drawing a Single Polygon

For a single polygon, the POLYGON data set contains an X and Y column that defines the polygon points, and an ID column. The polygon segments are drawn in the order in which they occur in the data. If the polygon overlaps any graphics elements that were drawn earlier, those elements are obscured. In that case, you can use transparency to enable the underlying graphics elements to show through.

Here is a listing of example data for a simple four-sided polygon identified as P1 that starts at point X=40, Y=100.

Obs	id	x	y
1	P1	40	100
2	P1	20	220
3	P1	160	200
4	P1	180	80
5	P1	40	100

The following figure shows how the polygon is drawn.



As shown in the figure, the polygon starting point is X=40, Y=100. The first segment (S1) is drawn from the starting point to X=20, Y=220. The second segment is drawn from X=20, Y=220 to X=160, Y=200. The remaining two segments (S3 and S4) follow the same pattern. Although provided in this example, the last observation (X=40, Y=100) is not required. If not provided in the data, the POLYGON statement draws the last segment (S4) automatically in order to close the polygon.

Drawing Multiple Polygons

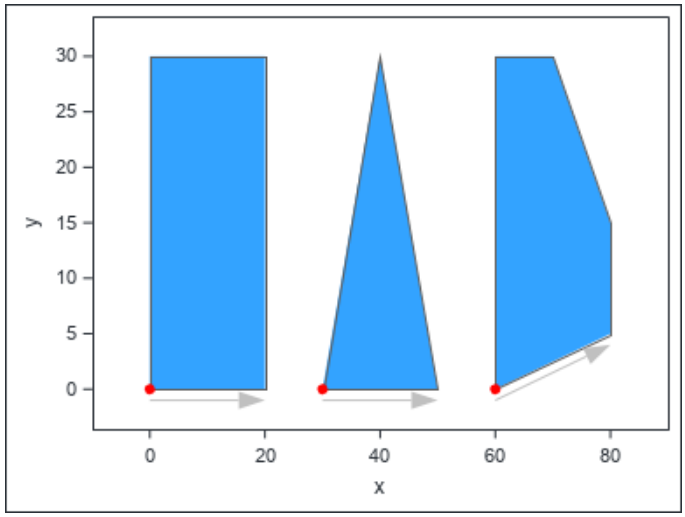
For multiple polygons, the POLYGON data ID column specifies a unique identifier for all of the observations that are associated with each polygon. The X and Y columns specify the polygon points. The polygons are overlaid on the graph in the order in which they occur in the data. For overlapping polygons, each polygon obscures part or all of the polygons and graphics elements that were drawn before it. In that case, you can use transparency to enable the underlying polygons and graphics elements to show through.

Here is a listing of example data for three separate polygons.

Obs	id	x	y
1	1	0	0
2	1	20	0
3	1	20	30
4	1	0	30
5	1	0	0
6	2	30	0
7	2	50	0
8	2	40	30
9	2	30	0
10	3	60	0
11	3	80	5
12	3	80	15
13	3	70	30
14	3	60	30
15	3	60	0

In addition to the ID, X, and Y columns, the LABEL column is added to label the polygons in the output. Notice that the observations for each ID value are grouped together in the data set. The observations for each ID must occur contiguously in the data. Otherwise, unexpected results might occur.

The following figure shows how the polygons are drawn from this data. Reference lines are provided to help you locate the polygon points in the output.



The polygons are drawn in the order in which they appear in the data: rectangle (ID=1), triangle (ID=2), and polygon (ID=3). The red dot on each shape indicates the starting point for that shape, and the gray arrows and dots indicate the subsequent points and drawing progression for each shape.

Drawing Holes in a Polygon

Using the POLYGONPLOT statement, you can draw one or more holes inside a polygon. To create data for a polygon with one or more holes:

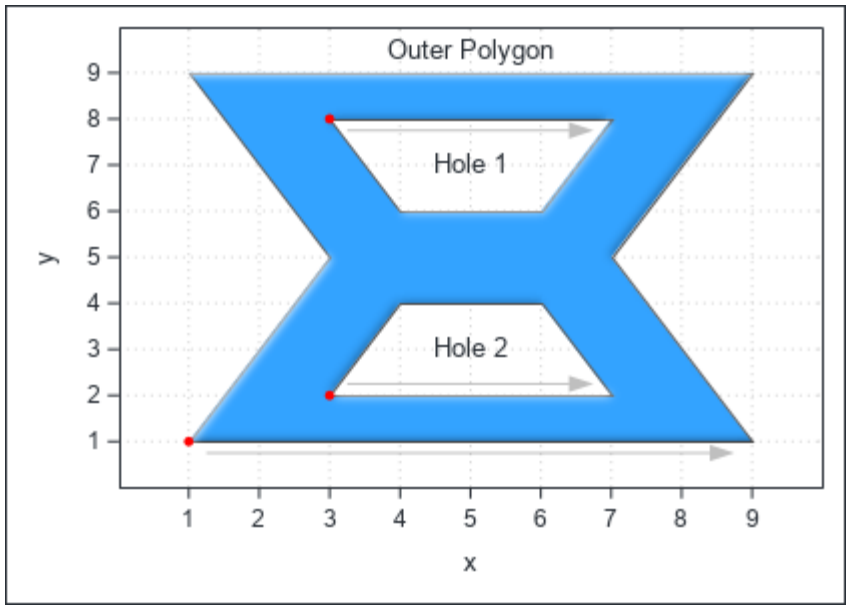
1. Specify the X and Y values for the outer polygon.
2. To start the data for a hole, add an observation that has missing X and Y values. The missing X and Y values signal the POLYGONPLOT statement that the observations that follow define the data for a hole.
3. Specify the X and Y values for the hole polygon.
4. Repeat Steps 2 and 3 for each additional hole.

Here is a listing of example data for a simple polygon that has two holes.

Obs	id	x	y
1	1	1	1
2	1	9	1
3	1	7	5
4	1	9	9
5	1	1	9
6	1	3	5
7	1	1	1
8	1	.	.
9	1	3	8
10	1	7	8
11	1	6	6
12	1	4	6
13	1	3	8
14	1	.	.
15	1	3	2
16	1	7	2
17	1	6	4
18	1	4	4
19	1	3	2

Observations 1–7 specify the data for the outer polygon. In observation 8, the X and Y values are missing, which indicates the start of the data for the first hole. Observations 9–13 define the data for the first hole polygon. Observation 14 indicates the start of the data for the second hole, which is defined by observations 15–19.

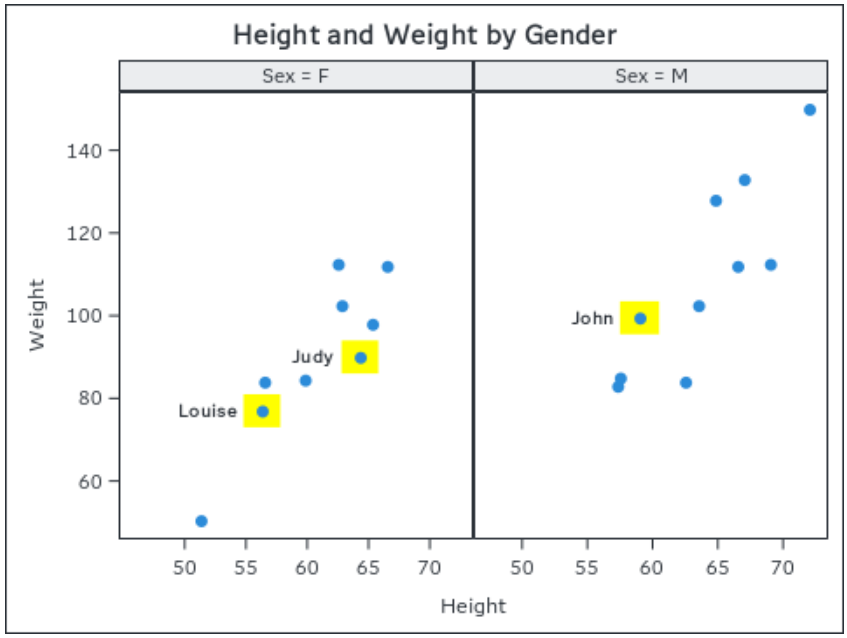
The following figure shows how the polygon is drawn from this data. Grid lines are provided to help you locate the polygon vertices in the output.



The outer polygon is drawn first, starting at point X=1, Y=1. The segments are drawn in data order, which is in a counterclockwise direction as indicated by the gray arrow. The first hole is drawn next, starting at point X=3, Y=8. Its segments are drawn in a clockwise direction. The second hole is drawn last, starting at point X=3, Y=2. Its segments are drawn in a counterclockwise direction.

Example: Highlight Data Items in a Scatter Plot Panel

This example shows how to use shaded rectangles to highlight data items in a panel of scatter plots. The POLYGON statement is used to draw the rectangles. The rectangles highlight the data points for Judy, Louise, and John as shown in the following figure.




```
data polydata;                                    /* 1 */
   set sashelp.class;
   retain id 1;
   if (name in ("Judy" "Louise" "John")) then do;
      x=height-1.5; y=weight-4; output;
      x=height-1.5; y=weight+4; output;
      x=height+1.5; y=weight+4; output;
      x=height+1.5; y=weight-4; output;
      x=height-1.5; y=weight-4; output;
      id=id+1;
   end;

   keep id x y sex name;
run;

data plotdata;                                    /* 2 */
   set sashelp.class polydata;
run;

title "Height and Weight by Gender";
proc sgpanel data=plotdata noautolegend;          /* 3 */
   panelby sex;
   polygon X=x Y=y id=id / tip=(height)           /* 4 */
      fill fillattrs=(color=yellow)
      label=name labelattrs=(weight=bold)
      labelloc=outsidebbox labelpos=xmin;
```

```
   scatter x=height y=weight;                    /* 5 */;
run;
```

1 This DATA step creates the data set Polydata, which stores the polygon data. The data needed to draw a rectangle around the markers for Judy, Louise, and John is generated. A unique ID is also generated for each rectangle. Only the required variables ID, X, Y, SEX, and NAME are retained.

2 This DATA step appends the data in the data set Polydata to data set Sashelp.Class and stores the result in data set Plotdata.

3 The SGPANEL procedure generates the panel from the data in data set Plotdata by variable SEX.

4 The POLYGON statement draws a yellow rectangle around the markers for Judy, Louise, and John that is labeled by name. The polygons are drawn first so that the scatter plot markers overlay the rectangles.

5 The SCATTER statement generates a scatter plot of height and weight for each gender.

REFLINE Statement

Creates a horizontal or vertical reference line.

Requirement: The REFLINE statement must be overlaid with another plot that is derived from data values that provide boundaries for the axis area.

Example: "About Reference Lines" on page 25

Syntax

REFLINE *variable* | *value-1*< ...*value-n*> </ *options*>;

Summary of Optional Arguments

Appearance options

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all lines from discrete X or Y values.

DISCRETETHICKNESS=*numeric-value*
: specifies that the line thickness should be interpreted as a percentage of the mid-point spacing.

LINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the reference line.

NOCLIP

extends the plot axes to contain the reference lines.

TRANSPARENCY=*value*
specifies the degree of transparency for the reference line.

Axis options

AXIS=X | Y
specifies the axis that contains the reference line values.

Label options

LABEL
creates labels for each reference line from the reference line values.

LABEL=*variable-name* | *"text-string"* | (*"text-string-1"* ... *"text-string-n"*)
creates labels for each reference line from values stored in a variable or from one or more specified label values.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels.

LABELPOS=MIN | MAX
specifies the position of the labels.

LEGENDLABEL=*"text-string"*
specifies a label that identifies the plot in the legend.

SPLITCHAR=*"character-list"*
specifies one or more characters on which the text for labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot reference options

NAME=*"text-string"*
specifies a name for the plot.

Required Arguments

variable
draws a reference line for each value of the specified variable.

Requirements Values must be the same type as the data type of the axis. For example, you should use numeric SAS date or time values (or SAS date/time constants) for a time axis.

Unformatted numeric values do not map to a formatted discrete axis. If reference lines are drawn on a formatted discrete axis, the specified variable must use the same format that is used for the axis.

value-1 <... value-n>
draws one or more reference lines at the values that you specify.

Requirements Values must be the same type as the data type of the axis. For example, you should use numeric SAS date or time values (or SAS date/time constants) for a time axis.

Unformatted numeric values do not map to a formatted discrete axis. If reference lines are drawn on a formatted discrete axis, the specified values must be the formatted values as they appear on the axis.

Optional Arguments

AXIS=X | Y
specifies the axis that contains the reference line values. For example, if you specify AXIS= X, vertical reference lines are drawn at points on the X axis.

Default Y

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 8.32 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

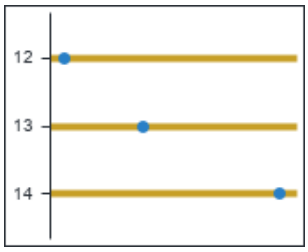
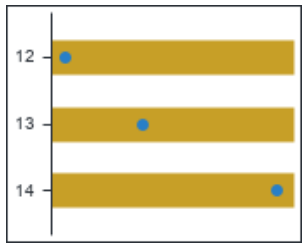
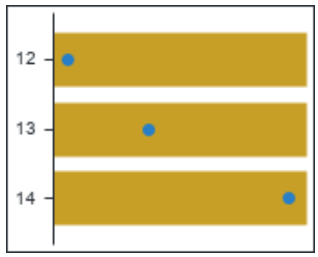
See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from discrete X or Y values. Specify a value from –0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the X or Y axis is discrete.

DISCRETETHICKNESS=*numeric-value*
specifies that the line thickness should be interpreted as a percentage of the mid-point spacing. The following partial images show horizontal reference lines with varying values for DISCRETETHICKNESS=:

DISCRETETHICKNESS=0.1 (Occupies one tenth of the mid-point spacing)	DISCRETETHICKNESS=0.5 (Occupies one half of the mid-point spacing)	DISCRETETHICKNESS=0.75 (Occupies 75% of the mid-point spacing)
		

Default 0.01

Range 0–1

Requirement This option is applicable only when the X or Y axis is discrete.

Interaction When DISCRETETHICKNESS= is used, specifying the LINEATTRS thickness option has no effect.

Note Specifying a value of 0 effectively hides the line.

Tip This feature enables you to display category color bands.

LABEL
creates labels for each reference line from the reference line values.

LABEL=*variable-name* | *"text-string"* | (*"text-string-1"* ... *"text-string-n"*)
creates labels for each reference line from values stored in a variable or from one or more specified label values.

variable-name
specifies the name of a variable that contains the reference line labels.

Restriction The specified label variable is used only when a variable is used for the REFLINE value. If this condition fails, the label variable is ignored and a message is written to the log.

"text-string"
specifies a text string for the label value.

Restriction The label string does not apply when a variable is used for the REFLINE value. In that situation, the label string is ignored and a message is written to the log.

Note If you specify multiple reference values and only one label, the specified label appears on each reference line.

"text-string-1" ... "text-string-n"
specifies a text string for each label value when multiple reference values are specified.

Restriction The label string does not apply when a variable is used for the REFLINE value. In that situation, the label strings are ignored and a message is written to the log.

Note The labels correspond to the reference values in the order specified. When too few labels are specified, some reference lines are not labeled. Excess labels are ignored.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the LABEL option is also specified.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LABELPOS=MIN | MAX
specifies the position of the labels. MIN specifies the label is placed at the minimum value of the data axis. MAX specifies that the label is placed at the maximum value of the data axis.

LEGENDLABEL="*text-string*"
specifies a label that identifies the plot in the legend. By default, the label "Reference Line" is used.

Interaction This option has no effect unless you also specify the NAME= option.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the reference line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphReference style element in the current style.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOCLIP
extends the plot axes to contain the reference lines. By default, if a reference line is created outside of the data range, then the reference line is not visible. This option has no effect if you do not create reference lines that are outside of the data range.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless LABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TRANSPARENCY=*value*
specifies the degree of transparency for the reference line.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Note This option does not affect the reference-line label.

Details

Reference lines are always drawn perpendicular to the axes. They are drawn from one axis boundary to the companion boundary (X to X2 or Y to Y2). Axis offsets do not apply to reference lines. A plot statement that provides a sufficient data range for determining axis extents must be included with the REFLINE statement in the SGPANEL procedure step. For example, a REFLINE statement can be used with a SCATTER or SERIES statement, but it cannot be used by itself. You can specify reference lines by using one or more constant values in the REFLINE statement, or you can specify one or more reference lines by using a data column name in the REFLINE statement. You can use the REFLINE options to add a label to each reference line and to control the appearance of the reference lines and labels.

A reference line can display a data tip in HTML output. Data tips are not enabled by default. To enable data tips, specify the IMAGEMAP option in an ODS GRAPHICS statement. For one or more reference lines that are specified by using constant values in the REFLINE statement, the data tip is formatted as X = *value* or Y = *value*, depending on the specified axis, where *value* is the axis value that was specified for that reference line. For reference lines that are specified by using a column name, the data tip is formatted as *column-name* = *value* or, if a label is assigned to the column, *column-label* = *value*, where *value* is the value in the specified column for that reference line.

REG Statement

Creates a fitted regression line or curve.

Restriction: The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

Interaction: A linear regression (DEGREE=1) cannot be used with logarithmic axes.

See: "Marker Fills and Outlines" on page 1805

"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Examples: "About Regression Plots" on page 45

"Example 2: Creating a Panel of Regression Curves" on page 833

Syntax

REG X=*numeric-variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

LINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the fit line.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
specifies a classification variable to divide the values into groups.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Legend options

LEGENDLABEL="*text-string*"
specifies a label that identifies the regression curve in the legend.

NOLEGCLI
hides the legend entry for the individual predicted value confidence limits.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

Limit options

CLI

CLI="*text-string*"
creates confidence limits for the individual predicted values.

CLIATTRS=*style-element* | (*CLI-attrs-suboptions*)
specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying line and fill attributes.

CLM

CLM="*text-string*"
: creates confidence limits.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
: specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY=*numeric-value*
: specifies the degree of transparency for the confidence limits.

Marker options

FILLEDOUTLINEDMARKERS
: specifies that markers have a fill and an outline.

JITTER
: specifies that data markers are jittered when multiple observations have the same response value.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
: specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the marker outlines.

NOMARKERS
: removes the scatter markers from the plot.

Plot options

ALPHA=*numeric-value*
: specifies the confidence level for the confidence limits.

DEGREE=*n*
: specifies the degree of the polynomial fit.

FREQ=*numeric-variable*
: specifies a variable for the frequency count for each observation in the input data.

MAXPOINTS=*n*
: specifies the maximum number of predicted points for the regression curve and for any confidence limits.

WEIGHT=*numeric-variable*
: specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
: specifies a name for the plot.

Required Arguments

X=*numeric-variable*
specifies the variable for the X axis.

Y=*numeric-variable*
specifies the variable for the Y axis.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default .05

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect unless you specify the CLI or CLM option (or both).

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CLI
CLI="*text-string*"
creates confidence limits for the individual predicted values. The optional text string overrides the default legend label for the prediction limits. The confidence level is set by the ALPHA= option.

CLIATTRS=*style-element* | (*CLI-attrs-suboptions*)
specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying line and fill attributes.

Note: The CLIATTRS= option has no effect unless you also specify the CLI option.

CLI-attrs-suboptions can be the following suboptions:

CLIFILLATTRS=*style-element* <(*options*)> | (*options*)
Specify the fill attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Default The Color and Transparency attributes of the GraphConfidence style element in the current style.

Interaction The CLIFILLATTRS= suboption has no effect unless you also specify the FILL suboption.

Example
```
cliattrs=(fill clifillattrs=(color=yellow transparency=.8))
```

CLILINEATTRS=*style-element* <(*options*)> | (*options*)
Specify the line attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphPredictionLimits style element in the current style.

Example `cliattrs=(clilineattrs=(color=orange pattern=longdash thickness=8px))`

FILL
NOFILL
specifies whether the prediction band is filled.

The FILL suboption shows the fill color. The NOFILL suboption hides the fill color.

Default NOFILL

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

OUTLINE
NOOUTLINE
specifies whether the prediction band has outlines.

The OUTLINE suboption shows the outlines. The NOOUTLINE suboption hides the outlines.

Default OUTLINE

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

Default The default style element for CLIATTRS= is GraphPredictionLimits.

Interaction The CLIATTRS= option has no effect unless you also specify the CLI option.

CLM
CLM="*text-string*"
creates confidence limits. The optional text string overrides the default legend label for the confidence limit. The confidence level is set by the ALPHA= option.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

Note: The CLMATTRS= option has no effect unless you also specify the CLM option.

style-element
specifies the name of a style element.

CLM-attrs-suboptions
specifies one or more CLM suboptions.

Specify one or more of the following suboptions:

CLMFILLATTRS=*style-element* <(*options*)> | (*options*)
Specify the fill attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interactions CLMFILLATTRS= has no effect unless the fill is displayed. The fill is displayed by default. You can manually control this setting with the FILL and NOFILL suboptions.

The transparency specified with this option overrides the CLMTRANSPARENCY= option.

Tip If you want to display only the outlines, you can specify the OUTLINE and NOFILL suboptions.

Example
```
clmattrs=(clmfillattrs=(color=red transparency=.8))
```

CLMLINEATTRS=*style-element* <(*options*)> | (*options*)
Specify the line attributes of the confidence limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can specify the OUTLINE option.

Default For ungrouped data, the GraphConfidence style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interaction Specifying OUTLINE also hides the fill color, unless FILL is also specified.

Example
```
clmattrs=(outline clmlineattrs=(color=red thickness=8))
```

FILL
shows the fill color.

Default FILL

NOFILL
hides the fill color.

Default FILL

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

NOOUTLINE
hides the outlines.

Default NOOUTLINE

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

OUTLINE
shows the outlines

Default NOOUTLINE

Interaction The CLMATTRS= option has no effect unless you also specify the CLM option.

CLMTRANSPARENCY=*numeric-value*
specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default 0.0

Interaction This option takes effect only if you also specify the CLM option.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label. Specify one of the following values:

MIN
places the label at the part of the curve closest to the minimum X-axis value.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

START
places the curve label at the first point on the curve.

END
places the curve label at the last point on the curve.

Default END

Interaction This option has no effect unless the CURVELABEL option is also specified.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction This option has no effect if you also specify the NOMARKERS option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DEGREE=*n*
specifies the degree of the polynomial fit. For example, 1 specifies a linear fit, 2 specifies a quadratic fit, and 3 specifies a cubic fit.

Default 1

Range 1–10

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interaction Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

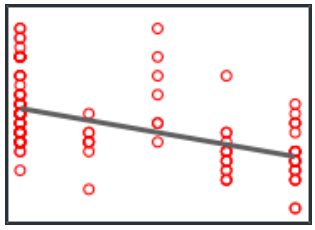
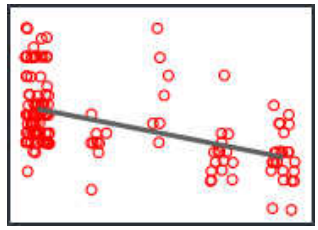
GROUP=*variable*
specifies a classification variable to divide the values into groups. A separate plot is created for each unique value of the classification variable.

Interactions If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

When the GROUP= option is used, the data is rendered in sorted order by the group variable.

JITTER
specifies that data markers are jittered when multiple observations have the same response value. When jittering is enabled, markers that represent the same response value are offset slightly in order to make all of the markers visible.

The following partial images show the effect of jittering.

Default	JITTER Specified
	

Default No jittering. Markers that represent the same response value are overlaid, which results in some markers being obscured.

Notes This option affects only how the scatter plot is drawn. It has no effect on the REG curve.

By default, the width of the jitter space is 40% of the minimal interval width of the axis. When the minimum data interval is very small, the jitter offset might not be noticeable.

LEGENDLABEL="*text-string*"
specifies a label that identifies the regression curve in the legend. By default, the label "Regression" is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fit line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphFit style element in the current style.

For grouped data, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphFit style element in the current style.

Note If you overlay multiple REG plots using ungrouped data, the procedure uses GraphFit and GraphFit2 for the first two plots. If three or more plots are requested, the GraphData1–GraphData*n* style elements are used instead for the ContrastColor and LineStyle attributes. In this case, the LineThickness attribute comes from the GraphFit element.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you

specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interaction Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MAXPOINTS=*n*
specifies the maximum number of predicted points for the regression curve and for any confidence limits.

Default 10

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOLEGCLI
hides the legend entry for the individual predicted value confidence limits.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

NOMARKERS
removes the scatter markers from the plot.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Note The option affects only the scatter plot in this statement.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement except the confidence limits. The CLMTRANSPARENCY option must still be used to control transparency for confidence limits.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction When the CLI= option is used with this option, the confidence band for individual predicted values is displayed as a high-low plot instead of a band.

SCATTER Statement

Creates a scatter plot.

See: "Marker Fills and Outlines" on page 1805
"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types
Example: "About Scatter Plots" on page 31

Syntax

SCATTER X=*variable* Y=*variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all markers from discrete X or Y values.

ERRORBARATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the error bars in the plot.

ERRORCAPSCALE=*numeric-value*
: specifies a positive number to use as a multiplier for changing the default width of the error-bar caps.

LABELSTRIP
: strips leading and trailing blanks from marker characters or data labels with fixed positions before they are displayed in the plot.

NOERRORCAPS
: suppresses the serif cap on error bars, if error bars are displayed.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
: specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
: specifies how to display grouped markers.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot.

LEGENDLABEL="*text-string*"
specifies a label that identifies the markers from the plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

JITTER
specifies that markers that represent the same response value are offset slightly in order to make all of the markers visible.

JITTER=UNIFORM
specifies that data markers should first be jittered and then squeezed together uniformly across all category values.

JITTERWIDTH=*positive-number*
specifies the width of the jitter space as a fraction of either the midpoint spacing or of the minimal interval width.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERCHAR=*variable*
specifies a variable whose values replace the marker symbols in the plot.

MARKERCHARATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot when you use the MARKERCHAR= option.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

Plot options

FREQ=*numeric-variable*
: specifies a variable for the frequency count for each observation in the input data.

URL=*character-variable*
: specifies an HTML page to be displayed when parts of the plot are selected.

XERRORLOWER=*numeric-variable*
: specifies a variable that contains the lower endpoints for the X error bars.

XERRORUPPER=*numeric-variable*
: specifies a variable that contains the upper endpoints for the X error bars.

YERRORLOWER=*numeric-variable*
: specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER=*numeric-variable*
: specifies a variable that contains the upper endpoints for the Y error bars.

Plot reference options

NAME="*text-string*"
: specifies a name for the plot.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*variable*
specifies the variable for the Y axis.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

For interval data, when GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(color-list)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorAltRamp style element

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the marker color, and the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element.

Interaction This option has no effect unless the DATALABEL option is also specified.

Note When the DATALABELPOS= option is specified, the data label font size might be reduced in some cases to avoid overlapping labels and markers.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all marker symbols. If the marker symbol is not filled, then the data skin is applied to the outlines. Specify one of the following:

Table 8.33 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	**PRESSED**	**SHEEN**

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interaction You can use the MARKERATTRS= option to specify a filled marker symbol.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all markers from discrete X or Y values. Specify a value from –0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the X or Y axis is discrete.

ERRORBARATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the error bars in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more of the following options:

COLOR=*color*
specifies the color of the line.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Defaults For ungrouped data, the ContrastColor attribute of the GraphError style element in the current style.

For grouped data, the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

PATTERN=*line-pattern*
specifies the line pattern for the line.

You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default The LineStyle attribute of the GraphError style element in the current style.

THICKNESS=*n <units>*
specifies the thickness of the line.

You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The LineThickness attribute of the GraphError style element in the current style.

ERRORCAPSCALE=*numeric-value*
specifies a positive number to use as a multiplier for changing the default width of the error-bar caps. For example, ERRORCAPSCALE=2 doubles the default cap width and ERRORCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction This option has no effect if NOERRORCAPS is specified.

Tip The thickness (height) of a cap honors the line thickness specified in the ERRORBARATTRS= option. The cap pattern is always solid.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interaction Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped markers.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interactions This option is ignored unless GROUP= is specified.

For interval data, when GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

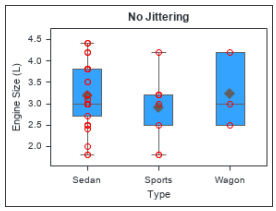
Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

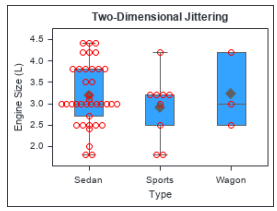
JITTER
specifies that markers that represent the same response value are offset slightly in order to make all of the markers visible. When jittering is enabled, markers that represent the same response value are offset slightly in order to make all of the markers visible. The following examples show how the jittering differs based on the variable type that is being plotted on the X and Y axes.

Note: Although the examples were created using the SGPLOT procedure, the jittering feature works the same in the SGPANEL procedure.

The following figures show a box plot and a scatter plot. The figures show the case in which the X axis is discrete and the Y axis is linear.

Table 8.34 *X Axis Is Discrete and Y Axis Is Linear*





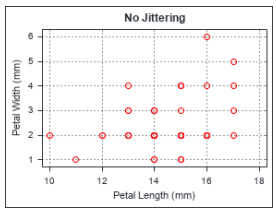
In the first case (No Jittering), markers that represent the same Y value are overlaid, which results in some markers being obscured.

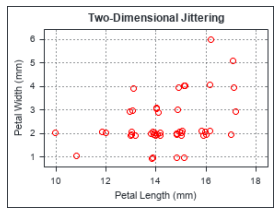
In the second figure, the JITTER option is specified in the SCATTER statement. In this case, one-dimensional systematic jittering occurs along the X axis. The markers that represent the same Y value are offset along the X axis from the midpoint of that value in order to make all of the markers visible.

Note: When the JITTER width is not wide enough to fit long strings of markers, those long strings are squeezed to fit. The other strings that do fit are not squeezed. To force the JITTER option to squeeze all the strings of markers proportionally, specify JITTER=UNIFORM.

The next figures show the case in which both axes are linear. In the first figure (No Jittering), markers that represent the same X and Y bin value are overlaid, which results in some markers being obscured. In the second figure, two-dimensional random jittering occurs along the X and Y axes. The markers are offset randomly along both the X and Y axes in order to make all of the markers visible.

Table 8.35 *Both Axes Are Linear*





Default — When JITTER is not specified, markers that represent the same response value are overlaid. As a result, some markers might be obscured.

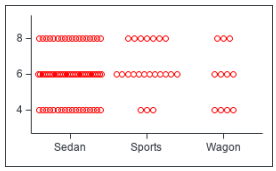
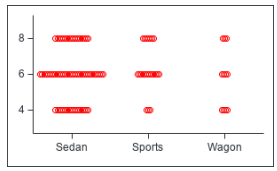
Interaction — The JITTER option is not supported if MARKERCHAR= is also specified. The combination of these two options can produce unpredictable results.

Note — By default, the width of the jitter space is 40% of the minimal interval width of the axis. When the minimum data interval is very small, the jitter offset might not be noticeable.

JITTER=UNIFORM
specifies that data markers should first be jittered and then squeezed together uniformly across all category values. If UNIFORM is not specified, data markers are squeezed together only for those categories whose data markers cannot fit within the jittering space.

The following partial graphs show the effect of specifying the UNIFORM argument.

Table 8.36 *JITTER versus JITTER=UNIFORM*

JITTER	JITTER=UNIFORM
	

Restriction The JITTER=UNIFORM option is available only for discrete, one-dimensional jittering.

JITTERWIDTH=*positive-number*
specifies the width of the jitter space as a fraction of either the midpoint spacing or of the minimal interval width.

Default 0.85

Notes The specified number can be greater than 1.

When used with a discrete axis, changes to JITTERWIDTH= take effect only as long as the markers overlap. Once the specified jitter width reaches the point where the markers are clustered side-by-side without overlapping, further increases to JITTERWIDTH= have no effect on the markers.

LABELSTRIP
strips leading and trailing blanks from marker characters or data labels with fixed positions before they are displayed in the plot. The MARKERCHAR= option specifies the variable that provides the marker strings that are used in place of marker symbols.

Interactions This option affects marker strings only when the MARKERCHAR= option is specified.

This option affects data labels only when DATALABEL is specified.

Tip Stripping the blanks from the numeric value strings helps center each string relative to its data point. Stripping is useful when you want to overlay the data values near or inside the markers for a plot.

LEGENDLABEL="*text-string*"
specifies a label that identifies the markers from the plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interaction Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

MARKERCHAR=*variable*
specifies a variable whose values replace the marker symbols in the plot. If a format is associated with the variable, then the formatted values are used as the marker symbols. If there is not a format associated with the variable and the variable contains numeric data, then the BEST6. format is used.

Interaction The MARKERCHAR= option overrides the DATALABEL= option and the SYMBOL= suboption of the MARKERATTRS= option.

Tip If the MARKERCHAR= variable contains one or more long character strings, large axis offsets might result. In that case, try using the LABELSTRIP option in the SCATTER statement to reduce the offsets. If the result is unsatisfactory, use the OFFSETMIN= and OFFSETMAX= options in the axis statement to override the automatic axis offsets. See "Plot Axes" on page 1784.

MARKERCHARATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot when you use the MARKERCHAR= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the

ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
MARKERCHARATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
MARKERCHARATTRS=GraphValueText
```

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see “Marker Fills and Outlines” on page 1805.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOERRORCAPS
suppresses the serif cap on error bars, if error bars are displayed.

Interaction The appearance of the error bars is controlled by the ERRORBARATTRS= option.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, “Using Range Attribute Maps,” on page 1921

“Overview of Attribute Maps” on page 1897

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(variable-list)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Notes When TIP= is not specified, the X, Y, COLORRESPONSE, and GROUP variables are displayed, if applicable, when you hover over a marker. The Y and XERRORLOWER or XERRORUPPER variables are displayed when you hover over an X error bar or error bar cap.

The X and YERRORLOWER or YERRORUPPER variables are displayed when you hover over a Y error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a marker. No variables are displayed when you hover over an error bar or error bar cap.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

XERRORLOWER=*numeric-variable*
specifies a variable that contains the lower endpoints for the X error bars.

XERRORUPPER=*numeric-variable*
specifies a variable that contains the upper endpoints for the X error bars.

YERRORLOWER=*numeric-variable*
specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER=*numeric-variable*
specifies a variable that contains the upper endpoints for the Y error bars.

SERIES Statement

Creates a series plot.

Restriction: The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

Note: The SERIES statement plots data in data order. For this reason, the input data set should be sorted by the X variable. If it is not sorted by the X variable, unexpected results might occur.

See: "Marker Fills and Outlines" on page 1805
"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Series Plots" on page 33

Syntax

SERIES X=*variable* Y=*variable* </*options*>;

Summary of Optional Arguments

Appearance options

ARROWHEADPOS=START | END | BOTH
specifies a position for arrowheads.

ARROWHEADSCALE=*positive-number*
specifies an arrowhead scale factor based on the thickness of the arrow line.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
specifies a shape for arrowheads.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from discrete X or Y values.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the series line.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

THICKMAX=*numeric-value*<*units*>
specifies the maximum line thickness when a response variable is used to determine the line thickness.

THICKMAXRESP=*numeric-value*
specifies the response value that corresponds to the maximum line thickness.

THICKRESP=*numeric-variable*
specifies a response variable that is used to map a line thickness to each group value.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

GROUPLC=*variable*
specifies a variable that determines the line colors for a grouped plot independently of the GROUP= variable.

GROUPLP=*variable*
specifies a variable that determines the line patterns for a grouped plot independently of the GROUP= variable.

GROUPMC=*variable*
specifies a variable that determines the marker colors for a grouped plot independently of the GROUP= variable.

GROUPMS=*variable*
specifies a variable that determines the marker symbols for a grouped plot independently of the GROUP= variable.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

LCATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPLC= variable.

LPATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPLP= variable.

MCATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPMC= variable.

MSATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPMS= variable.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot.

LEGENDLABEL="*text-string*"
specifies a label that identifies the series plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

MARKERS
adds data point markers to the series plot data points.

Plot options

BREAK
creates a break in the line for each missing value for the Y variable.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME=*"text-string"*
specifies a name for the plot.

Series options

SMOOTHCONNECT
specifies that a smoothed line passes through all vertices.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*variable*
specifies the variable for the Y axis.

Optional Arguments

ARROWHEADPOS=START | END | BOTH
specifies a position for arrowheads. The arrowheads are placed at the starting and ending points using the X values in data order. In order to position the arrowheads properly, you might need to sort the data by the X column.

START displays an arrowhead at the starting point of each line.

END displays an arrowhead at the ending point of each line.

BOTH displays an arrowhead at each end of each line.

Default No arrowheads are displayed when this option is not specified

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSCALE=*positive-number*
specifies an arrowhead scale factor based on the thickness of the arrow line.

Default 1.0

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

Tip Use a factor greater than 1.0 to make a larger arrowhead.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
specifies a shape for arrowheads. The following figure shows each of the arrowhead shapes.

Default OPEN

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BREAK
creates a break in the line for each missing value for the Y variable.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

For interval data, when markers are displayed and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest

effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorAltRamp style element

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label. Specify one of the following values:

MIN
places the label at the part of the curve closest to the minimum X-axis value.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

START
places the curve label at the first point on the curve.

END
places the curve label at the last point on the curve.

Default END

Interaction This option has no effect unless the CURVELABEL option is also specified.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 8.37 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

If you specify markers with the plot, then the data skin affects the markers as well.

Table 8.38 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from discrete X or Y values. Specify a value from –0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the X or Y axis is discrete.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interactions This option has no effect unless MARKERS is also specified.

Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interactions This option is ignored unless GROUP= is specified.

For interval data, when markers are displayed, and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced. The size could be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

GROUPLC=*variable*
specifies a variable that determines the line colors for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series line colors are selected from the GraphData1–GraphData*n* style elements based on the GROUPLC= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

This option overrides the COLOR= suboption of the LINEATTRS= option.

Tips Use the GROUPLP= option to set the line pattern.

You can also use LINEATTRS= option to set the line pattern and thickness.

Use the LCATTRID= option to associate the GROUPLC= variable with an attribute map ID.

See For an overview, see "Example 1: Using Group Options without an Attribute Map" on page 1323 in the SGPLOT procedure.

GROUPLP=*variable*
specifies a variable that determines the line patterns for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series line patterns are selected from the GraphData1–GraphData*n* style elements based on the GROUPLP= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

This option overrides the PATTERN= suboption of the LINEATTRS= option.

Tips Use the GROUPLC= option to set the line color. You can also use the LINEATTRS= option to set the line thickness and color.

Use the LPATTRID= option to associate the GROUPLP= variable with an attribute map ID.

See For an overview, see "Example 1: Using Group Options without an Attribute Map" on page 1323 in the SGPLOT procedure.

GROUPMC=*variable*
specifies a variable that determines the marker colors for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series marker colors are selected from the GraphData1–GraphData*n* style elements based on the GROUPMC= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

The MARKERS option must also be specified for this option to have any effect.

This option overrides the COLOR= suboption of the MARKERATTRS= option.

Tips Use the GROUPMS= option to set the marker symbol. You can also use the MARKERATTRS= option to set the marker size and symbol.

Use the MCATTRID= option to associate the GROUPMC= variable with an attribute map ID.

See For an overview, see "Example 1: Using Group Options without an Attribute Map" on page 1323 in the SGPLOT procedure.

GROUPMS=*variable*
specifies a variable that determines the marker symbols for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series marker symbols are selected from the GraphData1–GraphData*n* style elements based on the GROUPMS= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

The MARKERS option must also be specified for this option to have any effect.

This option overrides the SYMBOL= suboption of the MARKERATTRS= option.

Tips Use the GROUPMC= option to set the marker color. You can also use the MARKERATTRS= option to set the marker size and color.

Use the MSATTRID= option to associate the GROUPMS= variable with an attribute map ID.

See For an overview, see "Example 1: Using Group Options without an Attribute Map" on page 1323 in the SGPLOT procedure.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

LCATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPLC= variable. This option enables the plot to associate an attribute map with the line color group. The option specifies the value of the ID variable that maps attributes to LINECOLOR in the attribute map.

Requirement The procedure statement must specify the attribute map data set using the DATTRMAP= option.

Interactions This option is ignored unless the GROUPLC= option is specified.

This option overrides the COLOR= suboption of the LINEATTRS= option.

Note Do not use this option and the ATTRID= option in the same plot statement. Doing so can cause unexpected results.

See For an overview, see "Example 1: Using Group Options without an Attribute Map" on page 1323.

Chapter 18, "Using Discrete Attribute Maps," on page 1901

LEGENDLABEL="*text-string*"
specifies a label that identifies the series plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the series line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

LPATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPLP= variable. This option enables the plot to associate an attribute map with the line pattern group. The option specifies the value of the ID variable that maps attributes to LINEPATTERN in the attribute map.

Requirement The procedure statement must specify the attribute map data set using the DATTRMAP= option.

Interactions This option is ignored unless the GROUPLP= option is specified.

This option overrides the PATTERN= suboption of the LINEATTRS= option.

Note Do not use this option and the ATTRID= option in the same plot statement. Doing so can cause unexpected results.

See For an overview, see "Example 1: Using Group Options without an Attribute Map" on page 1323.

Chapter 18, "Using Discrete Attribute Maps," on page 1901

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interactions This option has no effect unless you also specify the MARKERS option.

Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

Suboption COLOR= in option LINEATTRS= overrides the default marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color

- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKERS
adds data point markers to the series plot data points.

MCATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPMC= variable. This option enables the plot to associate an attribute map with the marker color group. The option specifies the value of the ID variable that maps attributes to MARKERCOLOR in the attribute map.

Requirement The procedure statement must specify the attribute map data set using the DATTRMAP= option.

Interactions This option is ignored unless the GROUPMC= option is specified.

This option is ignored unless the MARKERS option is specified.

This option overrides the COLOR= suboption of the MARKERATTRS= option.

Note Do not use this option and the ATTRID= option in the same plot statement. Doing so can cause unexpected results.

See For an overview, see "Example 1: Using Group Options without an Attribute Map" on page 1323.

Chapter 18, "Using Discrete Attribute Maps," on page 1901

MSATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPMS= variable. This option enables the plot to associate an attribute map with the marker symbol group. The option specifies the value of the ID variable that maps attributes to MARKERSYMBOL in the attribute map.

Requirement The procedure statement must specify the attribute map data set using the DATTRMAP= option.

Interactions This option is ignored unless the GROUPMS= option is specified.

This option is ignored unless the MARKERS option is specified.

This option overrides the SYMBOL= suboption of the MARKERATTRS= option.

Note Do not use this option and the ATTRID= option in the same plot statement. Doing so can cause unexpected results.

See For an overview, see "Example 1: Using Group Options without an Attribute Map" on page 1323.

Chapter 18, "Using Discrete Attribute Maps," on page 1901

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

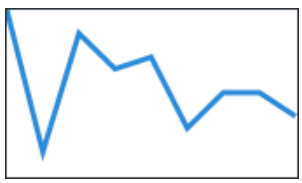
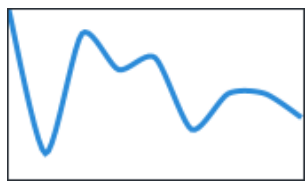
Interaction This option has no effect unless GROUP= is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SMOOTHCONNECT
specifies that a smoothed line passes through all vertices. The following graphics fragments show the effect of using SMOOTHCONNECT.

Default Series	SMOOTHCONNECT Specified
	

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THICKMAX=*numeric-value<units>*
specifies the maximum line thickness when a response variable is used to determine the line thickness. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

By default, this option determines the thickness of the line that represents the maximum response column value.

Default Ten times the thickness that is specified by the GraphDataDefault style element LineThickness attribute.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The THICKMAXRESP= option specifies the response value at which this maximum line thickness is reached. The line thickness for response values that exceed the THICKMAXRESP= value are set to the value that is specified by this option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

THICKMAXRESP=*numeric-value*
specifies the response value that corresponds to the maximum line thickness.

Default The maximum value in the response column that is specified in the THICKRESP= option.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The thickness for all lines that exceed the maximum response value is set to the value specified in the THICKMAX= option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

THICKRESP=*numeric-variable*
specifies a response variable that is used to map a line thickness to each group value.

Default The GraphDataDefault style element LineThickness attribute.

Restriction The THICKRESP= values are assumed to be constant for each group value. If the THICKRESP column has multiple values for a single GROUP value, only one of the THICKRESP= values is used for that group.

Requirement The GROUP= option must be specified with the THICKRESP= option. Otherwise, the THICKRESP= option is ignored.

Interactions When the column values are all zero, all negative, or all missing, this option is ignored. In that case, the default line thickness is used for all of the lines.

The THICKNESS= suboption of the LINEATTRS= option overrides this option for the line thickness attribute.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

TIP=(*variable-list*) | NONE

specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)

a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE

suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

SPLINE Statement

Creates a series plot with a quadratic Bézier spline interpolation that produces smooth curves.

Notes: The SPLINE statement plots data in data order. For this reason, the input data set should be sorted by the X variable. If it is not sorted by the X variable, unexpected results might occur.

The spline plot is not guaranteed to intersect the original data points. For more information, see "Details" on page 588.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Spline Plots" on page 34

Syntax

SPLINE X=*variable* Y=*variable* </*options*>;

Summary of Optional Arguments

Appearance options

ARROWHEADPOS=START | END | BOTH
specifies a position for arrowheads.

ARROWHEADSCALE=*positive-number*
: specifies an arrowhead scale factor based on the thickness of the arrow line.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
: specifies a shape for arrowheads.

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all lines from discrete X or Y values.

LINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the spline line.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

THICKMAX=*numeric-value*<*units*>
: specifies the maximum line thickness when a response variable is used to determine the line thickness.

THICKMAXRESP=*numeric-value*
: specifies the response value that corresponds to the maximum line thickness.

THICKRESP=*numeric-variable*
: specifies a response variable that is used to map a line thickness to each group value.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*

specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

GROUPLC=*variable*
specifies a variable that determines the line colors for a grouped plot independently of the GROUP= variable.

GROUPLP=*variable*
specifies a variable that determines the line patterns for a grouped plot independently of the GROUP= variable.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label.

LEGENDLABEL="*text-string*"
specifies a label that identifies the spline plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

BREAK
creates a break in the line for each missing value for the Y variable.

SEGPOINTS=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the INTERVAL= axis option.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*variable*
specifies the variable for the Y axis.

Optional Arguments

ARROWHEADPOS=START | END | BOTH
specifies a position for arrowheads. The arrowheads are placed at the starting and ending points using the X values in data order. In order to position the arrowheads properly, you might need to sort the data by the X column.

START displays an arrowhead at the starting point of each line.

END displays an arrowhead at the ending point of each line.

BOTH displays an arrowhead at each end of each line.

Default No arrowheads are displayed when this option is not specified

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSCALE=*positive-number*
specifies an arrowhead scale factor based on the thickness of the arrow line.

Default 1.0

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

Tip Use a factor greater than 1.0 to make a larger arrowhead.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
specifies a shape for arrowheads. The following figure shows each of the arrowhead shapes.

Default OPEN

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BREAK
creates a break in the line for each missing value for the Y variable.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interaction This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example
```
colormodel=TwoColorRamp
```

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorAltRamp style element

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label. Specify one of the following values:

MIN
places the label at the part of the curve closest to the minimum X-axis value.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

START
places the curve label at the first point on the curve.

END
places the curve label at the last point on the curve.

Default END

Interaction This option has no effect unless the CURVELABEL option is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 8.39 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS

MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from discrete X or Y values. Specify a value from –0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the X or Y axis is discrete.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interaction This option is ignored unless GROUP= is specified.

GROUPLC=*variable*
specifies a variable that determines the line colors for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series line colors are selected from the GraphData1–GraphData*n* style elements based on the GROUPLC= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

This option overrides the COLOR= suboption of the LINEATTRS= option.

Tips Use the GROUPLP= option to set the line pattern.

You can also use LINEATTRS= option to set the line pattern and thickness.

See For an overview, see Using Group Options on page 1323 in the SGPLOT procedure.

GROUPLP=*variable*
specifies a variable that determines the line patterns for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series line patterns are selected from the GraphData1–GraphData*n* style elements based on the GROUPLP= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

This option overrides the PATTERN= suboption of the LINEATTRS= option.

Tip Use the GROUPLC= option to set the line color. You can also use the LINEATTRS= option to set the line thickness and color.

See For an overview, see Using Group Options on page 1323 in the SGPLOT procedure.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

LEGENDLABEL="*text-string*"
specifies a label that identifies the spline plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the spline line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SEGPOINTS=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the INTERVAL= axis option.

Default 20

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same

split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THICKMAX=*numeric-value<units>*
specifies the maximum line thickness when a response variable is used to determine the line thickness. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

By default, this option determines the thickness of the line that represents the maximum response column value.

Default Ten times the thickness that is specified by the GraphDataDefault style element LineThickness attribute.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The THICKMAXRESP= option specifies the response value at which this maximum line thickness is reached. The line thickness for response values that exceed the THICKMAXRESP= value are set to the value that is specified by this option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

THICKMAXRESP=*numeric-value*
specifies the response value that corresponds to the maximum line thickness.

Default The maximum value in the response column that is specified in the THICKRESP= option.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The thickness for all lines that exceed the maximum response value is set to the value specified in the THICKMAX= option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

THICKRESP=*numeric-variable*
specifies a response variable that is used to map a line thickness to each group value.

Default The GraphDataDefault style element LineThickness attribute.

Restriction The THICKRESP= values are assumed to be constant for each group value. If the THICKRESP column has multiple values for a single GROUP value, only one of the THICKRESP= values is used for that group.

Requirement The GROUP= option must be specified with the THICKRESP= option. Otherwise, the THICKRESP= option is ignored.

Interactions When the column values are all zero, all negative, or all missing, this option is ignored. In that case, the default line thickness is used for all of the lines.

The THICKNESS= suboption of the LINEATTRS= option overrides this option for the line thickness attribute.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

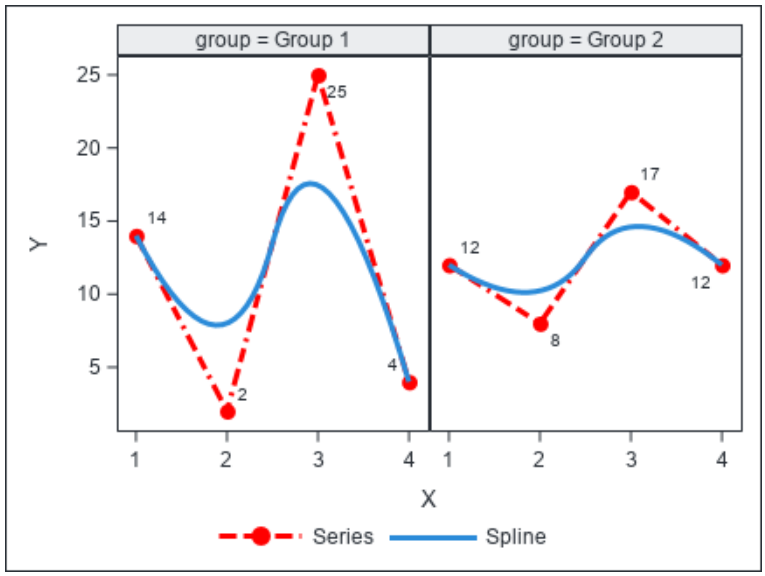
Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

Details

Bézier curves are used to achieve smooth curves. They do this by approximating sequences of line segments.

The following figure shows an example of a spline plot overlaid on a series plot. As shown here, the spline plot does not always intersect the original data points.

Figure 8.9 *Spline and Series Plots*



STEP Statement

Creates a step plot.

Restrictions: The vertical axis that is used with the STEP statement cannot be a discrete axis.

The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

Note: The STEP statement plots data in data order. For this reason, the input data set should be sorted by the X variable. If it is not sorted by the X variable, unexpected results might occur.

See: "Marker Fills and Outlines" on page 1805

"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Step Plots" on page 35

Syntax

STEP X=*variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ARROWHEADPOS=START | END | BOTH
: specifies a position for arrowheads.

ARROWHEADSCALE=*positive-number*
: specifies an arrowhead scale factor based on the thickness of the arrow line.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
: specifies a shape for arrowheads.

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all step lines from discrete X values.

ERRORBARATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the error bars in the plot.

ERRORCAPSCALE=*numeric-value*
: specifies a positive number to use as a multiplier for changing the default width of the error-bar caps.

JUSTIFY=LEFT | CENTER | RIGHT
: specifies the location of each step relative to its data point.

LINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the step line.

NOERRORCAPS
: suppresses the serif cap on error bars, if error bars are displayed.

THICKMAX=*numeric-value*<*units*>
: specifies the maximum line thickness when a response variable is used to determine the line thickness.

THICKMAXRESP=*numeric-value*
: specifies the response value that corresponds to the maximum line thickness.

THICKRESP=*numeric-variable*
: specifies a response variable that is used to map a line thickness to each group value.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)

applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
: specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
: specifies how to display grouped step lines.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the ordering of the groups within a category.

NOMISSINGGROUP
: specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL
: adds a label to the curve or curves.

CURVELABEL="*text-string*"
: specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS=MIN | MAX | START | END
: specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
: displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
: specifies the location of the data label with respect to the plot.

LEGENDLABEL="*text-string*"
: specifies a label that identifies the step plot in the legend.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the text for curve and data labels is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

Marker options

FILLEDOUTLINEDMARKERS
: specifies that markers have a fill and an outline.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

MARKERS
adds markers to the step plot data points.

Plot options

BREAK
creates a break in the line for each missing value.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

YERRORLOWER=*numeric-variable*
specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER=*numeric-variable*
specifies a variable that contains the upper endpoints for the Y error bars.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*numeric-variable*
specifies the variable for the Y axis.

Optional Arguments

ARROWHEADPOS=START | END | BOTH
specifies a position for arrowheads. The arrowheads are placed at the starting and ending points using the X values in data order. In order to position the arrowheads properly, you might need to sort the data by the X column.

START displays an arrowhead at the starting point of each line.

END displays an arrowhead at the ending point of each line.

BOTH displays an arrowhead at each end of each line.

Default No arrowheads are displayed when this option is not specified

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSCALE=*positive-number*
specifies an arrowhead scale factor based on the thickness of the arrow line.

Default 1.0

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

Tip Use a factor greater than 1.0 to make a larger arrowhead.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
specifies a shape for arrowheads. The following figure shows each of the arrowhead shapes.

Default OPEN

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BREAK
creates a break in the line for each missing value.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

For interval data, when markers are displayed and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest

effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the

ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label. Specify one of the following values:

MIN
places the label at the part of the curve closest to the minimum X-axis value.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

START
places the curve label at the first point on the curve.

END
places the curve label at the last point on the curve.

Default END

Interaction This option has no effect unless the CURVELABEL option is also specified.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 8.40 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

If you specify markers with the plot, then the data skin affects the markers as well.

Table 8.41 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all step lines from discrete X values. Specify a value from –0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the X axis is discrete.

ERRORBARATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the error bars in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

For grouped data, the LineStyle and LineThickess attributes of the GraphDataDefault style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

ERRORCAPSCALE=*numeric-value*
specifies a positive number to use as a multiplier for changing the default width of the error-bar caps. For example, ERRORCAPSCALE=2 doubles the default cap width and ERRORCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction This option has no effect if NOERRORCAPS is specified.

Tip The thickness (height) of a cap honors the line thickness specified in the ERRORBARATTRS= option. The cap pattern is always solid.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interactions This option has no effect unless MARKERS is also specified.

Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped step lines.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interactions This option is ignored unless GROUP= is specified.

For interval data, when markers are displayed, and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced. The size could be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

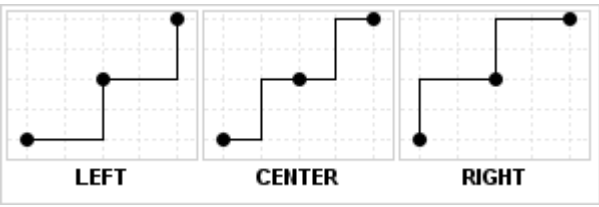
By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

JUSTIFY=LEFT | CENTER | RIGHT
specifies the location of each step relative to its data point. Figure 8.11 on page 599 shows the effect of each option:

Figure 8.10 *Values for JUSTIFY=*



LEGENDLABEL="*text-string*"
specifies a label that identifies the step plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the step line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interactions This option has no effect unless you also specify the MARKERS option.

Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

Suboption COLOR= in option LINEATTRS= overrides the default marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKERS
adds markers to the step plot data points.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOERRORCAPS
suppresses the serif cap on error bars, if error bars are displayed.

Interaction The appearance of the error bars is controlled by the ERRORBARATTRS= option.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See “Fit Policies for Axis Tick Values, Curve Labels, and Data Labels” on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See “Fit Policies for Axis Tick Values, Curve Labels, and Data Labels” on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THICKMAX=*numeric-value<units>*
specifies the maximum line thickness when a response variable is used to determine the line thickness. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

By default, this option determines the thickness of the line that represents the maximum response column value.

Default Ten times the thickness that is specified by the GraphDataDefault style element LineThickness attribute.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The THICKMAXRESP= option specifies the response value at which this maximum line thickness is reached. The line thickness for response values that exceed the THICKMAXRESP= value are set to the value that is specified by this option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

THICKMAXRESP=*numeric-value*
specifies the response value that corresponds to the maximum line thickness.

Default The maximum value in the response column that is specified in the THICKRESP= option.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The thickness for all lines that exceed the maximum response value is set to the value specified in the THICKMAX= option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

THICKRESP=*numeric-variable*
specifies a response variable that is used to map a line thickness to each group value.

Default The GraphDataDefault style element LineThickness attribute.

Restriction The THICKRESP= values are assumed to be constant for each group value. If the THICKRESP column has multiple values for a single GROUP value, only one of the THICKRESP= values is used for that group.

Requirement The GROUP= option must be specified with the THICKRESP= option. Otherwise, the THICKRESP= option is ignored.

Interactions When the column values are all zero, all negative, or all missing, this option is ignored. In that case, the default line thickness is used for all of the lines.

The THICKNESS= suboption of the LINEATTRS= option overrides this option for the line thickness attribute.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

TIP=(*variable-list*) | NONE

specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)

a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE

suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Notes When TIP= is not specified, the X, Y, and GROUP variables are displayed, if applicable, when you hover over a line or marker. The X and ERRORUPPER or ERRORLOWER variables are displayed when you hover over an error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a line or marker. No variables are displayed when you hover over an error bar or error bar cap.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

YERRORLOWER=*numeric-variable*
specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER=*numeric-variable*
specifies a variable that contains the upper endpoints for the Y error bars.

SYMBOLCHAR Statement

Defines a marker symbol using a Unicode character that can be referenced in other statements. The procedure can contain more than one SYMBOLCHAR statement.

Syntax

SYMBOLCHAR NAME=*identifier*
CHAR=*"hex-string"x* | *keyword* < */options* >;

Summary of Optional Arguments

HOFFSET=*offset*
: specifies a horizontal offset for the marker symbol.

ROTATE=*number*
: specifies the angle of rotation for the marker symbol in degrees.

SCALE=*double*
: specifies a scale factor for the marker symbol as a percentage.

TEXTATTRS=*style-element*<(***options***)> | (*options*)
: specifies the text attributes for the character symbol.

VOFFSET=*offset*
: specifies a vertical offset for the marker symbol.

Required Arguments

NAME=*identifier*
specifies a unique identifier for the marker symbol. The identifier can be used in statements that support marker symbols. If multiple SYMBOLCHAR statements are used in a procedure, each statement must have a unique *identifier*. The identifier is not case-sensitive.

Interaction If the identifier matches one of the system-provided symbol names such as CIRCLE, the system symbol is replaced by the user-defined symbol. See marker symbols on page 1834.

Note Do not enclose the identifier in quotation marks.

CHAR="*hex-string*"x | *keyword*
specifies a glyph (character) to be used as the marker symbol. The character is specified using its Unicode specification or its keyword equivalent.

***"hex-string"*x**
specifies a four-byte hexadecimal constant, such as '03c3'x, that represents a Unicode character in the current font.

You can find a complete listing of the Unicode hexadecimal constants at the following URL: http://www.unicode.org/charts/charindex.html

keyword
specifies a SAS keyword for a Unicode character, such as alpha or alpha_u.

See Appendix 2, "Reserved Keywords and Unicode Values," on page 2081.

Note: The "_u" in a keyword makes the character uppercase.

TIP This statement attempts to access the specified Unicode value in the current font. Some fonts do not support accessing characters using their Unicode value while other fonts support only a limited set of Unicode values. If the Unicode value is not accessible, this statement might be ignored or a nonprintable character might be substituted.

Optional Arguments

HOFFSET=*offset*
specifies a horizontal offset for the marker symbol.

Default 0 (the marker symbol is centered on its data point)

Range −0.5 to +0.5, where 0.5 represents one-half of the original marker size.

Interaction The specified offset is also applied to the marker symbol that is displayed in the legend.

Note A positive offset moves the marker symbol to the right within its bounding box, and a negative offset moves it to the left. The bounding box remains centered on the data point. After the offset, the size, and the rotation are applied to the marker symbol, any portion of the symbol that falls outside of the marker bounding box is clipped.

Tip If clipping occurs, you can use this option, the VOFFSET= option, and the SCALE= option to remove the clipping. In the accompanying plot statement that uses markers, you can also use the suboption SIZE= in the MARKERATTRS= option to help remove the clipping.

ROTATE=*number*
specifies the angle of rotation for the marker symbol in degrees. Positive angles are measured in the counterclockwise direction, and negative angles are measured in the clockwise direction.

Default 0 (no rotation is performed)

Note An angle that exceeds 360 degrees in absolute value can be specified.

SCALE=*double*
specifies a scale factor for the marker symbol as a percentage. The scale factor is applied to the character's height.

Default 1.0 (100%)

Range Greater than zero. Very small (for example, 0.1) or very large (for example, 4) scale factors can make the markers invisible or truncated, respectively.

TEXTATTRS=*style-element*<(*options*)> | (*options*)
specifies the text attributes for the character symbol. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Note: Only the text attributes FAMILY=, STYLE=, and WEIGHT= are used. The color and size text attributes are derived from the MARKERATTRS= option in the plot statement.

Default The FontFamily, FontStyle, and FontWeight attributes of the GraphUnicodeText style element in the current style.

Examples
```
TEXTATTRS=(Family='Albany AMT' Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TEXTATTRS=GraphDataText
```

VOFFSET=*offset*
specifies a vertical offset for the marker symbol.

Default 0 (the marker symbol is centered on its data point)

Range −0.5 to +0.5, where 0.5 represents one-half of the original marker size.

Interaction The specified offset is also applied to the marker symbol that is displayed in the legend.

Note A positive offset moves the marker symbol up within its bounding box, and a negative offset moves it down. The bounding box remains centered on the data point. After the offset, the size, and the rotation are applied to the marker symbol, any portion of the symbol that falls outside of the marker bounding box is clipped.

Tip If clipping occurs, you can use this option, the HOFFSET= option, and the SCALE= option to remove the clipping. In the accompanying plot statement that uses markers, you can also use the suboption SIZE= in the MARKERATTRS= option to help remove the clipping.

Details

The SYMBOLCHAR statement defines a custom marker symbol from a Unicode character. The custom marker symbol can be specified in any plot statement that uses markers.

Unicode symbol specifications are not validated at compile time. An invalid specification might not generate a warning when the output is rendered and might create unexpected results.

The custom marker symbol can also be specified for the value of the MARKERSYMBOL variable in an attribute map.

The procedure can contain more than one SYMBOLCHAR statement. In addition, you can have more than one SYMBOLCHAR statement that defines the same character. This enables you to use SYMBOLCHAR statements in IF/ELSE statements.

For an example that uses more than one SYMBOLCHAR statements, see the SAS blog More symbols, you say?.

Note: You might need to specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement to ensure that the data does not use color as the primary distinction between symbols. The need to specify ATTRPRIORITY=NONE depends on which ODS style you are using. If your results do not cycle through the symbols, try specifying ATTRPRIORITY=NONE in the ODS GRAPHICS statement.

You can modify the appearance of markers that are created by the SYMBOCHAR statement in the following ways:

- use the COLOR=, SIZE=, and TRANSPARENCY= suboptions of the MARKERATTRS= option in the plot statement.

  Note: The WEIGHT= suboption has no effect on these markers.

- use the FAMILY=, WEIGHT= , and STYLE= suboptions of the TEXTATTRS= option in the SYMBOLCHAR statement to format the symbol character. The color and size are controlled using the MARKERATTRS= option.

When you use the SCALE= option, you might need to test the output with multiple trials before you find the best output. This is true especially if you also specify the marker size (using the MARKERATTRS= option in the plot statement).

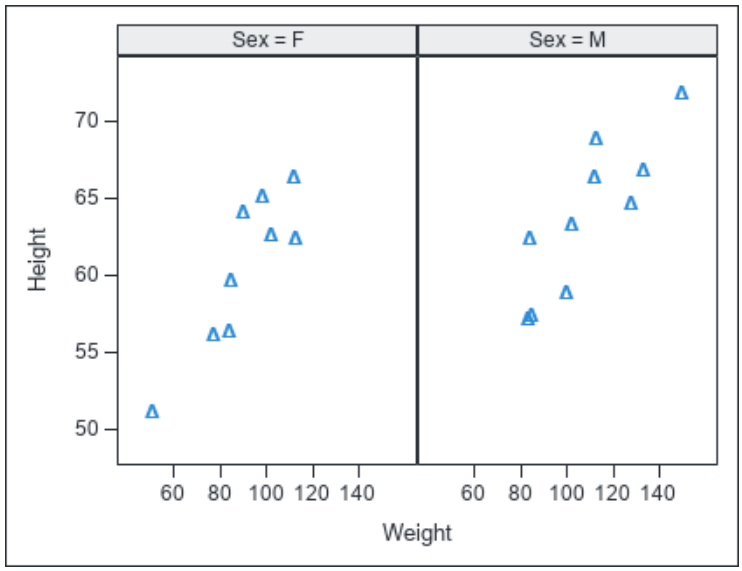
When rendering the graphics output, the procedure performs the following steps:

1. makes adjustments for the HOFFSET and VOFFSET values, if they are specified
2. scales the markers as specified by the SCALE= option
3. clips the markers to the size that is specified in the MARKERATTRS= option in the plot statement
4. rotates the markers, if ROTATE= is specified

Examples

Example 1: Simple SYMBOLCHAR Example

This example specifies the SAS keyword DELTA_U, which produces the delta (Δ) Unicode symbol. A bold font style is applied to the symbol. The scatter plot statement references the symbol name and specifies the marker size.




```
proc sgpanel data=sashelp.class;
   panelby sex;
   symbolchar name=sym1 char=delta_u  /
   textattrs=(Weight=Bold);
   scatter x=weight y=height / markerattrs=(symbol=sym1 size=15pt);
run;
```

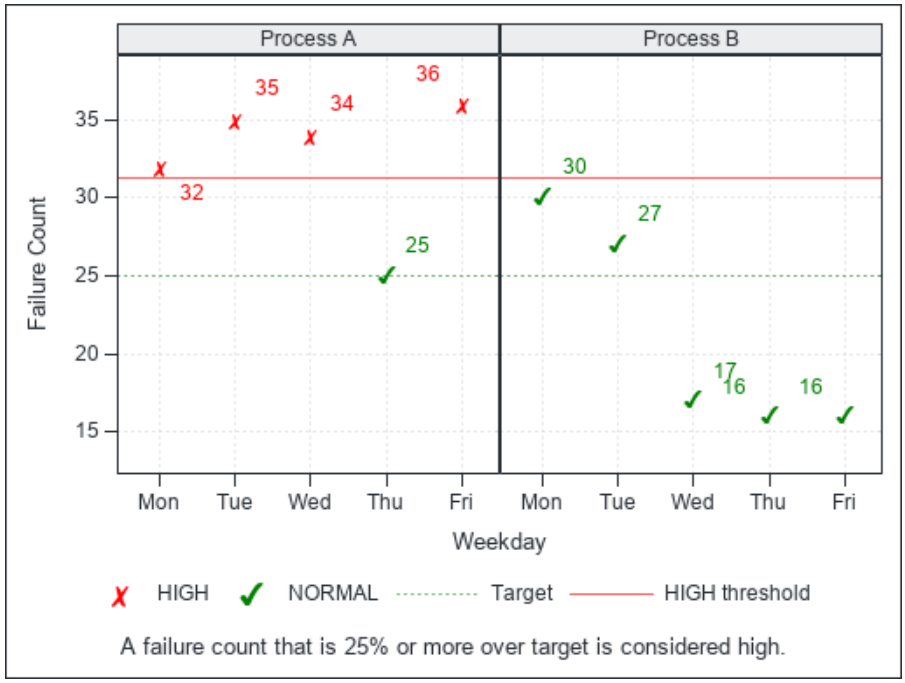
Example 2: SYMBOLCHAR Used with an SG Attribute Map

This example generates different marker symbols for the failure rate in two production processes. The example uses an SG attribute map to assign colors and symbols for each grouping of the variable STATUS.

In the example, the following marker symbols are used:

✓	indicates a failure count that is no more than 25% over the target failure count
✗	indicates a failure count that is over the target failure count by more than 25%

The procedure uses two SYMBOLCHAR statements to identify the marker symbols used in the plot. The example creates marker symbols from the Unicode check mark, heavy character ('2714'x) and the Ballot X character ('2717'x). The example then shows how to use the symbols in a scatter plot.




```
/* Summarize the data in SASHELP.FAILURE to compute the
   failure count by day and process */
proc sort data=sashelp.failure out=temp;
   by day process;
run;

proc summary data=temp;
   by day process;
   var count;
   output out=failure sum=count;
run;

/* Determine the daily status for each process based
   on the target failure count */
%let target=25;
data failure;
   length stat $20;
   set failure;
   if (count <= (&target*1.25)) then status="NORMAL";
   if (count > (&target*1.25)) then status="HIGH";
   sympos = count+2;
   label day="Weekday" count="Failure Count" status="Status";
run;

/* Create the attribute map that assigns
   symbols for each grouping of SIZE. */
data myattrmap;
  length markercolor $15;
  input ID $ value  $ markersymbol $ markercolor $;
  datalines;
myid  NORMAL normal green
myid  HIGH   high   red
;
run;
```

```
/*  Specify the marker symbols used in the scatter plot,
    and apply the SG attribute map. */
footnote "A failure count that is 25% or more over target is
considered high.";
proc sgpanel data=failure dattrmap=myattrmap;
  panelby process / novarname;
  symbolchar name=normal char='2714'x;
  symbolchar name=high char='2717'x;
  scatter x=day y=count / name="scatter" attrid=myid datalabel=count
     markerattrs=(size=18pt) datalabelattrs=(size=9pt)
     group=status;
  refline &target / name="target" lineattrs=(color=green pattern=dot)
     legendlabel="Target";
  refline %sysevalf(&target*1.25) / name="HIGH"
     lineattrs=(color=lightred pattern=solid)
     legendlabel="HIGH threshold";
  rowaxis offsetmax=0.12 grid gridattrs=(color=lightgray pattern=dot);
  colaxis offsetmax=0.1 valuesformat=downame3. grid
     gridattrs=(color=lightgray pattern=dot);
  keylegend "scatter" "target" "HIGH";
  format day downame3.;
run;
footnote;
```

SYMBOLIMAGE Statement

Defines a marker symbol using an image. The marker symbol can then be referenced in other statements.

Syntax

SYMBOLIMAGE NAME=*identifier* IMAGE="*image-file-specification*" < */options* >;

Summary of Optional Arguments

HOFFSET=*offset*
: specifies a horizontal offset for the marker symbol.

ROTATE=*number*
: specifies the angle of rotation for the marker symbol in degrees.

SCALE=*double*
: specifies a scale factor for the marker symbol as a percentage.

VOFFSET=*offset*
: specifies a vertical offset for the marker symbol.

Required Arguments

NAME=*identifier*
specifies an identifier for the marker symbol. The identifier can be used in statements that support marker symbols.

Interaction If the identifier matches one of the system-provided symbol names such as CIRCLE, the system symbol is replaced by the user-defined symbol. See marker symbols on page 1834.

Note Do not enclose the identifier in quotation marks.

IMAGE=*"image-file-specification"*
specifies the name and location of the image file. The supported image types are GIF, JPEG, and PNG.

Requirements The image file specification must be enclosed in quotation marks.

The image file must be located on the local file system. URL access is not supported.

Example
```
image="c:\temp\mylogo.gif"
```

Optional Arguments

HOFFSET=*offset*
specifies a horizontal offset for the marker symbol.

Default 0 (the marker symbol is centered on its data point)

Range –0.5 to +0.5, where 0.5 represents one-half of the original marker size.

Interaction The specified offset is also applied to the marker symbol that is displayed in the legend.

Note A positive offset moves the marker symbol bounding box to the right. A negative offset moves it to the left.

ROTATE=*number*
specifies the angle of rotation for the marker symbol in degrees. Positive angles are measured in the counterclockwise direction, and negative angles are measured in the clockwise direction.

Default 0 (no rotation is performed)

Note An angle that exceeds 360 degrees in absolute value can be specified.

SCALE=*double*
specifies a scale factor for the marker symbol as a percentage. The scale factor is applied to the character's height.

Default 1.0 (100%)

Range Greater than zero. Very small (for example, 0.1) or very large (for example, 4) scale factors can make the markers invisible or truncated, respectively.

VOFFSET=*offset*
specifies a vertical offset for the marker symbol.

Default 0 (the marker symbol is centered on its data point)

Range −0.5 to +0.5, where 0.5 represents one-half of the original marker size.

Interaction The specified offset is also applied to the marker symbol that is displayed in the legend.

Note A positive offset moves the marker symbol bounding box up. A negative offset moves it down.

Details

The SYMBOLIMAGE statement defines a custom marker symbol from an image that is stored in an image file. The image file must exist on the local file system. URL access is not supported. The supported image formats are GIF, JPG, and PNG.

The custom marker symbol can then be specified in any plot statement that supports the MARKERATTRS= option. These plots include the following:

- marker-based plots such as dot and scatter plots.
- line-based plots that enable the addition of markers. These plots include line plots, fit plots (loess, PBspline, regression), needle, series, and step plots.

The custom marker symbol can also be specified for the value of the MARKERSYMBOL variable in an attribute map.

Symbol specifications are not validated at compile time. An invalid specification might not generate a warning when the output is rendered and might create unexpected results.

You can use the SIZE= and TRANSPARENCY= suboptions of the MARKERATTRS= option to modify the appearance of markers that are created by the SYMBOCHAR statement. The COLOR= and WEIGHT= suboptions have no effect on these markers.

The FILLEDOUTLINEDMARKERS= option also has no effect on these markers.

When you use the SCALE= option, you might need to test the output with multiple trials before you find the best output. This is true especially if you also specify the marker size (using the MARKERATTRS= option in the plot statement).

When rendering the graphics output, the procedure performs the following steps:

1 makes adjustments for the HOFFSET and VOFFSET values, if they are specified

2 scales the markers as specified by the SCALE= option

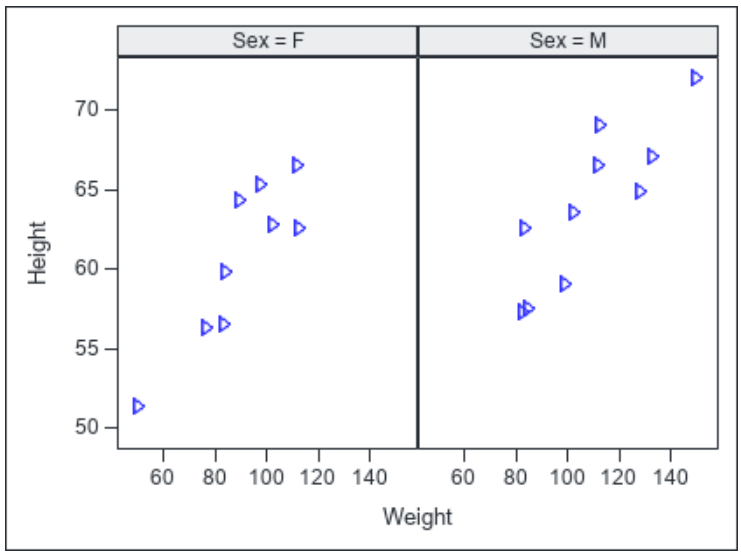
3 clips the markers to the size that is specified in the MARKERATTRS= option in the plot statement

4 rotates the markers, if ROTATE= is specified

Examples

Example 1: Simple SYMBOLIMAGE Example

This example specifies the image file for a triangle (▽). The image is rotated 90 degrees. The scatter plot statement references the symbol name and specifies the marker size.




```
proc sgpanel data=sashelp.class;
   panelby sex;
  symbolimage name=sym1 image="c:\temp\triangle.png" / rotate=90;
  scatter x=weight y=height / markerattrs=(symbol=sym1 size=10pt);
run;
```

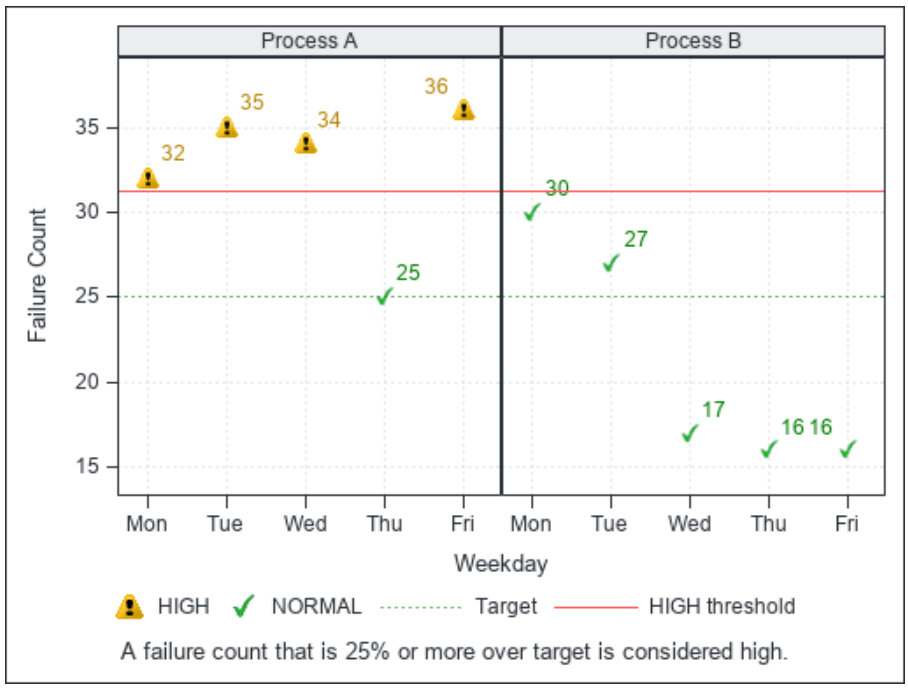
Example 2: SYMBOLIMAGE Used with an SG Attribute Map

This example generates different marker symbols for the failure rate in two production processes. The example uses an SG attribute map to assign colors and symbols for each grouping of the variable STATUS.

In the example, the following images are used:

✓	indicates a failure count that is no more than 25% over the target failure count
⚠	indicates a failure count that is over the target failure count by more than 25%

The procedure uses two SYMBOLIMAGE statements to identify the marker symbols used in the plot. The example then shows how to use the symbols in a scatter plot.




```
/* Summarize the data in SASHELP.FAILURE to compute the
   failure count by day and process */
proc sort data=sashelp.failure out=temp;
   by day process;
run;

proc summary data=temp;
   by day process;
   var count;
   output out=failure sum=count;
run;

/* Determine the daily status for each process based
   on the target failure count. */
%let target=25;
data failure;
   length stat $20;
   set failure;
   if (count <= (&target*1.25)) then status="NORMAL";
   if (count > (&target*1.25)) then status="HIGH";
   sympos = count+2;
   label day="Weekday" count="Failure Count" status="Status";
run;

/* Create the attribute map that assigns
   symbols for each grouping of SIZE. */
data myattrmap;
  length markercolor $15;
  input ID $ value  $ markersymbol $ markercolor $;
  datalines;
```

```
myid  NORMAL normal green
myid  HIGH   high   darkgoldenrod
;
run;

/*  Specify the marker symbols used in the scatter plot,
    and apply the SG attribute map. */
footnote "A failure count that is 25% or more over target is
considered high.";
proc sgpanel data=failure dattrmap=myattrmap;
  panelby process / novarname;
  symbolimage name=normal image='c:\temp\GreenCheck.gif';
  symbolimage name=high image='c:\temp\Alert.gif';
  scatter x=day y=count / name="scatter" attrid=myid datalabel=count
     markerattrs=(size=12pt) datalabelattrs=(size=9pt)
     group=status;
  refline &target / name="target" lineattrs=(color=green pattern=dot)
     legendlabel="Target";
  refline %sysevalf(&target*1.25) / name="HIGH"
     lineattrs=(color=lightred pattern=solid)
     legendlabel="HIGH threshold";
  rowaxis offsetmax=0.12 grid gridattrs=(color=lightgray pattern=dot);
  colaxis offsetmax=0.1 valuesformat=downame3. grid
     gridattrs=(color=lightgray pattern=dot);
  keylegend "scatter" "target" "HIGH";
  format day downame3.;
run;
footnote;
```

TEXT Statement

Displays the associated text values at (X, Y) locations in the graph. The text can be numbers or characters.

Tip: Use the TEXT statement rather than the SCATTER statement with the MARKERCHAR= option when you want more control over the appearance of the text. The TEXT statement enables you to rotate the text to any angle, manage the text position, split the text into multiple lines, display a bounding box around the text, add a back-light effect to the text, and so on.

See: “Plot Compatibility” on page 1780 for information about overlaying this plot with other plot types

Example: “About Text Plots” on page 36

Syntax

TEXT X=*variable* Y=*variable* TEXT=*variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BACKFILL
: displays filled bounding boxes around the text.

BACKLIGHT<=*number*>
: specifies a back-light effect for the marker text.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

CONTRIBUTEOFFSETS=ALL | NONE | (*axis-offset-list*)
: specifies whether this plot's space requirements contribute to the calculation of the axis offsets.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all markers from discrete X or Y values.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

OUTLINE
: displays outlined bounding boxes around the text.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the color of the outlines around the text boxes.

PAD=*dimension*<*units*> | (*pad-options*)
: specifies the amount of extra space that is added inside the text-marker border.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

CLUSTERAXIS=X | Y
: specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
: specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
: specifies how to display grouped text markers.

NOMISSINGGROUP
: specifies that missing values of the group variable are not included in the plot.

Legend options

LEGENDLABEL="*text-string*"
: specifies a label that identifies the markers from the plot in the legend.

Plot options

URL=*character-variable*
: specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
: specifies a name for the plot.

Text options

POSITION=*position-option* | *character-variable*
: specifies the position of the text with respect to the location of the data point.

ROTATE=*number* | *numeric-column*
: specifies the angle of rotation in degrees for the text markers.

SIZEMAX=*numeric-value<units>*
: specifies the maximum font size for the largest text marker when a response variable is used to size the text-marker font.

SIZEMAXRESPONSE=*number*
: specifies the response value that corresponds to the maximum font size for text markers.

SIZEMIN=*numeric-value<units>*
: specifies the minimum font size for text markers when a response variable is used to size the font for text values.

SIZERESPONSE=*numeric-variable*
: specifies a response variable that is used to determine the font size for each text value.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the text is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

SPLITPOLICY=NONE | SPLIT | SPLITALWAYS
specifies a policy for avoiding collisions among the text markers.

SPLITWIDTH=*width-in-characters*
specifies the maximum width of each split line, expressed as a character count.

STRIP
specifies that leading and trailing blanks should be stripped from the marker text before it is displayed.

TEXTATTRS=*style-element*< (*text-options*)> | (*text-options*)
specifies the color and font properties of the marker text.

VCENTER=BBOX | BASELINE
specifies whether the text is vertically centered with respect to the text bounding box or the text baseline.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*variable*
specifies the variable for the Y axis.

TEXT=*variable*
specifies the variable for the text values that are used for the markers.

Tip If the TEXT= variable contains one or more long character strings, large axis offsets might result. In that case, use the OFFSETMIN= and OFFSETMAX= options in the axis statement to override the automatic axis offsets. See "Plot Axes" on page 1784.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BACKFILL
displays filled bounding boxes around the text.

Tips Use the OUTLINE option to specify outlined boxes around the text.

Use the TRANSPARENCY= option to change the transparency of the backfill and the text. To change the transparency of only the backfill, use the TRANSPARENCY= suboption of the FILLATTRS= OPTION.

BACKLIGHT<=*number*>
specifies a back-light effect for the marker text. The effect is applied to the marker text only. *number* specifies the degree of the back-light effect.

The following figures show the effect of applying back light to the text. In these examples, both BACKFILL and OUTLINE have also been specified to help illustrate the back-light effect.

BACKLIGHT=0	BACKLIGHT=1

The back light is based on text color. For dark colors, a white back-light effect is used. For lighter colors, a black back-light effect is used. The following figures show the back-light effects when full back light is applied (BACKLIGHT=1). In the first two examples, BACKFILL and OUTLINE have been specified. The third example shows green text against a white background.

Black Text	Gray Text	Green Text

Defaults 0.5 when no GROUP= or COLORRESPONSE= option is used.

0.75 when the GROUP= or COLORRESPONSE= option is used.

Range 0.0–1.0, where 0.0 specifies no effect and 1.0 specifies maximum effect

Note This option is most effective when text color has a low level of contrast with the background. It is also effective when the background is cluttered.

CLUSTERAXIS=X | Y
specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.

Default Uses the discrete axis for clustering groups when only one axis is discrete. Uses the X axis for clustering if both axes are discrete or interval.

Interaction The GROUPDISPLAY= option must be set to CLUSTER for this option to have any effect. The GROUP= option must also be used.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interaction This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorAltRamp style element

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

CONTRIBUTEOFFSETS=ALL | NONE | (*axis-offset-list*)
specifies whether this plot's space requirements contribute to the calculation of the axis offsets. This value determines which axis offsets can be affected by the plot.

The plot statement can implement an offset that prevents clipping of any text strings that appear at the ends of the axes. The offset is based on the longest string. If the character lengths vary significantly, this can result in wasted space when the shorter strings appear near the ends of the axes. In that case, you can use the CONTRIBUTEOFFSETS= option to modify or eliminate this plot's contribution to the offset calculations in order to reclaim that space.

ALL
the space requirements for this plot are contributed to the axis offset calculations.

NONE
the space requirements for this plot are not contributed to the axis offset calculations.

(*axis-offset-list*)
a space-delimited list of specific contributions that this plot makes to the axis offset calculations.

The list is one or more of the following values enclosed in parentheses:

XMAX	the space requirements for this plot are contributed to the X-axis offset calculation for the maximum end.
XMIN	the space requirements for this plot are contributed to the X-axis offset calculation for the minimum end.
YMAX	the space requirements for this plot are contributed to the Y-axis offset calculation for the maximum end.
YMIN	the space requirements for this plot are contributed to the Y-axis offset calculation for the minimum end.

Default ALL

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all markers from discrete X or Y values.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset)

Requirement This option is applicable only when the X or Y axis is discrete.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect unless BACKFILL is also specified.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interactions When the COLORRESPONSE= option is specified with the GROUP= option, the GROUP= option is ignored.

When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped text markers.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interaction This option is ignored unless GROUP= is specified.

LEGENDLABEL="*text-string*"
specifies a label that identifies the markers from the plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OUTLINE
displays outlined bounding boxes around the text.

Tip Use the BACKFILL option to display filled bounding boxes around the text.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the color of the outlines around the text boxes. You can specify the outline color by using a style element or by specifying the COLOR= option. Only option COLOR= is honored. Options PATTERN= and THICKNESS= are ignored. If you specify a style element, you can also specify the COLOR= option to override the color specification in the current style.

For a description of the line COLOR= option, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor attribute of the GraphOutlines style element in the current style.

For grouped data, the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless OUTLINE is also specified.

PAD=*dimension*<*units*> | (*pad-options*)
specifies the amount of extra space that is added inside the text-marker border.

dimension
specifies a dimension to use for the extra space at the left, right, top, and bottom of the text-marker border.

(*pad-options*)
a space-separated list of one or more of the following name-value-pair options enclosed in parentheses:

LEFT=*dimension*<*units*> TOP=*dimension*<*units*>
RIGHT=*dimension*<*units*> BOTTOM=*dimension*<*units*>

Default Padding is a fraction of the font height.

Note Sides that are not assigned padding are padded with the default amount.

Tips This option is meaningful only when you also specify OUTLINE, BACKFILL, or both.

Use *pad-options* to create non-uniform padding.

Note The default units for *dimension* are pixels. If you want to specify values in other units, then you must specify the desired units with the value. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

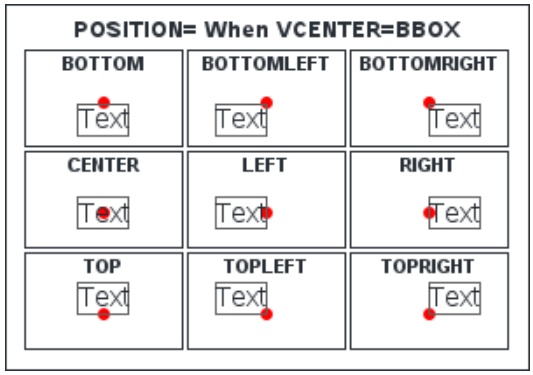
POSITION=*position-option* | *character-variable*

specifies the position of the text with respect to the location of the data point. You can either specify a position option or specify the name of a variable that contains one or more position options.

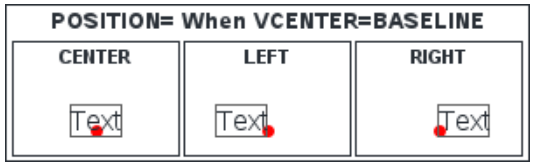
Here are the position options:

BOTTOM	CENTER	TOP
BOTTOMLEFT	LEFT	TOPLEFT
BOTTOMRIGHT	RIGHT	TOPRIGHT

The VCENTER= option specifies whether the position is relative to the text bound box or the text baseline. By default, the positions are relative to the text bounding box. The following figure shows the effect of each of these values on the position of an outlined text when VCENTER=BBOX is in effect. The red dot indicates the data-point location.



When CENTER, LEFT, or RIGHT is specified, and VCENTER=BASELINE is in effect, the positions are relative to the text baseline as shown in the following figure.



Default CENTER

Interaction When *character-variable* is specified and SPLITJUSTIFY= is also specified, the position values specified in *character-variable* are ignored and the text is positioned BOTTOMRIGHT.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

ROTATE=*number* | *numeric-column*
specifies the angle of rotation in degrees for the text markers. Positive angles are measured in the counterclockwise direction, and negative angles are measured in clockwise direction. An angle that exceeds 360 degrees in absolute value can be used.

Default 0

SIZEMAX=*numeric-value<units>*
specifies the maximum font size for the largest text marker when a response variable is used to size the text-marker font. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

By default, the font size of the text values that are associated with the maximum response variable value is set to the value specified by this option.

Default Three times the size specified in the GraphDataText style element for the maximum response variable value marker.

Interactions The SIZERESPONSE= option must be specified for this option to have any effect.

The SIZEMAXRESPONSE= option specifies the response value at which the maximum font size for a text marker is reached. The font size for all text values that exceed the SIZEMAXRESPONSE= value is set to the value specified in this option.

Tips Use the SIZEMAXRESPONSE= option to specify the response value at which the maximum font size for a text marker is reached.

Use the SIZEMIN= option to specify the minimum font size for text markers.

SIZEMAXRESPONSE=*number*
specifies the response value that corresponds to the maximum font size for text markers.

Default The maximum value in the response column specified in the SIZERESPONSE= option.

Interaction The SIZERESPONSE= option must be specified for this option to have any effect.

Note When this option is used with the SIZEMAX= option, the response value is set at the maximum size. Any response values larger than SIZEMAXRESPONSE= are constrained to the SIZEMAX= font size.

SIZEMIN=*numeric-value<units>*
specifies the minimum font size for text markers when a response variable is used to size the font for text values. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The size specified in the GraphDataText style element for the minimum response column value marker.

Interaction The SIZERESPONSE= option must be specified for this option to have any effect.

Tip Use the SIZEMAX= option to specify the maximum text size.

SIZERESPONSE=*numeric-variable*
specifies a response variable that is used to determine the font size for each text value.

Default The size specified in the GraphDataText style element for all text values.

Notes When the variable value for an observation is 0, the font size for the text value for that observation is set to the SIZEMIN= option value.

When the variable value for an observation is negative or missing, the text value for that observation is not displayed in the text plot. However, that observation still contributes to the axis ranges, legend, and so on.

When all the variable values are 0 or missing, this option is ignored. In that case, the default font size is used for all of the text values.

Tip Use the SIZEMIN= and SIZEMAX= options to limit the minimum and maximum font size for the text values.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITPOLICY=NONE | SPLIT | SPLITALWAYS
specifies a policy for avoiding collisions among the text markers.

NONE
does not split the text for text markers that collide.

SPLIT
splits the marker text at a split character only if a split is needed at that character in order to make the text fit the available space.

No split occurs at split characters that occur where a split is not needed. If the text does not contain any of the specified split characters, then a split does not occur. In that case, if the text does not fit the available space, then it might collide with the adjoining text markers.

See See the SPLITCHAR= option for information about specifying the split characters.

SPLITALWAYS
splits the marker text at every occurrence of a split character.

If the text does not contain any of the specified split characters, then a split does not occur.

See See the SPLITCHAR= option for information about specifying the split characters.

Default NONE

SPLITWIDTH=*width-in-characters*
specifies the maximum width of each split line, expressed as a character count. When a width is specified, the marker text is split unconditionally after every *width-in-characters* characters.

Default Uses the width of the longest inter-split-character substring.

Restriction This option has effect only when SPLITPOLICY=SPLIT.

STRIP
specifies that leading and trailing blanks should be stripped from the marker text before it is displayed.

Default Blanks are not stripped

Tip Stripping the blanks from numeric value strings helps center each string relative to its data point.

TEXTATTRS=*style-element*< (*text-options*)> | (*text-options*)
specifies the color and font properties of the marker text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For grouped data, the GraphDataText style element.

For grouped data, the Font attribute of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element.

Interactions When this option's COLOR= suboption is used with the GROUP= option, the color of all of the text markers is specified by the COLOR= suboption.

This option's COLOR= suboption overrides the COLORRESPONSE= option. In that case, if a continuous legend is requested for the plot, the legend is not drawn.

Note When the POSITION= option is specified, the data label font size might be reduced in some cases to avoid overlapping labels.

TIP=(*variable-list*) | NONE

specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)

a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE

suppresses the data tips from this plot.

Restriction: Data tips in text plots are not supported in the SVG output format. To generate a text plot with data tips, use a bitmapped output format such as PNG. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP OUTPUTFMT=PNG;
```

Requirement: You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction: This option replaces all of the information that is displayed by default.

Tip: Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example: tip=(age weight)

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default: The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement: A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction: This option has no effect unless TIP= is also specified.

Note: A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

VCENTER=BBOX | BASELINE

specifies whether the text is vertically centered with respect to the text bounding box or the text baseline.

BBOX

vertically centers the text with respect to its bounding box.

BASELINE

vertically centers the text with respect to the text baseline.

If the text is split into multiple lines, the text is centered on the baseline of the last line of text.

Restriction This option is valid only when POSITION= is set to CENTER, LEFT, or RIGHT. If POSITION= is set to any other value, VCENTER=BBOX is used instead.

Default BBOX

Tip Use the POSITION= option to specify the text position with respect to the text bounding box or the text baseline.

VBAR Statement

Creates a vertical bar chart that summarizes the values of a category variable.

Restrictions: The VBAR statement can be combined only with other vertical categorization plot statements in the SGPANEL procedure. See "Plot Compatibility" on page 1780.

When category plots are overlaid, all of the plots must use the same category variable.

Tips: Bar charts can be combined with basic plot types using the HBARBASIC and VBARBASIC statements.

The VBAR statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

Example: "About Bar Charts" on page 53

Syntax

VBAR *category-variable* </option(s)>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
: specifies whether the baseline is displayed.

FILL

NOFILL
: specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLENDCOLOR=*color*
: specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
: specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
: specifies the thickness of the bars when the category data is numeric.

NOZEROBARS
: suppresses zero-length bars.

OUTLINE

NOOUTLINE
: specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each bar.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELFITPOLICY=*policy-value*
specifies a policy for avoiding collisions among the bar labels, when displayed.

SEGLABEL
displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*

specifies the text format used to display the bar segment labels.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

STATLABEL

NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels.

Limit options

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITS=BOTH | LOWER | UPPER
specifies which limit lines to display.

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR.

Plot options

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend.

MISSING
accepts a missing value as a valid category value.

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

WEIGHT=*numeric-variable*
specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Statistics options

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
: specifies the statistic to use for computing the response colors when the COLORRESPONSE= option is specified.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
: specifies the statistic for the vertical axis.

Required Argument

category-variable
: specifies the variable whose values determine the categories of data represented by the bars. The variable generates the midpoints to which each observation in the data set contributes.

  Interval bar charts are supported when the category axis is set to TYPE=LINEAR.

Optional Arguments

ALPHA=*numeric-value*
: specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

  Default .05

  Interactions This option has no effect if you do not specify LIMITSTAT=CLM.

  If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

  See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

  “Overview of Attribute Maps” on page 1897

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

  For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

  If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

  Defaults .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

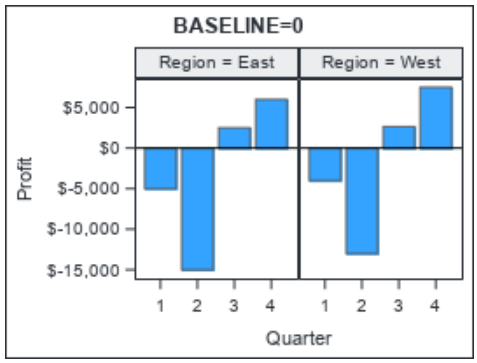
Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*

specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interactions If GROUPDISPLAY=STACK is specified, this option is ignored.

When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CATEGORYORDER=RESPASC | RESPDESC

specifies the order in which the categories are arranged. By default, the plot is sorted in ascending order based on the category values. You can change the order so that it is based on the response values.

Specify one of the following values:

RESPASC
sorts by the response values in ascending order.

RESPDESC
sorts by the response values in descending order.

Default The plot is sorted in ascending order based on the category values.

Restriction This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Requirement This option requires that you configure the panel to use either one column or one row, depending on the orientation of your charts. Use the ROWS= option or the COLUMNS= option in the PANELBY statement. If you do not use this option and your graph contains multiple cells, the specified sort order is not correctly applied to all cells.

Interactions When a group variable is used with the CATEGORYORDER= option, the category order is not affected by the value of the groups. The categories are always sorted by the response statistic at a category level.

When this option and the GROUPORDER= option are both specified, the GROUPORDER= option is ignored.

Notes If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

When two or more observations have the same response values, there is no guarantee that the tied values will be sub-sorted.

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified or COLORSTAT= must be FREQ or PCT.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors when the COLORRESPONSE= option is specified. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

SUM
sum values for the color response

MEAN
mean values for the color response

MEDIAN
median values for the color response

Default SUM when option COLORRESPONSE= is also specified.

Requirement The COLORRESPONSE= option must also be specified. If COLORRESPONSE= is not specified, COLORSTAT= is ignored.

Note This option is independent of the STAT= and RESPONSE= options.

DATALABEL
DATALABEL=*variable*
displays a label for each bar.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITS or LIMITSTAT, then the limit labels are also shown. Limit labels are shown when you use DATALABEL by itself without a variable.

This option is supported with the GROUP= option only when GROUPDISPLAY=CLUSTER is specified. Otherwise, the label is not drawn. The labels are shown if you also specify the DATALABELPOS= option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=*policy-value*
specifies a policy for avoiding collisions among the bar labels, when displayed. Select one of the following values:

NONE
does not rotate the bar labels.

Labels that are too long overlap.

ROTATE
rotates the text 90 degrees, but only if collisions occur.

SPLIT
splits the labels at the character or characters specified in the SPLITCHAR= option.

No split occurs at split characters where a split is not needed. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Tips Use the SPLITCHAR= option to specify a split character.

The split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

SPLITALWAYS
always splits the values at the character or characters specified in the SPLITCHAR= option.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Tips Use the SPLITCHAR= option to specify a split character.

The split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

Defaults ROTATE if the column is numeric.

SPLIT if the column is character.

Interaction This option has no effect unless DATALABEL= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 8.42 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS

MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.

Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL
specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note

This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips

Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)

specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element

specifies the name of a style element.

options

specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

COLOR=*color*

specifies a color to use for the fill lines.

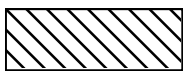
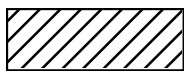
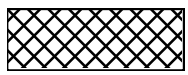
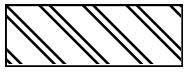
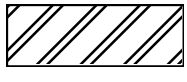
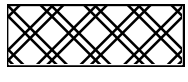
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

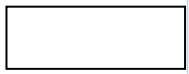
PATTERN=*line-pattern*

specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.43 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interactions If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

When FREQ= and MISSING are both specified, the MISSING option has no effect.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, “Using Attribute Maps to Control Visual Attributes,” on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP=is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default ASCENDING

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

INTERVALBARWIDTH=*numeric-value<units>*
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is "Frequency".

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITS= or LIMITSTAT= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITS=BOTH | LOWER | UPPER

specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending from each bar. Upper limits extend to the right of the bar and lower limits extend to the left of the bar. By default, no limits are displayed unless you specify either the LIMITS= option or LIMITSTAT= option. If you specify the LIMITSTAT= option only, then LIMITS=BOTH is the default. Specify one of the following values:

BOTH
adds lower and upper limit lines to the plot.

LOWER
adds lower limit lines to the plot.

UPPER
adds upper limit lines to the plot.

Default By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interactions Limit lines are displayed only when you specify STAT=MEAN.

If you use the GROUP= option in the plot statement, the LIMITS= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITSTAT=CLM | STDDEV | STDERR

specifies the statistic for the limit lines. Specify one of the following statistics:

CLM
confidence limits

STDDEV
standard deviation

STDERR
standard error

Default If you specify the LIMITS= option, then the default for LIMITSTAT= is CLM. Otherwise, there is no default.

Interactions If you specify the LIMITSTAT= option only, then the default value for the LIMITS= option is BOTH.

Limits lines are displayed only when you specify STAT=MEAN.

If you use the GROUP= option in the plot statement, the LIMITSTAT= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

Tip To display limit labels, specify DATALABEL (without a variable).

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

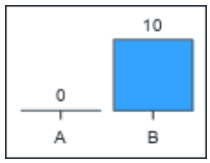
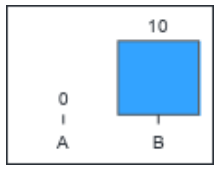
Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS
suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR. You can specify any positive number, including decimals.

Default 1

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphOutlines style element
- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

This option is ignored if the DATASKIN= option is also specified.

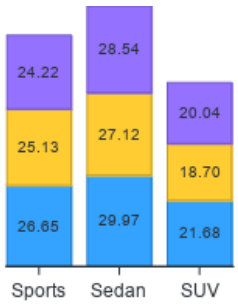
RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the vertical axis.

SEGLABEL
displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element in the current style.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

NONE
no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP
does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN
drops any bar segment label that does not fit within its segment.

The label width must not exceed the bar width, and the text height must not exceed the segment height.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the vertical axis. Specify one of the following:

FREQ
the frequencies of the response or category variable.

The frequencies are calculated as follows:

- If you specify the RESPONSE= option, FREQ calculates the frequency of the response variable.
- If you do not specify the RESPONSE= option, FREQ calculates the frequency of the category variable.

MEAN
the mean of the response variable.

Interaction For STAT=MEAN to take effect, you must also specify the RESPONSE= option.

MEDIAN
the median of the response variable.

Interaction For STAT=MEDIAN to take effect, you must also specify the RESPONSE= option.

PERCENT
the percentage of the response or category variable.

The percentage is calculated as follows:

- If you specify the RESPONSE= option, PERCENT calculates the percentage of the sum of the response variable.

- If you do not specify the RESPONSE= option, PERCENT calculates the percentage of the frequency of the category variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the PCTLEVEL= option in the PROC SGPANEL statement.

You can use the PCTNDEC= option in the SGPANEL procedure statement to control the number of decimals to be used when calculating the percent values.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level will be zero.

SUM
the sum of the response variable.

This is the default value when you specify the RESPONSE= option.

Interaction For this value to take effect, you must also specify the RESPONSE= option.

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Restriction If you do not also specify the RESPONSE= option, then only the FREQ or PERCENT statistic is calculated (FREQ is the default). If you specify RESPONSE=, then you can use any of the statistics.

Interaction When the graph is generated, the statistic is appended to the variable name in the axis label and the legend (if it is created). However, if a label has been assigned to the variable, then the label appears in the axis label and legend instead of the statistic.

STATLABEL | NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels. STATLABEL forces the statistic to be displayed. NOSTATLABEL removes the statistic from the axis and legend labels.

Normally, the procedure displays the statistic along with the name of the response variable. However, when a custom label is assigned to the response variable, the procedure does not display the statistic. In each case, you can control whether the statistic is displayed.

Defaults The statistic is displayed for the response variable.

When a custom label is assigned to the response variable, the statistic is not displayed.

Interactions This option has no effect unless the RESPONSE= option is specified.

This option has no effect if you specify the axis label using the LABEL= option in an AXIS statement.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces the information that is displayed by default. However, if RESPONSE= is not specified and TIP= is other than NONE, the computed frequency is added to the specified tips.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

If your categorical chart is overlaid with other categorical charts, each chart can specify its own URL variable. The URL variable specified for each chart overlay is honored. This capability affects the following charts: DOT, HBAR, HLINE, VBAR, VLINE.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction If your plot is overlaid with other categorization plots that also specify WEIGHT=, then the first WEIGHT variable that you specified is used for all of the plots.

VBARBASIC Statement

Creates a vertical bar chart that is compatible with other categorization charts as well as basic plots, such as scatter and series plots, and box plots.

Notes: When using the VBARBASIC statement, axes are not guaranteed to be uniform across BY groups.

The bars are sorted in data order based on the category values. The sort order cannot be changed. This arrangement of the bars differs from the VBAR statement, which by default sorts bars in alphabetical order based on the category values.

Tip: The VBARBASIC statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Bar Charts" on page 53

Syntax

VBARBASIC *category-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
: specifies whether the baseline is displayed.

FILL

NOFILL
: specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLENDCOLOR=*color*
: specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
: specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
: specifies the thickness of the bars when the category data is numeric.

NOZEROBARS
: suppresses zero-length bars.

OUTLINE

NOOUTLINE
: specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

Data tip options

TIP=(*role-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL
displays the bar statistic value for each bar.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL option.

DATALABELFITPOLICY=NONE
specifies that no fit policy is implemented for the bar labels.

DATALABELFORMAT=*format*
specifies the text format used to display the bar label.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend.

SEGLABEL
displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN

specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Plot options

MISSING
accepts a missing value as a valid category value.

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Statistics options

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors.

STAT=FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic for the vertical axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the bars. The variable generates the midpoints to which each observation in the data set contributes.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Defaults .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

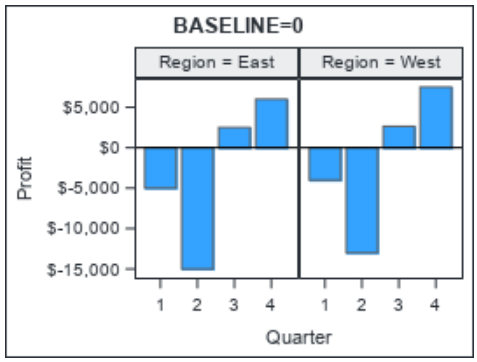
Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*

specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interaction When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style

element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CLUSTERWIDTH=*numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)

specifies a color ramp that is to be used to determine response colors.

style-element

specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example colormodel=TwoColorRamp

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example colormodel=(blue yellow green)

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified or COLORSTAT= must be FREQ or PCT.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Interaction This option is ignored if COLORSTAT=FREQ or COLORSTAT=PCT.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

When COLORRESPONSE= is specified, the following values are valid:

SUM
sum values for the color response

MEAN
mean values for the color response

MEDIAN
median values for the color response

Default SUM when option COLORRESPONSE= is also specified.

Requirement The COLORRESPONSE= option must also be specified. If COLORRESPONSE= is not specified, COLORSTAT= is ignored.

Note This option is independent of the STAT= and RESPONSE= options.

DATALABEL
displays the bar statistic value for each bar. For grouped clustered bars, each bar is labeled with the summarized value of the bar. For grouped stacked bars, the segmented bar is labeled with the accumulated, summarized value of all the bar segments.

Default No label is shown

Tip The font and color attributes for the label are specified by the DATALABELATTRS= option. The text format is specified by the DATALABELFORMAT= option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=NONE
specifies that no fit policy is implemented for the bar labels. By default, if the data labels collide, the fit policy is to rotate the data labels 90 degrees, and if they still collide, drop all of the data labels. As a result, the labels might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

Default Rotate the labels 90 degrees, and if they still collide, drop all of the labels.

Interaction This option has no effect unless DATALABEL is also specified.

DATALABELFORMAT=*format*
specifies the text format used to display the bar label.

Default The column format assigned to the RESPONSE= column, or BEST6 if no format is assigned.

Interaction This option has no effect unless DATALABEL= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 8.44 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See “Using Data Skins” on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.

Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL
specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interaction This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

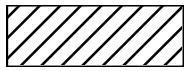
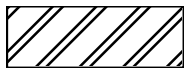
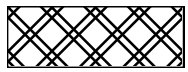
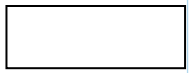
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.45 Fill Patterns

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, “Using Attribute Maps to Control Visual Attributes,” on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP=is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use

default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INTERVALBARWIDTH=*numeric-value<units>*
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is "Frequency".

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

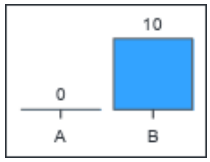
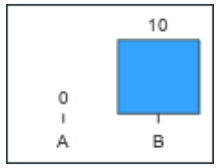
MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS
suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see “Line Attributes and Patterns” on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphOutlines style element
- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the

ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

This option is ignored if the DATASKIN= option is also specified.

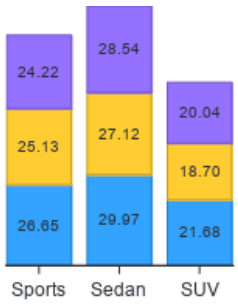
RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed for each value on the horizontal axis.

SEGLABEL
displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element in the current style.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

NONE
no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP
does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN
drops any bar segment label that does not fit within its segment.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

STAT=FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic for the vertical axis.

For bar charts with no RESPONSE= variable, the following values are valid:

FREQ
frequency count

PERCENT
percentages between 0 and 100

Alias PCT

PROPORTION
proportions between 0 and 1

For bar charts with a RESPONSE= variable, the following values are valid:

SUM
sum values for the response

MEAN
mean values for the response

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Interaction When this option is used with the GROUP=group option, the specified statistic is computed for each segment that is created for the unique group values.

TIP=(*role-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*role-list*)
a space-separated list of unique chart roles enclosed in parentheses.

The available roles for TIP are CATEGORY, GROUP, and RESPONSE. Specify one or more of these role names in *role-list*. Data tips are displayed using the data obtained from the specified roles.

Note: You must specify the GROUP and RESPONSE roles for the chart in order to use those roles for data tips.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP=ON;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(category response)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

VBARPARM Statement

Creates a vertical bar chart based on a pre-summarized response value for each unique value of the category variable. You can also assign variables to the upper and lower limits.

Requirements: Parameterized bar charts require a response variable in addition to the category variable. The response variable contains pre-summarized computed values such as a sum or a mean for each unique value of the category variable. This is the key distinction between VBARPARM and VBAR.

The data must contain only one response value per unique category variable. If more than one value is found, a warning is written to the SAS log, and the graph might produce unpredictable results.

Interaction: The VBARPARM statement can be combined only with other basic plot statements in the SGPANEL procedure. See "Plot Compatibility" on page 1780.

Note: The bars are sorted in data order based on the category values. The sort order cannot be changed. This arrangement of the bars differs from the VBAR statement, which by default sorts bars in alphabetical order based on the category values.

Tip: The VBARPARM statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

Example: "About Bar Charts" on page 53

Syntax

VBARPARM CATEGORY=*category-variable* RESPONSE=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
: specifies whether the baseline is displayed.

FILL

NOFILL
: specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLENDCOLOR=*color*
: specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
: specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
: specifies the thickness of the bars when the category data is numeric.

NOZEROBARS
: suppresses zero-length bars.

OUTLINE

NOOUTLINE
specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELFITPOLICY=*policy-value*
specifies a policy for avoiding collisions among the bar labels, when displayed.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend.

SEGLABEL
displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Limit options

LIMITATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITLOWER=*numeric-variable*
specifies values for the lower endpoints on the limit lines.

LIMITUPPER=*numeric-variable*
specifies values for the upper endpoints on the limit lines.

Plot options

MISSING
accepts a missing value as a valid category value.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

CATEGORY=*category-variable*
specifies the variable that categorizes the data. All values are treated as discrete values. The input data for this variable should contain unique values. When the category values are not unique, a warning is logged, and multiple bars are superimposed at the duplicated category values.

The CATEGORY axis is always discrete.

RESPONSE=*numeric-variable*
specifies a numeric response variable. The input data is expected to be pre-summarized computed values (sum, mean, and so on).

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Defaults .8

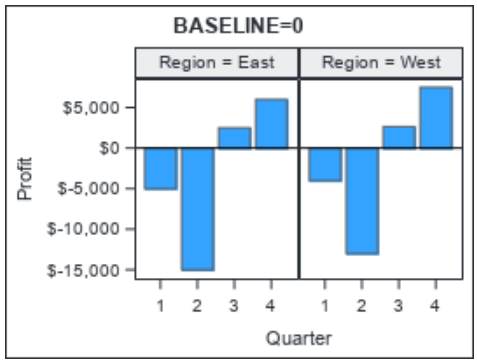
1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interaction When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CLUSTERWIDTH=*numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

DATALABEL
DATALABEL=*variable*
displays a label for each data point.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

This option is supported with the GROUP= option only when GROUPDISPLAY=CLUSTER is specified. Otherwise, the label is not drawn. The labels are shown if you also specify the DATALABELPOS= option.

If you also specify LIMITLOWER or LIMITUPPER, then the limit labels are also shown. Limit labels are shown when you use DATALABEL (without a variable).

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=*policy-value*
specifies a policy for avoiding collisions among the bar labels, when displayed. Select one of the following values:

NONE
does not rotate the bar labels.

Labels that are too long overlap.

ROTATE
rotates the text 90 degrees, but only if collisions occur.

SPLIT
splits the labels at the character or characters specified in the SPLITCHAR= option.

No split occurs at split characters where a split is not needed. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Tips Use the SPLITCHAR= option to specify a split character.

The split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

SPLITALWAYS
always splits the values at the character or characters specified in the SPLITCHAR= option.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Tips Use the SPLITCHAR= option to specify a split character.

The split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

Defaults ROTATE if the column is numeric.

SPLIT if the column is character.

Interaction This option has no effect unless DATALABEL= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 8.46 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.

Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL
specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

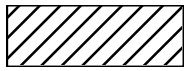
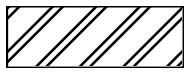
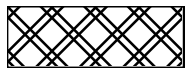
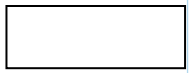
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.47 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions

For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note

This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, “Using Attribute Maps to Control Visual Attributes,” on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes

derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP=is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted

values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

INTERVALBARWIDTH=*numeric-value<units>*
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITLOWER= or LIMITUPPER= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITLOWER=*numeric-variable*
specifies values for the lower endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default The lower segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions If LIMITUPPER= is also specified, then the plot displays the lower and upper segments of the limit lines.

If you use the GROUP= option in the plot statement, the LIMITLOWER= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITUPPER=*numeric-variable*
specifies values for the upper endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default The upper segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions If LIMITLOWER= is also specified, then the plot displays the lower and upper segments of the limit lines.

If you use the GROUP= option in the plot statement, the LIMITUPPER= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

Tip To display limit labels, specify DATALABEL (without a variable).

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

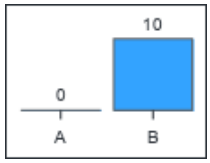
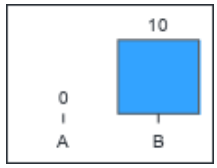
NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS
suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a

simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphOutlines style element

- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

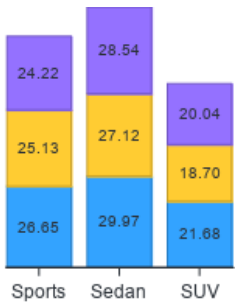
This option is ignored if the DATASKIN= option is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SEGLABEL
displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you

specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element in the current style.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

NONE
no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP
does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN
drops any bar segment label that does not fit within its segment.

The label width must not exceed the bar width, and the text height must not exceed the segment height.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Notes When TIP= is not specified, the CATEGORY, RESPONSE, COLORRESPONSE, and GROUP variables are displayed, if applicable, when you hover over a bar. The ERRORUPPER or ERRORLOWER variable is displayed when you hover over an error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a bar. No variables are displayed when you hover over an error bar or error bar cap.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see “Using the ODS GRAPHICS Statement” on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

VBOX Statement

Creates a vertical box plot that shows the distribution of your data.

Restriction: This plot has compatibility restrictions when used with other plots. See Table 13.2 on page 1781.

See: "Visual Description of Box Plot Percentile Boundaries" on page 721

Example: "About Box Plots" on page 47

Syntax

VBOX *numeric-analysis-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BOXWIDTH=*numeric-value*
: specifies the width of the box.

CAPSCALE=*numeric-value*
: specifies a positive number to use as a multiplier for changing the default width of the whisker caps.

CAPSHAPE=BRACKET | LINE | SERIF | NONE
: specifies the shape of the whisker cap lines.

CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
: specifies that a connect line joins a statistic from box to box.

CONNECTATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the lines that connect multiple boxes.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all boxes from the discrete tick marks.

EXTREME
: specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified.

FILL

NOFILL
: specifies whether the boxes are filled with color.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLPATTERN
enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern.

INTERVALBOXWIDTH=*numeric-value*<*units*>
specifies the box width when an interval category (X) variable is specified.

LINEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the box outlines.

MEANATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the marker that represents the mean in the box.

MEDIANATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the line that represents the median.

NOCAPS
hides the cap lines for the whiskers.

NOMEAN
hides the mean marker.

NOMEDIAN
hides the median line.

NOOUTLIERS
hides the outliers from the plot.

NOTCHES
specifies that the boxes be notched.

OUTLIERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the marker that represents the outliers.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

WHISKERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the whisker and cap lines.

WHISKERPCT=*number*
specifies the whisker length, in percentile units.

Data tip options

TIP=(*role-list*) | NONE
specifies the information to display when the cursor is positioned over a box or whisker in the box plot.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the midpoint spacing.

GROUP=*variable*

specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped boxes.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
adds data labels for the outlier markers.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

LABELFAR
specifies that only the far outliers have data labels.

LEGENDLABEL="*text-string*"
specifies a label that identifies the box plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

CATEGORY=*category-variable*
specifies the category variable for the plot.

DISPLAYSTATS=ALL | (*statistics-list*)
specifies statistics to be displayed for each box.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

MISSING
accepts a missing value as a valid category value.

PERCENTILE=1 | 2 | 3 | 4 | 5
specifies a method for computing the percentiles for the plot.

SPREAD
relocates outlier points that have identical values to prevent overlapping.

WEIGHT=*numeric-variable*
specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Argument

numeric-analysis-variable
specifies the analysis variable for the plot. If you do not specify the CATEGORY= option, then one box is created for the analysis variable.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BOXWIDTH=*numeric-value*
specifies the width of the box. Specify a value between 0.0 (0% of the available width) and 1.0 (100% of the available width).

Defaults 0.4

When GROUP is specified, the default box width is 0.6.

CAPSCALE=*numeric-value*
specifies a positive number to use as a multiplier for changing the default width of the whisker caps. For example, CAPSCALE=2 doubles the default cap width and CAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction This option has no effect if NOCAPS is specified or if CAPSHAPE=NONE.

Tips The thickness (height) of a SERIF, LINE, or BRACKET cap honors the line thickness specified in the WHISKERATTRS= option. The cap pattern is always solid.

Use the CAPSHAPE= option to specify the line shape of the caps.

CAPSHAPE=BRACKET | LINE | SERIF | NONE
specifies the shape of the whisker cap lines. Specify one of the following values:

BRACKET
displays a straight line with brackets.

LINE
displays a straight line.

SERIF
displays a short straight line.

NONE
does not display a cap.

Default SERIF

CATEGORY=*category-variable*
specifies the category variable for the plot. A box plot is created for each distinct value of the category variable.

If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete.

The default ordering of the tick values is ascending formatted order of the *category-variable*. To change this ordering, you can use the DISCRETEORDER=DATA option in the category axis statement.

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the midpoint spacing. Specify a value from 0.1 (narrowest) to 1.0 (widest).

Default 0.7

Interaction This option is applicable only when a GROUP is in effect and the category axis is discrete.

CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
specifies that a connect line joins a statistic from box to box.

Interaction This option applies only when the CATEGORY option is used to generate multiple boxes.

Tip You can use the CONNECTATTRS option to specify attributes for the connect line.

CONNECTATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the lines that connect multiple boxes. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphConnectLine style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interactions This option takes effect only if the CONNECT= option is also specified.

This option is ignored if the GROUP= option is also specified.

Examples
```
CONNECTATTRS=(Color="light green" Pattern=MediumDash Thickness=4)
```

This example specifies a style element:
```
CONNECTATTRS=GraphData3
```

DATALABEL
DATALABEL=*variable*
adds data labels for the outlier markers. If you specified a variable, then the values for that variable are used for the data labels. If you did not specify a variable, then the values of the analysis variable are used.

Note This option has no effect unless the plot contains outlier points.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

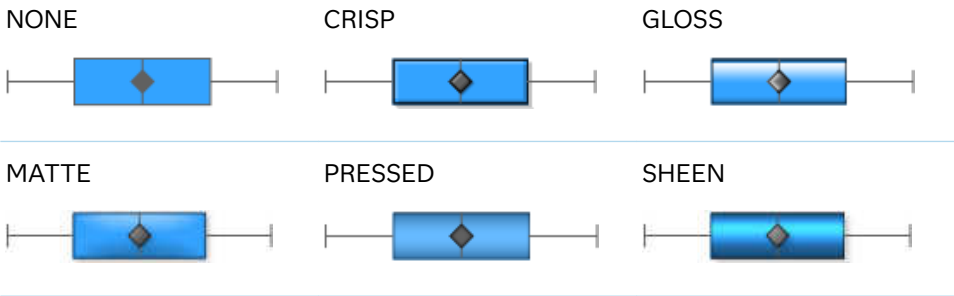
Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled boxes. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 8.48 *DATASKIN Options for Box Plots*



Though the figure shows horizontal boxes, the data skin for vertical boxes is identical.

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interaction If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*

specifies an amount to offset all boxes from the discrete tick marks.

Specify a value from -0.5 (left offset) to +0.5 (right offset). If you specify a value outside of this range, an error message appears in the SAS log and the graph is not produced.

Default 0.0 (no offset)

DISPLAYSTATS=ALL | (*statistics-list*)

specifies statistics to be displayed for each box.

ALL

displays all available statistics.

See the *statistics-list*.

statistics-list

specifies a space-separated list of one or more of the following statistics, enclosed in parentheses:

DATAMAX	maximum data value that includes not only the maximum whisker values but also the maximum outlier values. The DATAMAX value is greater than or equal to the MAX value and is always represented in the axis range.
DATAMIN	minimum data value that includes not only the minimum whisker values but also the minimum outlier values. The DATAMIN value is less than or equal to the MIN value and is always represented in the axis range.
IQR	interquartile range (Q3–Q1).
MAX	maximum data value below the box upper fence.
MEAN	mean data value for the box.
MEDIAN	median data value for the box.
MIN	minimum data value above the box lower fence.
N	number of observations for the box.
Q1	lower quartile (25th percentile) for the box.

Q3	upper quartile (75th percentile) for the box.
RANGE	range of the data (MAX–MIN).
STD	standard deviation of the data for the box.
SUMWGT	sum of the weights for the box.

Example `displaystats=(mean median)`

Restriction This option is ignored if the GROUP= option is also specified.

EXTREME
specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified. When you do not specify the EXTREME option, the whiskers cannot be longer than 1.5 times the length of the box.

FILL | NOFILL
specifies whether the boxes are filled with color. The FILL option shows the fill color. The NOFILL option hides the fill color.

Default FILL

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note
: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips
: Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

: Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more fill-pattern attribute options.

Specify one or both of the following options:

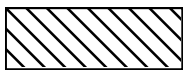
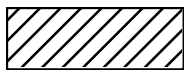
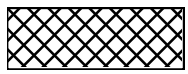
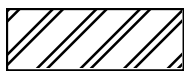
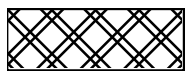
COLOR=*color*
specifies a color to use for the fill lines.

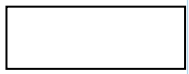
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 8.49 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FREQ=*numeric-variable*

specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interaction If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP=*variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped boxes.

CLUSTER
the boxes are drawn adjacent to each other.

OVERLAY
all the boxes for a given group value are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Defaults CLUSTER for a discrete category axis

OVERLAY for a linear axis

Interaction This option is ignored unless GROUP=is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INTERVALBOXWIDTH=*numeric-value*<*units*>
specifies the box width when an interval category (X) variable is specified. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Alias INTBOXWIDTH=

Restriction The axis type for the category axis must be LINEAR, and the variable must be numeric.

Note INTERVALBOXWIDTH= is valid starting with 2024.05. In 2024.04 and prior releases, use INTBOXWIDTH= instead.

Example
```
proc sgpanel data=sashelp.class;
  panelby sex;
  vbox weight / category=height intboxwidth=50;
  colaxis type=linear;
run;
```

LABELFAR
specifies that only the far outliers have data labels. Far outliers are points whose distance from the box is more than three times the length of the box.

Note This option has no effect if you do not specify the DATALABEL option, or if there are no far outliers.

LEGENDLABEL="*text-string*"
specifies a label that identifies the box plot in the legend. By default, the label of the analysis variable is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the box outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

Interaction This option is ignored if the DATASKIN= option is also specified.

MEANATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker that represents the mean in the box. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphBoxMean style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

MEDIANATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the line that represents the median. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphBoxMedian style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interaction This option is ignored if the NOMEDIAN option is also specified.

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOCAPS
hides the cap lines for the whiskers.

Interaction Using several options that hide box features can cause the NOCAPS option to be ignored. For example, if you use NOCAPS, NOFILL, NOMEAN, NOMEDIAN, and NOOOUTLIERS in the same statement, the NOCAPS option might be ignored.

NOMEAN
hides the mean marker.

NOMEDIAN
hides the median line.

NOOUTLIERS
hides the outliers from the plot.

NOTCHES
specifies that the boxes be notched. See "Notches" on page 722.

OUTLIERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker that represents the outliers. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphOutlier style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

PERCENTILE=1 | 2 | 3 | 4 | 5
specifies a method for computing the percentiles for the plot.

Default 5 (empirical distribution function with averaging)

See "Calculating Percentiles" on page 723

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPREAD
relocates outlier points that have identical values to prevent overlapping.

Note This option has no effect if your data does not contain two or more outliers with identical values for the analysis variable.

TIP=(*role-list*) | NONE
specifies the information to display when the cursor is positioned over a box or whisker in the box plot.

(*role-list*)
a space-separated list of unique box plot roles enclosed in parentheses.

The box plot roles for TIP are CATEGORY, N, STD, MIN, MAX, MEAN, MEDIAN, Q1, and Q3. Specify one or more of these role names in *role-list*. Data tips are displayed using the data obtained from the specified roles.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(mean median)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

WHISKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the whisker and cap lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphBoxWhisker style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

WHISKERPCT=*number*
specifies the whisker length, in percentile units. When this option is specified, *number* is used as the low percentile, and 100–*number* is used as the high percentile.

Here are some examples of values and their effect:

0 specifies the high and low extremes

10 specifies the 10th percentile low and the 90th percentile high

25 specifies the 25th percentile low and the 75th percentile high

Default The whiskers are drawn from the box to the most extreme point that is less than or equal to 1.5 times the IQR

Range 0–25

Notes When this option is specified, fences and far outliers are not drawn.

When this option is set to 25, no whiskers are drawn because the box extends from the 25th to the 75th percentile.

Details

Statement Summary

The plot displays a single box if only the analysis variable is provided. The plot displays multiple boxes if a category variable is also provided and that variable has more than one unique value.

The ANALYSIS variable is displayed on the vertical axis. The axis for the analysis column is always LINEAR.

By default for numeric or character columns, the CATEGORY= axis is TYPE=DISCRETE. You can override the default and set TYPE=LINEAR in the axis statement, provided that the category column is numeric.

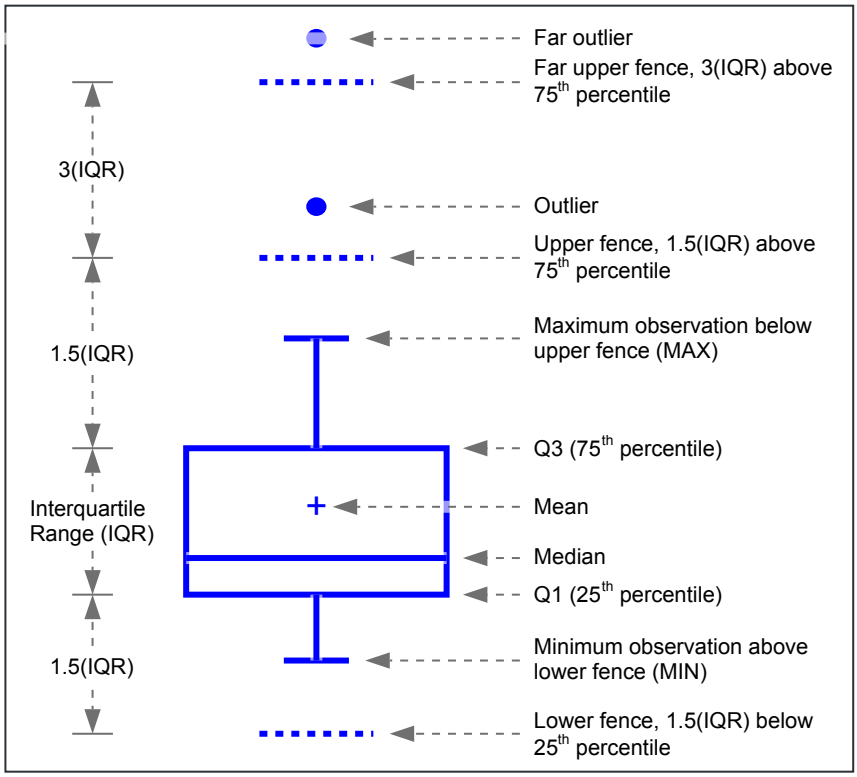
If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete. For the interval case, you can use the INTBOXWIDTH= option to specify the box width.

Two basic box plot representations can be drawn: a schematic (Tukey) box plot and a skeletal box plot. See the EXTREME option for details.

Visual Description of Box Plot Percentile Boundaries

Box plots display the distribution of data by using a rectangular box and whiskers. Whiskers are lines that indicate a data range outside of the box.

Figure 8.11 *Parts of a Box Plot*



In the previous figure, the bottom and top edges of the box indicate the intra-quartile range (IQR). That is the range of values between the first and third quartiles (the 25th and 75th percentiles). The marker inside the box indicates the mean value. The line inside the box indicates the median value.

The elements that are outside the box are dependent on your options. By default, the whiskers that extend from each box indicate the range of values that are outside of the intra-quartile range. However, they are close enough not to be considered outliers (a distance less than or equal to 1.5*IQR). If you specify the EXTREME option, then the whiskers indicate the entire range of values, including outliers.

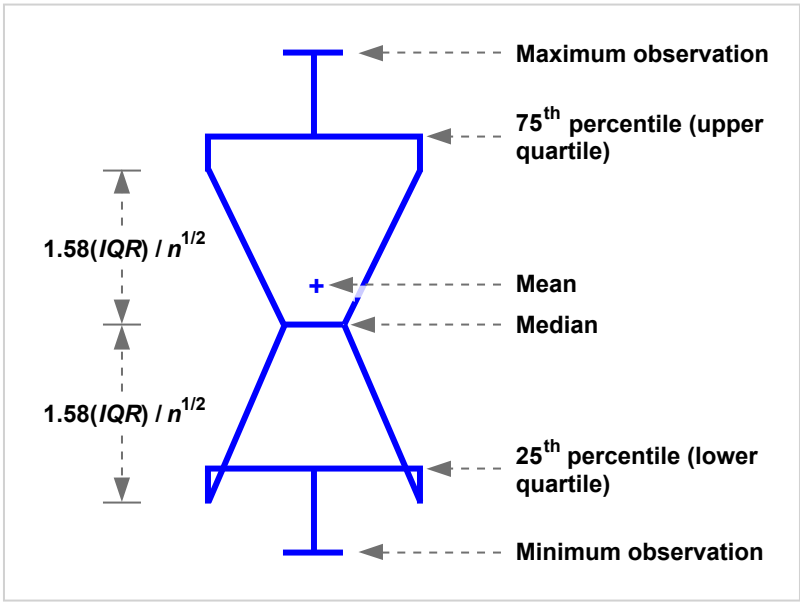
Outliers are observations that are more extreme than the upper and lower fences ($\pm$ 1.5 IQR). Outliers that are beyond upper and lower far fences ($\pm$ 3 IQR) are called FAR OUTLIERS. By default, outliers are indicated by markers. If you specify the DATALABEL= option, then the outlier points have data labels. If you also specify the LABELFAR option, then only outliers that are 3*IQR from the box have data labels.

Notches

In a notched box plot, the endpoints of the notches are at the following computed locations:

$$median \pm 1.58\,(IQR/\sqrt{N})$$

In the equation, IQR (IQR=Q3-Q1) is the interquartile range and *n* is the sample size. The medians (central lines) of the two boxes are significantly different at approximately the 0.05 level if the corresponding notches do not overlap.



Note: In some cases, the notches in the box plot can extend beyond DATAMIN and DATAMAX.

Calculating Percentiles

You can specify one of five definitions for computing the percentiles with the PERCENTILE= option. Let n be the number of nonmissing values for a variable, and let $X_1, X_2, \ldots, X_n$ represent the ordered values of the variable. X_1 is the smallest value, X_2 is the next smallest, and X_n is the largest value. Let the tth percentile be y, set:

$$p = \frac{t}{100}$$

and let:

$$np = j + g$$

when PERCENTILE=1, 2, 3, or 5, or let:

$$(n+1)p = j + g$$

when PERCENTILE=4, where j is the integer part of np, and g is the fractional part of np. Then the PERCENTILE= option defines the tth percentile, y, as described in the following table:

Percentile	Description	Equation	Notes
1	Weighted average at X_{np}	$y = (1-g)x_j + gx_{j+1}$	x_0 is taken to be x_1
2	Observation numbered closest to *np*	$y = x_j$	Used when $g < \frac{1}{2}$ or when $g = \frac{1}{2}$ and *j* is even
		$y = x_{j+1}$	Used when $g = \frac{1}{2}$ and *j* is odd or when $g > \frac{1}{2}$
3	Empirical distribution function	$y = x_j$	Used when $g = 0$
		$y = x_{j+1}$	Used when $g > 0$
4	Weighted average aimed at $x_{(n+1)p}$	$y = (1-g)x_j + gx_{j+1}$	x_{n+1} is taken to be x_n
5	Empirical distribution function with averaging	$y = \frac{1}{2}(x_j + x_{j+1})$	Used when $g = 0$
		$y = x_{j+1}$	Used when $g > 0$

VECTOR Statement

Creates a vector plot that draws arrows from a point of origin to each data point.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Vector Plots" on page 39

Syntax

VECTOR X=*numeric-variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ARROWDIRECTION=IN | OUT | BOTH
: specifies the location of the arrowheads for the vectors.

ARROWHEADSHAPE=*shape*
: specifies the shape of the arrowheads for the vectors.

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

LINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the vector line.

NOARROWHEADS
: removes the arrowheads from the vectors.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

THICKMAX=*numeric-value*<*units*>
: specifies the maximum line thickness when a response variable is used to determine the line thickness.

THICKMAXRESP=*numeric-value*
: specifies the response value that corresponds to the maximum line thickness.

THICKRESP=*numeric-variable*

specifies a response variable that is used to map a line thickness to each group value.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot.

LEGENDLABEL="*text-string*"
specifies a label that identifies the vector plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

XORIGIN=*numeric-value* | *numeric-variable*
specifies the X coordinate of the origin for the vectors.

YORIGIN=*numeric-value* | *numeric-variable*
specifies the Y coordinate of the origin for the vectors.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Required Arguments

X=*numeric-variable*
specifies a numeric variable for the X axis.

Y=*numeric-variable*
specifies numeric variable for the Y axis.

Optional Arguments

ARROWDIRECTION=IN | OUT | BOTH
specifies the location of the arrowheads for the vectors. Specify one of the following:

IN
places the arrowheads at the origin of the vector.

OUT
places the arrowheads at the ending point of the vector.

BOTH
places arrowheads at both the origin and ending point of the vector.

Default OUT

ARROWHEADSHAPE=*shape*
specifies the shape of the arrowheads for the vectors. Specify one of the following:

OPEN
resembles the letter "V".

Example:

CLOSED
an outline of a triangle.

Example:

FILLED
a solid triangle.

Example:

BARBED
a solid triangle with an indent at the base.

Example:

Default OPEN

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(color-list)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorAltRamp style element

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 272

"Using Gradient Color Legends" on page 1793

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 8.50 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

GROUP=*variable*
specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

LEGENDLABEL="*text-string*"
specifies a label that identifies the vector plot in the legend. By default, the label of the Y variable or the group value for each plot element is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the vector line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOARROWHEADS
removes the arrowheads from the vectors.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THICKMAX=*numeric-value<units>*
specifies the maximum line thickness when a response variable is used to determine the line thickness. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

By default, this option determines the thickness of the line that represents the maximum response column value.

Default Ten times the thickness that is specified by the GraphDataDefault style element LineThickness attribute.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The THICKMAXRESP= option specifies the response value at which this maximum line thickness is reached. The line thickness for response values that exceed the THICKMAXRESP= value are set to the value that is specified by this option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

THICKMAXRESP=*numeric-value*
specifies the response value that corresponds to the maximum line thickness.

Default The maximum value in the response column that is specified in the THICKRESP= option.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The thickness for all lines that exceed the maximum response value is set to the value specified in the THICKMAX= option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

THICKRESP=*numeric-variable*
specifies a response variable that is used to map a line thickness to each group value.

Default The GraphDataDefault style element LineThickness attribute.

Restriction The THICKRESP= values are assumed to be constant for each group value. If the THICKRESP column has multiple values for a single GROUP value, only one of the THICKRESP= values is used for that group.

Requirement The GROUP= option must be specified with the THICKRESP= option. Otherwise, the THICKRESP= option is ignored.

Interactions When the column values are all zero, all negative, or all missing, this option is ignored. In that case, the default line thickness is used for all of the lines.

The THICKNESS= suboption of the LINEATTRS= option overrides this option for the line thickness attribute.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

XORIGIN=*numeric-value* | *numeric-variable*
specifies the X coordinate of the origin for the vectors. You can specify either a numeric value or a numeric variable.

Default 0

YORIGIN=*numeric-value* | *numeric-variable*
specifies the Y coordinate of the origin for the vectors. You can specify either a numeric value or a numeric variable.

Default 0

VLINE Statement

Creates a vertical line chart (the line is horizontal). You can use the VLINE statement with the VBAR statement to create a bar-line chart.

Restrictions: The VLINE statement can be combined only with other categorization plot statements in the SGPANEL procedure. See "Plot Compatibility" on page 1780.

When category plots are overlaid, all of the plots must use the same category variable.

See: "Marker Fills and Outlines" on page 1805

Examples: "About Line Charts" on page 61

"Example 4: Creating a Panel of Line Charts" on page 836

Syntax

VLINE *category-variable* </*options*>;

Summary of Optional Arguments

Appearance options
ATTRID=*character-value*

specifies the value of the ID variable in a discrete attribute map data set.

BREAK
breaks the line whenever the computed statistic for a category value is missing.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from discrete category values.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the lines in the line plot.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELPOS=MIN | MAX | START | END
specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot.

LEGENDLABEL="*text-string*"
specifies the label that identifies the line plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

STATLABEL

NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels.

Limit options

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITS=BOTH | LOWER | UPPER
adds limit lines to the plot.

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines.

NOLIMITCAPS
suppresses the serif cap on limit lines, if limit lines are displayed.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

MARKERS
adds data point markers to the plot.

Plot options

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

MISSING
for group data, processes missing values as a valid category value and creates a line for it.

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

WEIGHT=*numeric-variable*
specifies how observations are weighted.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Statistics options

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the vertical axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the lines.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default .05

Interactions This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA value that you specify is used for all of the plots.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BREAK
breaks the line whenever the computed statistic for a category value is missing. For example, if a response variable is used and it has all missing values for a certain category value, the SUM or MEAN for this category value will be missing. By default in such cases, the response value for the previous category is joined to the response value for the next category value by a line segment. If BREAK is specified, this segment is not drawn.

Note The observation is excluded from the graph when there is a missing value for the FREQ variable.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged. By default, the plot is sorted in ascending order based on the category values. You can change the order so that it is based on the response values.

Specify one of the following values:

RESPASC
sorts by the response values in ascending order.

RESPDESC
sorts by the response values in descending order.

Default The plot is sorted in ascending order based on the category values.

Restriction This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Requirement This option requires that you configure the panel to use either one column or one row, depending on the orientation of your charts. Use the ROWS= option or the COLUMNS= option in the PANELBY statement. If you do not use this option and your graph contains multiple cells, the specified sort order is not correctly applied to all cells.

Interactions When a group variable is used with the CATEGORYORDER= option, the category order is not affected by the value of the groups. The categories are always sorted by the response statistic at a category level.

When this option and the GROUPORDER= option are both specified, the GROUPORDER= option is ignored.

Notes If CATEGORYORDER is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

When two or more observations have the same response values, there is no guarantee that the tied values will be sub-sorted.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interaction This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see “Text Attributes” on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELPOS=MIN | MAX | START | END

specifies the location of the curve label. Specify one of the following values:

MIN
places the label at the part of the curve closest to the minimum X-axis value.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

START
places the curve label at the first point on the curve.

END
places the curve label at the last point on the curve.

Default END

Interaction This option has no effect unless the CURVELABEL option is also specified.

DATALABEL
DATALABEL=*variable*

displays a label for each data point.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data

labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITS or LIMITSTAT, then the limit labels are also shown. Limit labels are shown when you use DATALABEL (without a variable).

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, for the label font, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style. For the label color, when DATALABELPOS=DATA, the ContrastColor attribute of a GraphData1–GraphData*n* style element. Otherwise, the Color attribute of the GraphDataText style element.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of the data label with respect to the plot. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 8.51 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	**PRESSED**	**SHEEN**

If you specify markers with the plot, then the data skin affects the markers as well.

Table 8.52 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	**PRESSED**	**SHEEN**

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See “Using Data Skins” on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from discrete category values. Specify a value from –0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the category axis is discrete.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the “Filled”

suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interactions This option has no effect unless MARKERS is also specified.

Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*

specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interactions If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

When FREQ= and MISSING are both specified, the MISSING option has no effect.

GROUP=*variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY

specifies how to display grouped lines.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interaction This option is ignored unless GROUP= is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default ASCENDING

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the line plot in the legend. By default, the label of the response variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) are used. If you do not specify a response variable, then the legend label is "Frequency".

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

For grouped data, the LineStyle and LineThickness attributes of the GraphError style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

LIMITS=BOTH | LOWER | UPPER
adds limit lines to the plot. Specify one of the following values:

BOTH
adds lower and upper limit lines to the plot.

LOWER
adds lower limit lines to the plot.

UPPER
adds upper limit lines to the plot.

Default By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interaction Limit lines are displayed only when you specify STAT= MEAN.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITS= or LIMITSTAT= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines. Specify one of the following statistics:

CLM
confidence limits

STDDEV
standard deviation

STDERR
standard error

Default If you specify the LIMITS= option, then the default for LIMITSTAT= is CLM. Otherwise, there is no default.

Interaction If you specify the LIMITSTAT= option, then the default value for the LIMITS= option is BOTH.

Tip To display limit labels, specify DATALABEL (without a variable).

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the lines in the line plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interactions This option has no effect unless you also specify the MARKERS option.

Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

Suboption COLOR= in option LINEATTRS= overrides the default marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default — For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement — Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction — This option overrides any color that is specified with the MARKERATTRS= option.

Tip — You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See — For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default — For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness.

Requirement — Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip — You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKERS
adds data point markers to the plot.

MISSING
for group data, processes missing values as a valid category value and creates a line for it. If more than one chart is specified in the procedure, the MISSING option affects the group calculations for all of the charts.

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOLIMITCAPS
suppresses the serif cap on limit lines, if limit lines are displayed.

Interaction The appearance of the limit lines is controlled by the LIMITATTRS= option.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR. You can specify any positive number, including decimals.

Default 1

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the vertical axis.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve and data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the vertical axis. Specify one of the following:

FREQ
the frequencies of the response or category variable.

The frequencies are calculated as follows:

- If you specify the RESPONSE= option, FREQ calculates the frequency of the response variable.
- If you do not specify the RESPONSE= option, FREQ calculates the frequency of the category variable.

MEAN
the mean of the response variable.

Interaction For STAT=MEAN to take effect, you must also specify the RESPONSE= option.

MEDIAN
the median of the response variable.

Interaction For STAT=MEDIAN to take effect, you must also specify the RESPONSE= option.

PERCENT
the percentage of the response or category variable.

The percentage is calculated as follows:

- If you specify the RESPONSE= option, PERCENT calculates the percentage of the sum of the response variable.
- If you do not specify the RESPONSE= option, PERCENT calculates the percentage of the frequency of the category variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the PCTLEVEL= option in the PROC SGPANEL statement.

You can use the PCTNDEC= option in the SGPANEL procedure statement to control the number of decimals to be used when calculating the percent values.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level will be zero.

SUM
the sum of the response variable.

This is the default value when you specify the RESPONSE= option.

Interaction For this value to take effect, you must also specify the RESPONSE= option.

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Restriction If you do not also specify the RESPONSE= option, then only the FREQ or PERCENT statistic is calculated (FREQ is the default). If you specify RESPONSE=, then you can use any of the statistics.

Interaction When the graph is generated, the statistic is appended to the variable name in the axis label and the legend (if it is created). However, if a label has been assigned to the variable, then the label appears in the axis label and legend instead of the statistic.

STATLABEL | NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels. STATLABEL forces the statistic to be displayed. NOSTATLABEL removes the statistic from the axis and legend labels.

Normally, the procedure displays the statistic along with the name of the response variable. However, when a custom label is assigned to the response variable, the procedure does not display the statistic. In each case, you can control whether the statistic is displayed.

Defaults The statistic is displayed for the response variable.

When a custom label is assigned to the response variable, the statistic is not displayed.

Interactions This option has no effect unless the RESPONSE= option is specified.

This option has no effect if you specify the axis label using the LABEL= option in an AXIS statement.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Notes When TIP= is not specified, the category, RESPONSE, and GROUP variables are displayed, if applicable, when you hover over a line or marker. The category and upper limit or lower limit variables are displayed when you hover over an error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a line or marker. No variables are displayed when you hover over an error bar or error bar cap.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

If your categorical chart is overlaid with other categorical charts, each chart can specify its own URL variable. The URL variable specified for each chart overlay is honored. This capability affects the following charts: DOT, HBAR, HLINE, VBAR, VLINE.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

WEIGHT=*numeric-variable*
specifies how observations are weighted. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction If your plot is overlaid with other categorization plots that also specify WEIGHT=, then the first WEIGHT variable that you specified is used for all of the plots.

COLAXIS Statement

Specifies the axis options for each X axis in the panel.

Notes: All COLAXIS options should be included in a single COLAXIS statement. Otherwise, some axis options might not be applied.

You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

COLAXIS *options*;

Summary of Optional Arguments

Appearance options

ALTERNATE
: adds reference ticks to each side of the panel and alternates the tick values for each row or column between the two sides.

COLORBANDS=NONE | EVEN | ODD
: specifies the display of alternating wall-color bands corresponding to the discrete axis bins.

COLORBANDSATTRS=*style-element*<(***options***)> | (*options*)
: specifies the fill appearance of the color band.

DISPLAY=ALL | NONE | (*options*)
: specifies which features of the axis are displayed.

GRID
: creates grid lines at each tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the grid lines.

MINORGRID
: creates grid lines at each minor tick on the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the minor grid lines.

Axis options

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED
: specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis.

INTEGER
: specifies that only integers are used for tick mark values.

INTERVAL=*interval-value*
: specifies the tick interval for a time axis.

INTERVALMULT=*positive-integer*
: specifies a multiplier to apply to the time interval that is in effect for the axis.

LOGBASE=2 | 10 | e
: specifies the base value for the logarithmic scale.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
: specifies how to scale and format the values for the major tick marks for logarithmic axes.

LOGVTYPE=EXPANDED | EXPONENT
: specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options.

MAX=*numeric-value*
: specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

MIN=*numeric-value*
: specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

MINOR
: adds minor tick marks to a linear, log, or time axis.

MINORCOUNT=*numeric-value*
: specifies the number of minor tick marks for the axis.

MINORINTERVAL=*time-interval*
: specifies the time interval between minor ticks.

NOTIMESPLIT
: prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX=*numeric-value*
: specifies an offset that follows the highest data value on the axis.

OFFSETMIN=*numeric-value*
: specifies an offset that precedes the lowest data value on the axis.

REFTICKS

REFTICKS=(*options*)
: adds tick marks to the side of the panel that is opposite from the specified axis.

REVERSE
: specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX=*numeric-value*
: Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN=*numeric-value*
: Specifies a threshold for displaying one more tick mark at the low end of the axis.

TYPE=DISCRETE | LINEAR | LOG | TIME
: specifies the type of axis.

Text options

FITPOLICY=*policy-value*
: specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally.

LABEL="*text-string*"
: specifies a label for the axis.

LABELATTRS=*style-element*<(**options**)> | (*options*)
: specifies the appearance of the axis labels.

LABELPOS=CENTER | DATACENTER | LEFT | RIGHT
: specifies the position of the axis label.

SPLITCHAR="*character-list*"
: splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

VALUEATTRS=*style-element*<(**options**)> | (*options*)
: specifies the appearance of the axis tick value labels.

VALUES=(*values-list*) | ("*string-list*")
: specifies the values for the ticks on an axis.

VALUESDISPLAY=("*value1*" "*value2*" ...)
: specifies the text that is to be displayed for the tick values that are defined in the VALUES= option.

VALUESFORMAT=DATA | *SAS-format*
: specifies how to format the values for major tick marks.

VALUESHINT
: specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option.

VALUESROTATE=DIAGONAL | DIAGONAL2 | VERTICAL
: specifies how the tick values are rotated on the axis.

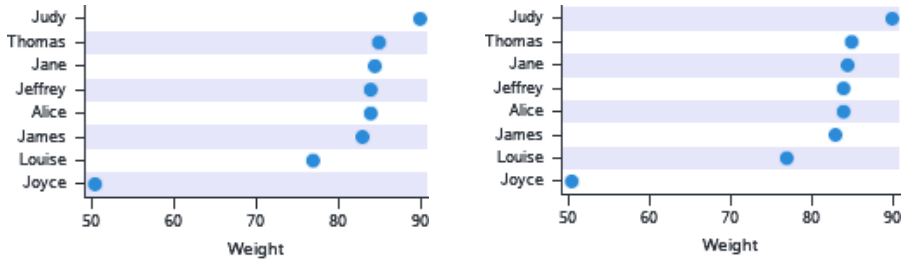
Optional Arguments

ALTERNATE
adds reference ticks to each side of the panel and alternates the tick values for each row or column between the two sides.

COLORBANDS=NONE | EVEN | ODD
specifies the display of alternating wall-color bands corresponding to the discrete axis bins. The following images show the results of ODD and EVEN settings:

Table 8.53 *Odd and Even COLORBANDS Settings*

COLORBANDS=ODD	COLORBANDS=EVEN



Default NONE

Restriction This option applies to discrete axes only.

Interaction Specifying this option for more than one axis in the graph might have unexpected results. The order in which color bands are drawn might not match the order in which the axis options are specified.

Note The full width of a color band is the distance between midpoints. When no axis offsets are specified, the first band begins at one-half of the midpoint distance, and the last band ends at one-half of the midpoint distance. When axis offsets are specified, the first and last color bands on the axis might extend into their adjacent offsets by as much as half the color-band width.

Tip Use the COLORBANDSATTRS= option to customize the color bands.

COLORBANDSATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill appearance of the color band. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interaction This option has no effect unless COLORBANDS= is also specified.

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED
specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis. Specify one of the following values:

DATA
places the values in the order in which they appear in the data.

Note: This value is not supported with data that is stored in a table.

FORMATTED
sorts the formatted values in ascending character order.

UNFORMATTED
sorts the unformatted values in ascending character order.

Default UNFORMATTED

DISPLAY=ALL | NONE | (*options*)
specifies which features of the axis are displayed.

ALL
displays all of the features of the axis.

NONE
specifies that none of the features of the axis are displayed.

(*options*)
specifies one or more axis features to hide.

Specify one or more of the following:

NOLABEL
hides the axis label

NOLINE
hides the axis line

Tips This value hides the axis line but has no effect on the graph's wall border, part of which occupies the same area as the axis line. To hide the border, specify NOBORDER in the PANELBY statement.

This value has no effect on baselines. For plots that support a baseline, such as bar charts and needle plots, you might need to suppress the baseline. In the plot statement, use the BASELINEATTRS= option to set the line thickness to 0.

NOTICKS
hides the tick marks on the axis

NOVALUES
hides the tick mark values on the axis

Default ALL

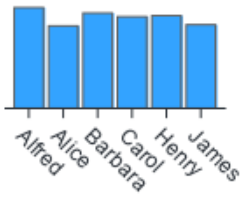
Interaction If the ALTERNATE option is also specified, the DISPLAY option affects both the primary axis and the alternate axis.

Example DISPLAY=(NOTICKS NOVALUES)

FITPOLICY=*policy-value*
specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally. Select one of the following values:

NONE
does not split the values.

ROTATE
rotates the value text 45 degrees.



TIP Use VALUESROTATE= to specify how the tick values are rotated on the axis.

Note: With a time axis, you must specify NOTIMESPLIT for this option to have any effect.

ROTATETHIN
attempts to use ROTATE, and then THIN to fit the values.

Note: With a time axis, you must specify NOTIMESPLIT for this option to have any effect.

SPLIT
splits the values at the character or characters specified in the SPLITCHAR= option.

No split occurs at split characters where a split is not needed. In that case, the split character is displayed with the text value. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Restriction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

SPLITALWAYS
always splits the values at the character or characters specified in the SPLITCHAR= option.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Interaction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

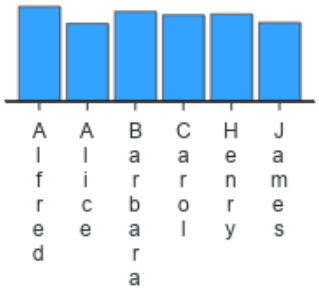
Example "Fit Policies for Axes" on page 1803

SPLITROTATE
attempts to use SPLIT, and then ROTATE to fit the values.

This is the default for discrete axes.

Note: This option has no effect on time axes.

STACKEDALWAYS
stacks the values vertically regardless of tick value collision.



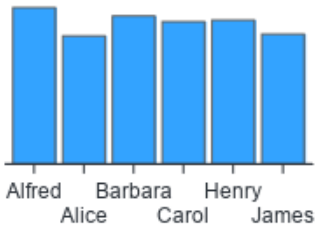
Interaction This option works only with a discrete axis.

STACKEDALWAYSTHIN
stacks the values vertically regardless of tick value collision.

If a collision is detected after stacking, then THIN as needed.

Interaction This option works only with a discrete axis.

STAGGER
shifts the values up and down.



Interaction When used with a time axis, this option has no effect unless you also specify NOTIMESPLIT in the axis statement.

STAGGERROTATE
attempts to use STAGGER, and then ROTATE to fit the values.

Interaction When used with a time axis, this option has no effect unless you also specify NOTIMESPLIT in the axis statement.

STAGGERTHIN
attempts to use STAGGER, and then THIN to fit the values.

Interaction When used with a time axis, this option has no effect unless you also specify NOTIMESPLIT in the axis statement.

THIN
removes some of the values from the axis.

This is the default for linear and time axes.

Defaults SPLITROTATE for discrete axes.

THIN for linear and time axes.

Restriction This option does not affect logarithmic axes.

See "Fit Policies for Axes" on page 1803

GRID
creates grid lines at each tick on the axis.

Interaction Grid lines are not displayed when you specify the COLORBANDS= option. The color bands take the place of grid lines.

Tip You can specify the MINORGRID option to create grid lines at each minor tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless GRID is also specified.

Examples
```
GRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:
```
GRIDATTRS=GraphAxisLines
```

INTEGER
specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL=*interval-value*
specifies the tick interval for a time axis. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

Specify one of the following values:

Table 8.54 Time Intervals

INTERVAL	Unit	Tick interval	Default tick value format
AUTO	DATE, TIME, or DATETIME	automatically chosen	automatically chosen
SECOND	TIME or DATETIME	second	TIME8.
MINUTE	TIME or DATETIME	minute	TIME8.
HOUR	TIME or DATETIME	hour	TIME8.
DAY	DATE or DATETIME	day	DATE9.
TENDAY	DATE or DATETIME	10 days	DATE9.
WEEK	DATE or DATETIME	7 days	DATE9.
SEMIMONTH	DATE or DATETIME	1st and 16th of each month	DATE9.
MONTH	DATE or DATETIME	month	MONYY7.
QUARTER	DATE or DATETIME	3 months	YYQC6.
SEMIYEAR	DATE or DATETIME	6 months	MONYY7.
YEAR	DATE or DATETIME	year	YEAR4.

Default AUTO

Tip If unwanted thinning of tick marks occurs when you specify INTERVAL=, try using minor intervals instead. For example, you might specify MINOR MINORINTERVAL=MONTH instead of INTERVAL=MONTH. As a result, in the example monthly tick marks are displayed even though the major values are thinned.

INTERVALMULT=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the axis.

Alias INTERVALMULTIPLIER

Default 1

Restriction This option applies to time axes only.

Interaction This option is ignored if VALUES=*values-list* is used.

Tip Use the INTERVAL= option to specify a different time interval.

Examples To specify 3-month intervals:
```
INTERVAL=MONTH INTERVALMULT=3
```

To specify 10-year intervals:
```
INTERVAL=YEAR INTERVALMULT=10
```

LABEL="*text-string*"
specifies a label for the axis.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LABELPOS=CENTER | DATACENTER | LEFT | RIGHT
specifies the position of the axis label.

CENTER
centers the axis label in the axis area (including any offsets).

The label is positioned below the tick values.

DATACENTER
centers the axis label in the axis tick display area (excluding any offsets).

The label is positioned below the tick values.

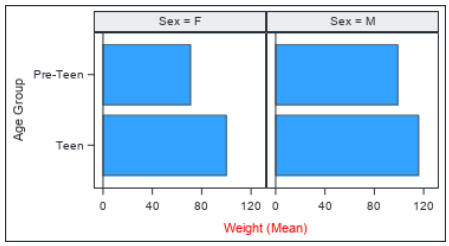
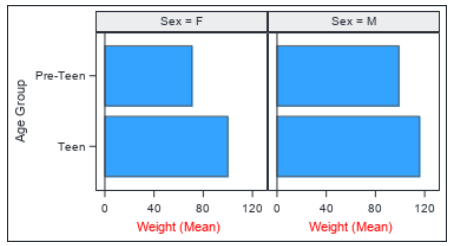
LEFT
positions the label to the left of the axis area.

The label is centered vertically in the axis area.

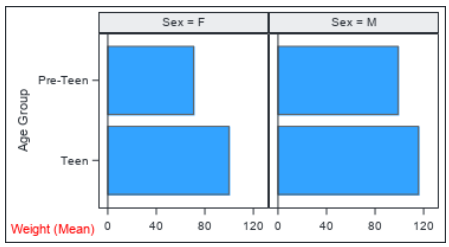
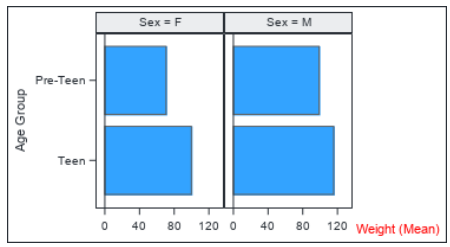
RIGHT
positions the label to the right of the axis area.

The label is centered vertically in the axis area.

The following figure shows the CENTER and DATACENTER positions for the red X-axis label “Weight (Mean).” An axis offset is applied to the maximum end of the axis in order to demonstrate the difference between CENTER and DATACENTER. CENTER centers the labels on the entire axis area, including the offset. DATACENTER centers the labels on the tick display areas, which does not include the offset.

CENTER	DATACENTER
	

The next figure shows the LEFT and RIGHT positions for the same axis labels.

LEFT	RIGHT
	

Default CENTER

LOGBASE=2 | 10 | e
specifies the base value for the logarithmic scale.

Default 10

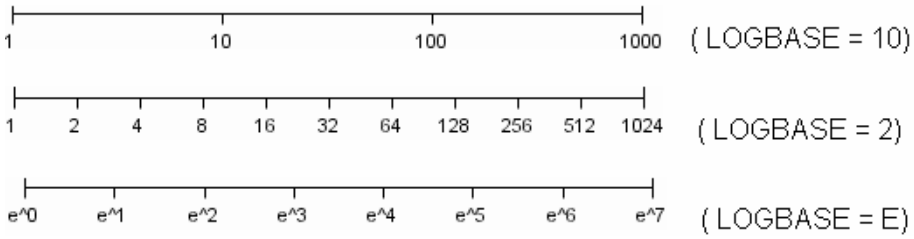
Interaction This option has no effect unless you also specify TYPE=LOG.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND
places the tick marks at integer powers of the base.

For example, if you specified **LOGBASE=2**, the tick marks might be at 1, 2, 4, 8, 16. See Figure 8.13 on page 765 .

Figure 8.12 *Graph Axes with LOGEXPAND*



LOGEXPONENT
places the tick marks at integer powers of the base, but identifies the values by the exponent.

For example, if you specified **LOGBASE=10**, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See Figure 8.14 on page 765 .

Figure 8.13 *An Axis with LOGEXPONENT*

LINEAR
places the tick marks at uniform linear intervals, but spaces them logarithmically.

In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See Figure 8.15 on page 765 .

Figure 8.14 *An Axis with LINEAR*

Default LOGEXPAND

Interaction This option has no effect unless you also specify TYPE=LOG.

LOGVTYPE=EXPANDED | EXPONENT
specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options. This option enables you to choose your preferred way of specifying log-axis values regardless of the LOGSTYLE= option value.

Specify one of the following values:

EXPANDED
the values are interpreted as integer powers of the base (decimal numbers).

EXPONENT
the values are interpreted as integer exponents of the base.

Default EXPANDED

Interaction This option has no effect unless you also specify TYPE=LOG. You must also specify values for the VALUES= option or the MIN= and MAX= options or all of them.

Tip This option is particularly useful when the log axis is an odd base (such as base E) or the axis log style is EXPONENT.

Examples The following example specifies MIN= and MAX= as exponent values instead of expanded values on an expanded Base 10 log axis. This results in Y-axis tick values of 10, 100, 1000, 10000, and 100000.

```
rowaxis type=log logbase=10 logstyle=logexpand
   logvtype=exponent
   min=1 max=5;
```

The following example specifies VALUES= as a list of expanded values instead of exponent values on an exponent Base 10 log axis. This results in X-axis tick values of 1, 2, 3, 4, and 5.

```
colaxis type=log logbase=10 logstyle=logexponent
   logvtype=expanded
   values=(10 100 1000 10000 100000);
```

MAX=*numeric-value*
specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

This option does not determine the maximum axis tick value displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the maximum value is expanded or interpreted as an exponent.

Tip The maximum axis tick value might differ from the MAX= value. The MAX= and MIN= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values. To display the MAX= value as the maximum tick value, use the VALUES= option.

MIN=*numeric-value*
specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

This option does not determine the minimum axis tick value displayed. The THRESHOLDMIN= value is used to determine the minimum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the minimum value is expanded or interpreted as an exponent.

Tip The minimum axis tick value might differ from the MIN= value. The MIN= and MAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values. To display the MIN= value as the minimum tick value, use the VALUES= option.

MINOR
adds minor tick marks to a linear, log, or time axis.

Restriction This option has no effect on discrete axes.

Interaction This option has no effect if you specify the VALUES= option.

Tip Use MINORCOUNT= to specify the number of tick marks.

MINORCOUNT=*numeric-value*
specifies the number of minor tick marks for the axis. This value determines the number of minor tick marks for each interval on the axis.

Restriction This option applies to linear and log axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

MINORGRID
creates grid lines at each minor tick on the axis.

Interaction This option has no effect unless GRID is also specified for the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the minor grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphMinorGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless MINORGRID is also specified.

Tip You can use GRIDATTRS= to change the appearance of the major grid lines.

Examples
```
MINORGRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:
```
MINORGRIDATTRS=GraphAxisLines
```

MINORINTERVAL=*time-interval*
specifies the time interval between minor ticks. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

For information about the intervals that you can select, see Table 8.75 on page 762.

Default AUTO

Restriction This option applies to time axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

NOTIMESPLIT
prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction This option applies to time axes only.

OFFSETMAX=*numeric-value*
specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the maximum end of the axis to the highest data value or highest tick value, whichever is greater.

For a discrete axis, the offset is measured from the maximum end of the axis to the midpoint of the last category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the maximum end of the axis. The sum of the minimum and maximum offsets does not exceed 50% of the axis length.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See “Adjusting Axis Offsets” on page 1788

OFFSETMIN=*numeric-value*
specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the beginning of the axis to the lowest data value or lowest tick value, whichever is less.

For a discrete axis, the offset is measured from the beginning of the axis to the midpoint of the first category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the minimum end of the axis. The sum of the minimum and maximum offsets does not exceed 50% of the axis length.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See "Adjusting Axis Offsets" on page 1788

REFTICKS
REFTICKS=(*options*)
adds tick marks to the side of the panel that is opposite from the specified axis. You can also specify *options*:

LABEL
in addition to the tick marks, displays the axis label.

NOTICKS
suppresses the tick marks.

VALUES
in addition to the tick marks, displays the values that are represented by the tick marks.

REVERSE
specifies that the tick values are displayed in reverse (descending) order.

SPLITCHAR="*character-list*"
splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect. When FITPOLICY=SPLIT is in effect, the text is split on occurrences of the specified split character or characters where it is required to fit the tick marks. When FITPOLICY=SPLITALWAYS is in effect, the text is split on all occurrences of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Restriction This option has no effect unless the axis is discrete.

Interactions This option has no effect unless FITPOLICY= is specified as either SPLIT, SPLITALWAYS, or SPLITROTATE. SPLITROTATE is the default for discrete X axes.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Default CENTER

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THRESHOLDMAX=*numeric-value*
Specifies a threshold for displaying one more tick mark at the high end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

THRESHOLDMIN=*numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

TYPE=DISCRETE | LINEAR | LOG | TIME

specifies the type of axis. Specify one of the following values:

DISCRETE

specifies an axis with discrete values.

If a character variable is assigned to an axis, then that the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable.

Note: Bar charts support a linear category axis.

LINEAR

specifies a linear scale for the axis.

This is the default axis type for numeric variables, except when the data is discrete, or when the numeric variable has a date or time format.

LOG

specifies a logarithmic scale for the axis.

This axis type is never a default.

Restriction A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

Interactions Use the LOGSTYLE= option to specify the scale and format for the tick values.

Use the LOGBASE= option to specify the base value.

Use the LOGVTYPE= option to specify how the values that are provided in the VALUES= option and the MIN= and MAX= options are interpreted.

TIME
specifies a time scale for the axis.

If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

VALUES=(*values-list*) | ("*string-list*")
specifies the values for the ticks on an axis. The syntax for this option varies depending on the type of axis.

- VALUES= (*values-list*) specifies tick values for linear, time, and logarithmic axes.
- VALUES=(*"string-list"*) specifies tick values for discrete axes. The values can be character or numeric.

values-list
specifies tick values for linear, time, and logarithmic axes.

Values in *values-list* can be specified as a simple list of values in the following form:

value <...*value-n*>

The values must be listed in either ascending or descending order. The values can also be specified as a range of values of a specific increment in the following form:

value-1 TO *value-2* BY *increment-value*

By default the increment value is 1. You can use a negative integer for increment to specify a value list in descending order.

You can use both forms when specifying values:

<*value ... value-n* >*value-1* TO *value-2* BY *increment-value* <*value ... value-n*>

In all forms, multiple *n* values can be separated by blanks or commas. Here are some examples:

- Place tick marks at 2, 4, and 6 on a linear axis:

```
values=(2 4 6)
```

- Place tick marks at 6, 4, and 2 on a linear axis:

```
values=(6,4,2)
```

- Place tick marks at 25MAY08, 04JUL08, and 23AUG08 on a time axis:

```
values=("25MAY08"d "04JUL08"d "23AUG08"d)
```

- Place tick marks from 2 to 10 at an increment of 2 on a linear axis:

```
values=(2 to 10 by 2)
```

- Place tick marks from 50 to 10 at an increment of -5 on a linear axis:

```
values=(50 to 10 by -5)
```

If the specified range is not evenly divisible by the increment value, the highest value displayed on the axis is the last incremental value below the ending value for the range. For example, this value list produces a maximum axis value of 9:

```
values=(0 to 10 by 3)
```

Restrictions This option has no effect on discrete axes.

If your VALUES= option creates more than 1000 values, then the option has no effect.

Interactions For logarithmic axes, use the LOGVTYPE option to control whether the values are expanded or interpreted as exponents.

If a custom format is applied to the value, the raw value is plotted first and then formatted.

Tips Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

For graphs such as histograms, the final bin might be truncated at the maximum data value. If you need to expand the axis in order to show all of the data, use the VALUESHINT option.

"string-list"

specifies a space-separated list of string values enclosed in parentheses for a discrete axis .

Each value in the list must be enclosed in quotation marks. Numeric values must also be enclosed in quotation marks.

Only the tick values that are included in the string list are displayed on the axis. The values are displayed in the order in which they are listed. The data values that are not in the list are dropped. The list can be a subset of the data values. It can also contain values that are not included in the actual data. A tick value that is not included in the data appears on the axis, but no data is represented at its tick mark.

Note: For an example that is used in the SGPLOT procedure, see Table 10.140 on page 1547.

Restrictions This option has no effect on linear, time, or logarithmic axes.

Each value must be enclosed in quotation marks and separated from adjacent values by a blank space.

Notes If the string list contains duplicate values, the first occurrence of the duplicated value in the list is honored while the remaining instances are ignored.

The axis data can be character or numeric.

If a numeric variable has an associated format, the specified values must use the same format.

Tip Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

Examples The following example specifies the axis tick values Sedan, Sports, Wagon, and SUV:

```
values=("Sedan" "Sports" "Wagon" "SUV")
```

The following example specifies the axis tick values 10, 20, 30, and 40:

```
values=("10" "20" "30" "40")
```

The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

TIP Along the axis, numeric tick values are arranged in ascending order while character values are arranged in the order in which they are used in the graph. In some cases, the resulting tick-value order might not be desirable, especially if the graph consists of multiple plots or if the data contains missing values. This option is useful in that case. You can use this option to set the order of the axis tick values.

VALUESDISPLAY=("*value1*" "*value2*" ...)
specifies the text that is to be displayed for the tick values that are defined in the VALUES= option. The list of values must be enclosed in parentheses. Each value must be enclosed in quotation marks and separated from adjacent values by a blank space. Numeric values must also be enclosed in quotation marks.

Restriction This option applies only to discrete and linear axes.

Interaction This option should be used with the VALUES= option. The number of items in the list for this option should equal the number of items in the list for the VALUES= option.

Example The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

VALUESFORMAT=DATA | *SAS-format*
specifies how to format the values for major tick marks. This option supports discrete and logarithmic axes.

DATA
uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

SAS-format
specifies a format to apply to the major tick values.

Restriction This option currently honors most, but not every, SAS format. For more information, see "Format Support in ODS Graphics" in *SAS Graph Template Language: Reference*.

Notes If you specify a format that significantly reduces precision, then, because of tick-value rounding, the plot data elements might not align properly with the axis tick values. In that case, specify a tick-value format with a higher precision.

A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F5.1

Alias TICKVALUEFORMAT

Restriction For discrete axes, only character formats are supported.

Interactions This option is ignored when LOGSTYLE=LOGEXPONENT.

When LOGSTYLE =LOGEXPAND, this option is honored for the base 10 and base 2 logarithmic scales, and is ignored for the base E scale.

When LOGSTYLE =LINEAR, this option is honored for the base 10, base 2, and base E logarithmic scales.

VALUESHINT
specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option. Tick values specified by the VALUES= option are treated as suggestions for the tick values that actually appear, so that the actual data range of the axis is preserved. If the *values-list* contains values that are less than the data minimum, or greater than the data maximum, those values are ignored.

Interaction This option has no effect unless you also specify the VALUES= option.

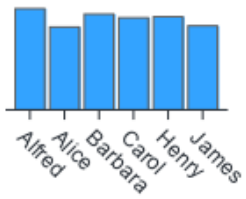
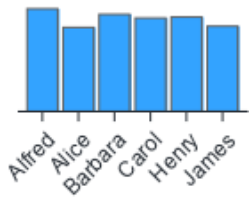
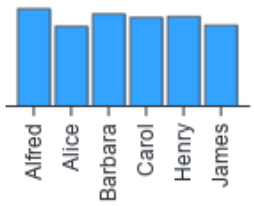
VALUESROTATE=DIAGONAL | DIAGONAL2 | VERTICAL
specifies how the tick values are rotated on the axis.

DIAGONAL
rotates the tick values diagonally.

DIAGONAL2
rotates the tick values to a reverse diagonal angle.

VERTICAL
rotates the tick values vertically.

The values are always drawn from bottom to top.

DIAGONAL	DIAGONAL2	VERTICAL
		

Default DIAGONAL

Interactions This option works through the axis fit policy. If there are no tick collisions, no rotation occurs.

With a time axis, you must specify NOTIMESPLIT for this option to have any effect.

COLAXISTABLE Statement

Creates a basic plot that displays data values at specific locations along the horizontal axis.

Requirements: At least one other plot statement must be used in the procedure. This other plot becomes the primary plot. The primary plot provides the default placement of values along the axis table's horizontal axis.

For the axis table, you must specify at least one variable. This variable determines which values are displayed in the axis table.

Interactions: You can specify a variable for the X= option to identify the locations along the axis where the values are displayed. If you do not specify X=, the locations are determined by the X variable in the primary plot. When the primary plot is a categorical chart (bar, line, or dot), the category variable from the chart is used. With the exception of that X (or category) variable and any group variable that might be specified, axis tables are separate plots and are unaware of the options specified in the accompanying plots.

When used with bar charts, line charts, and dot plots, all axis tables must align with the category axis of the chart. If a statement uses the wrong orientation, the statement is rejected with a message in the SAS log. For example, if your procedure has an HBAR statement along with a COLAXISTABLE statement, the COLAXISTABLE statement is rejected with a message.

Axis tables cannot be used when the primary plot is one of the following plot types: BAND, BLOCK, FRINGE, REG, LOESS, and PBSPLINE. The axis table requires at least one other plot statement that is not one of these types. Otherwise, the axis table is not created and an error is written to the log.

Note: You can include multiple COLAXISTABLE statements in an SGPANEL procedure step.

Syntax

COLAXISTABLE *variable* <*...variable-n*> <*/options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORGROUP=*variable*
: specifies a variable that is used to determine the color of the table values.

DROPONMISSING
: specifies that the entire axis table is dropped when all of the values are missing.

NOMISSINGCHAR
: suppresses the display of the MISSING character (.) for missing numeric values.

PAD=*numeric-value*<*units*> | (*pad-options*)
: specifies the amount of extra space that is added inside the table border.

POSITION=BOTTOM | TOP
: specifies the position of the axis table at the bottom or top of the graph.

SEPARATOR

creates a separating line between the axis table or axis tables and the plot.

TEXTGROUP=*attribute-map-group-variable*
specifies the group variable that is used in a discrete attribute map data set to map text attributes to values for each observation.

TEXTGROUPID=*attribute-map-id*
specifies an attribute ID for the TEXTGROUP= option.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the axis table values.

Class options

CLASS=*variable*
creates a separate axis table for each unique value of the specified variable.

CLASSDISPLAY=STACK | CLUSTER
specifies how the class values are displayed.

CLASSORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the order in which the class values are displayed.

NOMISSINGCLASS
specifies that missing values of the class variable are not included in the table.

Label options

LABEL

LABEL="*text-string*"

NOLABEL
specifies whether the table label is shown or hidden.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the color and font attributes of the axis table label.

LABELPOS=LEFT | RIGHT
specifies the position of the labels at the left or right side of the axis table.

STATLABEL

NOSTATLABEL
specifies whether the variable statistic is displayed in the table's label.

Plot options

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the axis table.

X=*variable*
Specifies the X variable to use to align the table values to the X axis.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Title options

TITLE="*text-string*"
specifies a title for the axis table.

TITLEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the title for the axis table.

Required Argument

***variable* <...*variable-n*>**
specifies one or more variables for the axis table. When multiple variables are specified, the axis tables are stacked one on the other. Any options that you add to the statement apply to all the variables that are specified in that statement.

Note When the variable specified is a character variable, the first value of each category is displayed in the axis table. When the variable is numeric, the axis table displays the sum statistic. You can modify the statistic using the STAT= option.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CLASS=*variable*
creates a separate axis table for each unique value of the specified variable. This option acts as a classification variable for the axis table. Each axis table is labeled by the class value.

Restriction The CLASS option is ignored when the axis table is used with bar, line, or dot charts. If the GROUP= option is specified in the chart, that group variable is used as the CLASS variable for all axis tables.

Interaction If NOLABEL is also specified, then the class labels are removed.

Tips Use the CLASSDISPLAY= option to control whether the class values are clustered or stacked.

Use the CLASSORDER= option to control the order in which the class values are displayed.

CLASSDISPLAY=STACK | CLUSTER
specifies how the class values are displayed.

STACK
displays the class values vertically at each midpoint value on the X axis.

CLUSTER
displays the class values horizontally at each midpoint value on the X axis.

Restriction The CLASSDISPLAY=CLUSTER option applies only when the axis table is on a discrete axis.

Default STACK

Interaction For this option to have any effect, the CLASS= option must be specified.

CLASSORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the order in which the class values are displayed.

DATA
displays the class values in the order in which they occur in the data.

REVERSEDATA
displays the class values in the reverse order from which they occur in the data.

Tip This option is useful when the plot axis is reversed.

ASCENDING
displays the class values in ascending order.

DESCENDING
displays the class values in descending order.

Default DATA

Interactions This option is ignored when the axis table is used with DOT, HBAR, VBAR, HLINE, and VLINE statements. If the GROUPORDER= option is specified in the chart, that group order variable is used as the class order variable for all axis tables.

For this option to have any effect, the CLASS= option must be specified.

COLORGROUP=*variable*
specifies a variable that is used to determine the color of the table values. Once the variable values are found, the value colors are taken from the GraphData1–GraphData*n* style elements in the current style. The CONTRASTCOLOR attribute is used for the value text.

Restriction When used with DOT, HBAR, VBAR, HLINE, and VLINE statements, this option has no effect unless *variable* is used in the accompanying chart.

Note This option is used only to color the table values. If you want to set additional text attributes, used the TEXTGROUP= option instead.

DROPONMISSING
specifies that the entire axis table is dropped when all of the values are missing. Consider using this option if the SAS log indicates that the specified data column used for the axis table is missing all values.

LABEL
LABEL="*text-string*"
NOLABEL
specifies whether the table label is shown or hidden. If you specify LABEL=, then you can also specify a text string for the label.

Defaults LABEL

If you do not specify a text string, then the variable name is used for the label. Or, if CLASS= is also specified, then the unique values of the specified class variable are used for the labels.

Tip Use the LABELATTRS= option to modify the label text attributes. Use the LABELPOS= option to move the label.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the color and font attributes of the axis table label. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

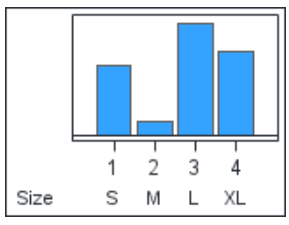
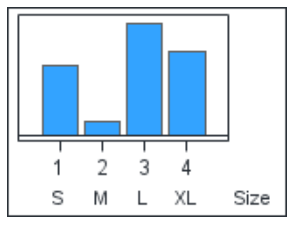
For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Restriction Group behavior occurs only when the CLASS= and COLORGROUP= option values are the same.

Interactions This option has no effect if NOLABEL is also specified.

If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphValueText style element.

LABELPOS=LEFT | RIGHT
specifies the position of the labels at the left or right side of the axis table.

LEFT	RIGHT
	

Default LEFT

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGCHAR
suppresses the display of the MISSING character (.) for missing numeric values. Missing numeric values are displayed as blanks.

NOMISSINGCLASS
specifies that missing values of the class variable are not included in the table.

Interaction For this option to have any effect, the CLASS= option must be specified.

PAD=*numeric-value<units>* | (*pad-options*)
specifies the amount of extra space that is added inside the table border. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

numeric-value<units>
specifies a dimension to use for the extra space at the table border.

(*pad-options*)
specifies a space-separated list of pad options, enclosed in parentheses.

Specify one or more of the following name-value-pair options:

LEFT=*numeric-value<units>*
specifies the amount of extra space added to the left side.

Default 4 px

RIGHT=*numeric-value<units>*
specifies the amount of extra space added to the right side.

Default 4 px

TOP=*dimension*
specifies the amount of extra space added to the top.

Default 0 px

BOTTOM=*dimension*
specifies the amount of extra space added to the bottom.

Default 0 px

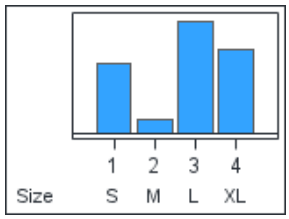
Note Sides that are not assigned padding are padded with the default amount of space.

Tip Use *pad-options* to create non-uniform padding.

Note The default units for *dimension* are pixels. If you want to specify values in other units, then you must specify the desired units with the value. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

POSITION=BOTTOM | TOP

specifies the position of the axis table at the bottom or top of the graph.

BOTTOM	TOP
	

Default BOTTOM

SEPARATOR

creates a separating line between the axis table or axis tables and the plot.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM

specifies the statistic for the axis table. The statistic is applied only to numeric variables. For non-categorical plots, only the SUM and MEAN statistics are available.

Specify one of the following:

FREQ

the frequency of the axis table variable.

Interactions For this value to take effect, the graph must use a categorical plot (bar chart, line chart, or dot plot).

The FREQ value has a different effect when the axis table variable is the same as the categorical variable of the primary plot. In that case, the axis table variable is treated as a category (class) variable instead of a response (analysis) variable when computing the frequency. The FREQ value is the frequency of the categorical variable.

MEAN

the mean of the axis table variable.

MEDIAN

the median of the axis table variable.

Interactions For this value to take effect, the graph must use a categorical plot (bar chart, line chart, or dot plot).

This value is ignored when the axis table variable is the same as the categorical variable of the primary plot. In that case, the FREQ statistic is used and a message is written to the SAS log.

PERCENT

the percentage of the sum of the axis table variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions For this value to take effect, the graph must use a categorical plot (bar chart, line chart, or dot plot).

The PERCENT value has a different effect when the axis table variable is the same as the categorical variable of the primary plot. In that case, the axis table variable is treated as a category (class) variable instead of as a response (analysis) variable when computing the percentage. Therefore, the PERCENT value is calculated as a percentage of frequency instead of a percentage of the sum.

The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the PCTLEVEL= option in the PROC SGPANEL statement.

You can use the PCTNDEC= option in the PROC SGPANEL statement to control the number of decimals to be used when calculating the percent values. The default value is 1.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level will be zero.

SUM

the sum of the axis table variable.

Defaults SUM

When used with a categorical plot and the axis table variable is the same as the categorical variable, FREQ is the default.

Restrictions The STAT= option is applied only to numeric variables.

Only SUM and MEAN are supported for non-categorical plots.

When used with a categorical plot and the axis table variable is the same as the categorical variable, only FREQ and PERCENT are supported.

Interaction Any STAT= value specified in the primary plot has no effect on the axis table statistic.

STATLABEL | NOSTATLABEL
specifies whether the variable statistic is displayed in the table's label. STATLABEL forces the statistic to be displayed in the label. NOSTATLABEL removes the statistic from the label.

Defaults The statistic is displayed for the variable.

When a custom label is assigned to the variable, the statistic is not displayed.

Interaction This option has no effect unless the STAT= option is also specified in the axis table statement.

TEXTGROUP=*attribute-map-group-variable*
specifies the group variable that is used in a discrete attribute map data set to map text attributes to values for each observation. You specify this option only if you are using an attribute map to control visual attributes of the graph. The variable's values must correspond to the values in the VALUE variable in the attribute map data set. For more information, see Chapter 18, "Using Discrete Attribute Maps," on page 1901.

Requirement The TEXTGROUPID= option specifies the attribute ID to use for the attribute mapping. If TEXTGROUPID= is not specified, then the ATTRID= option is used. If the ATTRID= option is also not specified, then the TEXTGROUP option is ignored.

Interaction When this option is specified, the COLORGROUP= option is ignored.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

TEXTGROUPID=*attribute-map-id*
specifies an attribute ID for the TEXTGROUP= option.

Default If neither TEXTGROUPID= nor ATTRID= is specified, the TEXTGROUP= option is ignored.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

TITLE="*text-string*"
specifies a title for the axis table. It the axis table statement specifies more than one variable, the title is displayed for each variable.

Tip Use the TITLEATTRS= option to modify the title text attributes.

TITLEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the title for the axis table. You can specify the appearance by using a style element or by specifying specific options. If you

specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attribute of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attribute of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interactions This option has no effect unless TITLE= is also specified.

If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphDataText style element.

Examples
```
TITLEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TITLEATTRS=GraphTitleText
```

VALUEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the axis table values. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphDataText style element.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

X=*variable*

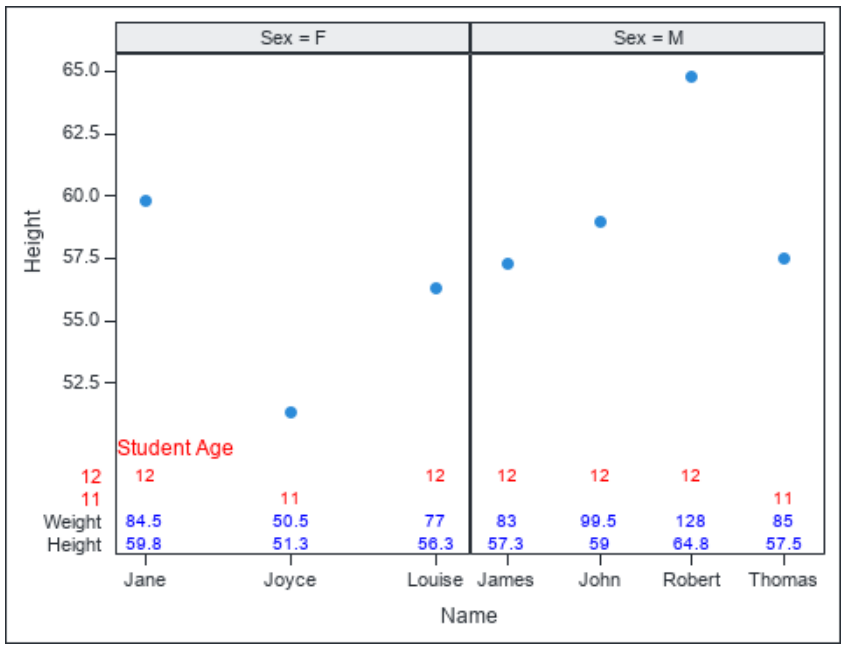
Specifies the X variable to use to align the table values to the X axis.

Default When this option is not specified, the X variable of the primary plot is used by default.

Interaction This option is ignored when the axis table is used with a categorical chart (bar, line, or dot). The category variable from the chart is used instead.

Example: COLAXISTABLE Statements

This example shows axis tables along the X axis of a scatter plot that plots student height.



Here are the noteworthy features about the axis tables:

- The first axis table statement displays the age (in red) of each student above the X axis. The ages appear on different rows because the CLASS= on page 812AGE option is used.
- The second axis table statement specifies two variables, height and weight.

Here is the SAS code for this example.

```
ods graphics / width=5in;

proc sgpanel data=sashelp.class (where=(age < 13));
  panelby sex / uniscale=row;
  scatter x=name y=height;
  colaxistable age / class=age title="Student Age"
      valueattrs=(color=red)
      labelattrs=(color=red)
      titleattrs=(color=red);
```

```
  colaxistable weight height / valueattrs=(color=blue);
run;
```

For an example that shows vertical axis tables along the Y axis, see "Example: ROWAXISTABLE Statements" on page 821.

ROWAXIS Statement

Specifies the axis options for each Y axis in the panel.

Notes:

All ROWAXIS options should be included in a single ROWAXIS statement. Otherwise, some axis options might not be applied.

You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

ROWAXIS *options*;

Summary of Optional Arguments

Appearance options

ALTERNATE
: adds reference ticks to each side of the panel and alternates the tick values for each row or column between the two sides.

COLORBANDS=NONE | EVEN | ODD
: specifies the display of alternating wall-color bands corresponding to the discrete axis bins.

COLORBANDSATTRS=*style-element*<(***options***)> | (*options*)
: specifies the fill appearance of the color band.

DISPLAY=ALL | NONE | (*options*)
: specifies which features of the axis are displayed.

GRID
: creates grid lines at each tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the grid lines.

MINORGRID
: creates grid lines at each minor tick on the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the minor grid lines.

Axis options

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED

specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis.

INTEGER
specifies that only integers are used for tick mark values.

INTERVAL=*interval-value*
specifies the tick interval for a time axis.

INTERVALMULT=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the axis.

LOGBASE=2 | 10 | e
specifies the base value for the logarithmic scale.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
specifies how to scale and format the values for the major tick marks for logarithmic axes.

LOGVTYPE=EXPANDED | EXPONENT
specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options.

MAX=*numeric-value*
specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

MIN=*numeric-value*
specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

MINOR
adds minor tick marks to a linear, log, or time axis.

MINORCOUNT=*numeric-value*
specifies the number of minor tick marks for the axis.

MINORINTERVAL=*time-interval*
specifies the time interval between minor ticks.

NOTIMESPLIT
prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX=*numeric-value*
specifies an offset that follows the highest data value on the axis.

OFFSETMIN=*numeric-value*
specifies an offset that precedes the lowest data value on the axis.

REFTICKS

REFTICKS=(*options*)
adds tick marks to the side of the panel that is opposite from the specified axis.

REVERSE
specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX=*numeric-value*
Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN=*numeric-value*
Specifies a threshold for displaying one more tick mark at the low end of the axis.

TYPE=DISCRETE | LINEAR | LOG | TIME
specifies the type of axis.

Text options

FITPOLICY=NONE | SPLIT | SPLITALWAYS | THIN
specifies the method that is used to fit tick mark values on a vertical axis when there is not enough room to draw them normally.

LABEL="*text-string*"
specifies a label for the axis.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the axis labels.

LABELPOS=BOTTOM | CENTER | DATACENTER | TOP
specifies the position of the axis label.

SPLITCHAR="*character-list*"
splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the axis tick value labels.

VALUES=(*values-list*) | ("*string-list*")
specifies the values for the ticks on an axis.

VALUESDISPLAY=("*value1*" "*value2*" ...)
specifies the text that is to be displayed for the tick values that are defined in the VALUES= option.

VALUESFORMAT=DATA | *SAS-format*
specifies how to format the values for major tick marks.

VALUESHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment for all of the tick values that are displayed on the axis.

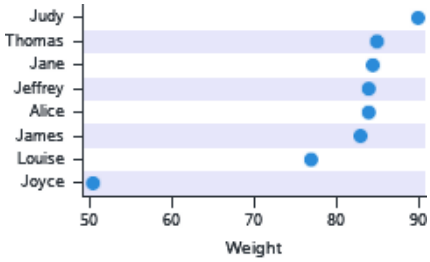
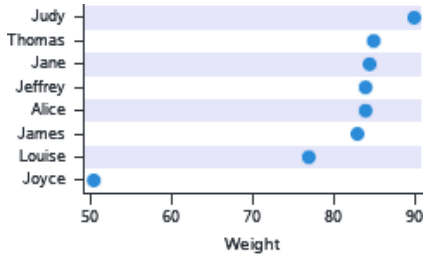
VALUESHINT
specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option.

Optional Arguments

ALTERNATE
adds reference ticks to each side of the panel and alternates the tick values for each row or column between the two sides.

COLORBANDS=NONE | EVEN | ODD
specifies the display of alternating wall-color bands corresponding to the discrete axis bins. The following images show the results of ODD and EVEN settings:

Table 8.55 *Odd and Even COLORBANDS Settings*

COLORBANDS=ODD	COLORBANDS=EVEN
	

Default NONE

Restriction This option applies to discrete axes only.

Interaction Specifying this option for more than one axis in the graph might have unexpected results. The order in which color bands are drawn might not match the order in which the axis options are specified.

Note The full width of a color band is the distance between midpoints. When no axis offsets are specified, the first band begins at one-half of the midpoint distance, and the last band ends at one-half of the midpoint distance. When axis offsets are specified, the first and last color bands on the axis might extend into their adjacent offsets by as much as half the color-band width.

Tip Use the COLORBANDSATTRS= option to customize the color bands.

COLORBANDSATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill appearance of the color band. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interaction This option has no effect unless COLORBANDS= is also specified.

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED
specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis. Specify one of the following values:

DATA
places the values in the order in which they appear in the data.

Note: This value is not supported with data that is stored in a table.

FORMATTED
sorts the formatted values in ascending character order.

UNFORMATTED
sorts the unformatted values in ascending character order.

Default UNFORMATTED

DISPLAY=ALL | NONE | (*options*)
specifies which features of the axis are displayed.

ALL
displays all of the features of the axis.

NONE
specifies that none of the features of the axis are displayed.

(*options*)
specifies one or more axis features to hide.

Specify one or more of the following:

NOLABEL
hides the axis label

NOLINE
hides the axis line

Tips This value hides the axis line but has no effect on the graph's wall border, part of which occupies the same area as the axis line. To hide the border, specify NOBORDER in the PANELBY statement.

This value has no effect on baselines. For plots that support a baseline, such as bar charts and needle plots, you might need to suppress the baseline. In the plot statement, use the BASELINEATTRS= option to set the line thickness to 0.

NOTICKS
hides the tick marks on the axis

NOVALUES
hides the tick mark values on the axis

Default ALL

Interaction If the ALTERNATE option is also specified, the DISPLAY option affects both the primary axis and the alternate axis.

Example DISPLAY=(NOTICKS NOVALUES)

FITPOLICY=NONE | SPLIT | SPLITALWAYS | THIN
specifies the method that is used to fit tick mark values on a vertical axis when there is not enough room to draw them normally. Select one of the following values:

NONE
does not split the values.

SPLIT
splits the values at the character or characters specified in the SPLITCHAR= option.

No split occurs at split characters where a split is not needed. In that case, the split character is displayed with the text value. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Restriction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

SPLITALWAYS
always splits the values at the character or characters specified in the SPLITCHAR= option.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Restriction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

Example "Fit Policies for Axes" on page 1803

THIN
removes some of the values from the axis.

Default THIN

See "Fit Policies for Axes" on page 1803

GRID
creates grid lines at each tick on the axis.

Interaction Grid lines are not displayed when you specify the COLORBANDS= option. The color bands take the place of grid lines.

Tip You can specify the MINORGRID option to create grid lines at each minor tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless GRID is also specified.

Examples `GRIDATTRS=(color=green pattern=longdash thickness=2)`

Here is an example that specifies a style element:
`GRIDATTRS=GraphAxisLines`

INTEGER
specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL=*interval-value*
specifies the tick interval for a time axis. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

Specify one of the following values:

Table 8.56 *Time Intervals*

INTERVAL	Unit	Tick interval	Default tick value format
AUTO	DATE, TIME, or DATETIME	automatically chosen	automatically chosen
SECOND	TIME or DATETIME	second	TIME8.
MINUTE	TIME or DATETIME	minute	TIME8.
HOUR	TIME or DATETIME	hour	TIME8.
DAY	DATE or DATETIME	day	DATE9.
TENDAY	DATE or DATETIME	10 days	DATE9.
WEEK	DATE or DATETIME	7 days	DATE9.
SEMIMONTH	DATE or DATETIME	1st and 16th of each month	DATE9.
MONTH	DATE or DATETIME	month	MONYY7.

INTERVAL	Unit	Tick interval	Default tick value format
QUARTER	DATE or DATETIME	3 months	YYQC6.
SEMIYEAR	DATE or DATETIME	6 months	MONYY7.
YEAR	DATE or DATETIME	year	YEAR4.

Default AUTO

Tip If unwanted thinning of tick marks occurs when you specify INTERVAL=, try using minor intervals instead. For example, you might specify MINOR MINORINTERVAL=MONTH instead of INTERVAL=MONTH. As a result, in the example monthly tick marks are displayed even though the major values are thinned.

INTERVALMULT=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the axis.

Alias INTERVALMULTIPLIER

Default 1

Restriction This option applies to time axes only.

Interaction This option is ignored if VALUES=*values-list* is used.

Tip Use the INTERVAL= option to specify a different time interval.

Examples To specify 3-month intervals:

```
INTERVAL=MONTH INTERVALMULT=3
```

To specify 10-year intervals:

```
INTERVAL=YEAR INTERVALMULT=10
```

LABEL="*text-string*"
specifies a label for the axis.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LABELPOS=BOTTOM | CENTER | DATACENTER | TOP
specifies the position of the axis label.

BOTTOM
positions the label at the bottom of the axis area.

The label is oriented horizontally (unrotated).

The label is right-justified in the axis area. If there is not sufficient room in the axis area to display the label, the label expands to the right of the axis.

CENTER
centers the axis label in the axis area (including any offsets).

The label is oriented vertically (rotated).

The label is positioned to the left of the tick values.

DATACENTER
centers the axis label in the axis tick display area (excluding any offsets).

The label is oriented vertically (rotated).

The label is positioned to the left of the tick values.

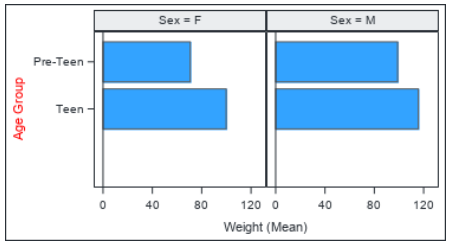
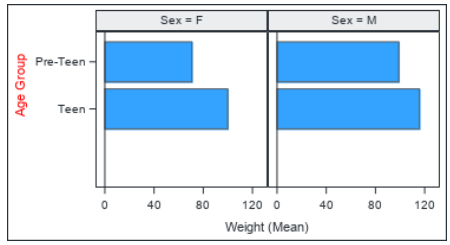
TOP
positions the label at the top of the axis area.

The label is oriented horizontally (unrotated).

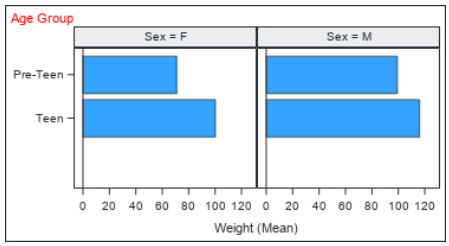
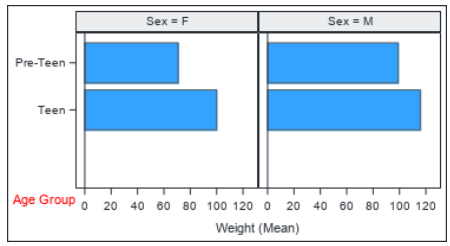
The label is right-justified in the axis area. If there is not sufficient room in the axis area to display the label, the label expands to the right of the axis.

The following figure shows the CENTER and DATACENTER positions for the red Y-axis label “Age Group.” An axis offset is applied to the maximum end of the axis in order to demonstrate the difference between CENTER and DATACENTER. (For HBAR plot statements, the Y axis is reversed by default. The offset is created at the bottom end of the Y axis.)

In the example, CENTER centers the labels on the entire axis area, including the offset. DATACENTER centers the labels on the tick display areas, which does not include the offset.

CENTER	DATACENTER
	

The next figure shows the TOP and BOTTOM positions for the same axis labels.

TOP	BOTTOM
	

Default CENTER

LOGBASE=2 | 10 | e
specifies the base value for the logarithmic scale.

Default 10

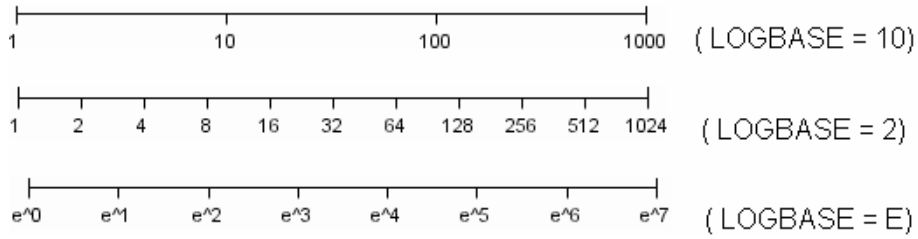
Interaction This option has no effect unless you also specify TYPE=LOG.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND
places the tick marks at integer powers of the base.

For example, if you specified **LOGBASE=2**, the tick marks might be at 1, 2, 4, 8, 16. See Figure 8.16 on page 797 .

Figure 8.15 *Graph Axes with LOGEXPAND*



LOGEXPONENT
places the tick marks at integer powers of the base, but identifies the values by the exponent.

For example, if you specified **LOGBASE=10**, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See Figure 8.17 on page 798 .

Figure 8.16 *An Axis with LOGEXPONENT*

LINEAR
places the tick marks at uniform linear intervals, but spaces them logarithmically.

In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See Figure 8.18 on page 798 .

Figure 8.17 *An Axis with LINEAR*

Default LOGEXPAND

Interaction This option has no effect unless you also specify TYPE=LOG.

LOGVTYPE=EXPANDED | EXPONENT
specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options. This option enables you to choose your preferred way of specifying log-axis values regardless of the LOGSTYLE= option value.

Specify one of the following values:

EXPANDED
the values are interpreted as integer powers of the base (decimal numbers).

EXPONENT
the values are interpreted as integer exponents of the base.

Default EXPANDED

Interaction This option has no effect unless you also specify TYPE=LOG. You must also specify values for the VALUES= option or the MIN= and MAX= options or all of them.

Tip This option is particularly useful when the log axis is an odd base (such as base E) or the axis log style is EXPONENT.

Examples The following example specifies MIN= and MAX= as exponent values instead of expanded values on an expanded Base 10 log axis.

This results in Y-axis tick values of 10, 100, 1000, 10000, and 100000.

```
rowaxis type=log logbase=10 logstyle=logexpand
   logvtype=exponent
   min=1 max=5;
```

The following example specifies VALUES= as a list of expanded values instead of exponent values on an exponent Base 10 log axis. This results in X-axis tick values of 1, 2, 3, 4, and 5.

```
colaxis type=log logbase=10 logstyle=logexponent
   logvtype=expanded
   values=(10 100 1000 10000 100000);
```

MAX=*numeric-value*

specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

This option does not determine the maximum axis tick value displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the maximum value is expanded or interpreted as an exponent.

Tip The maximum axis tick value might differ from the MAX= value. The MAX= and MIN= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values. To display the MAX= value as the maximum tick value, use the VALUES= option.

MIN=*numeric-value*

specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

This option does not determine the minimum axis tick value displayed. The THRESHOLDMIN= value is used to determine the minimum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the minimum value is expanded or interpreted as an exponent.

Tip The minimum axis tick value might differ from the MIN= value. The MIN= and MAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to

determine the axis tick values. To display the MIN= value as the minimum tick value, use the VALUES= option.

MINOR
adds minor tick marks to a linear, log, or time axis.

Restriction This option has no effect on discrete axes.

Interaction This option has no effect if you specify the VALUES= option.

Tip Use MINORCOUNT= to specify the number of tick marks.

MINORCOUNT=*numeric-value*
specifies the number of minor tick marks for the axis. This value determines the number of minor tick marks for each interval on the axis.

Restriction This option applies to linear and log axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

MINORGRID
creates grid lines at each minor tick on the axis.

Interaction This option has no effect unless GRID is also specified for the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the minor grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphMinorGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless MINORGRID is also specified.

Tip You can use GRIDATTRS= to change the appearance of the major grid lines.

Examples
```
MINORGRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:
```
MINORGRIDATTRS=GraphAxisLines
```

MINORINTERVAL=*time-interval*
specifies the time interval between minor ticks. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

For information about the intervals that you can select, see Table 8.77 on page 794.

Default AUTO

Restriction This option applies to time axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

NOTIMESPLIT
prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction This option applies to time axes only.

OFFSETMAX=*numeric-value*
specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the maximum end of the axis to the highest data value or highest tick value, whichever is greater.

For a discrete axis, the offset is measured from the maximum end of the axis to the midpoint of the last category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the maximum end of the axis. The sum of the minimum and maximum offsets does not exceed 50% of the axis height.

Interaction For HBOX, HBAR, HLINE, and DOT plot statements, the Y axis is reversed by default, so the axis origin is at the top. For these plots, OFFSETMAX= creates an offset at the end of the Y axis nearer to the origin.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See "Adjusting Axis Offsets" on page 1788

OFFSETMIN=*numeric-value*
specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the beginning of the axis to the lowest data value or lowest tick value, whichever is less.

For a discrete axis, the offset is measured from the beginning of the axis to the midpoint of the first category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the minimum end of the axis. The

sum of the minimum and maximum offsets does not exceed 50% of the axis height.

Interaction For HBOX, HBAR, HLINE, and DOT plot statements, the Y axis is reversed by default, so the axis origin is at the top. For these plots, OFFSETMIN= creates an offset at the end of the Y axis farther from the origin.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See "Adjusting Axis Offsets" on page 1788

REFTICKS
REFTICKS=(*options*)
adds tick marks to the side of the panel that is opposite from the specified axis. You can also specify *options*:

LABEL
in addition to the tick marks, displays the axis label.

NOTICKS
suppresses the tick marks.

VALUES
in addition to the tick marks, displays the values that are represented by the tick marks.

REVERSE
specifies that the tick values are displayed in reverse (descending) order.

SPLITCHAR="*character-list*"
splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect. When FITPOLICY=SPLIT is in effect, the text is split on occurrences of the specified split character or characters where it is required to fit the tick marks. When FITPOLICY=SPLITALWAYS is in effect, the text is split on all occurrences of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Restriction This option has no effect unless the axis is discrete.

Interactions This option has no effect unless FITPOLICY= is specified as either SPLIT or SPLITALWAYS.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP

specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT

specifies the horizontal alignment of the value text that is being split.

Default RIGHT

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THRESHOLDMAX=*numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

THRESHOLDMIN=*numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

TYPE=DISCRETE | LINEAR | LOG | TIME

specifies the type of axis. Specify one of the following values:

DISCRETE

specifies an axis with discrete values.

If a character variable is assigned to an axis, then that the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable.

Note: Bar charts support a linear category axis.

LINEAR

specifies a linear scale for the axis.

This is the default axis type for numeric variables, except when the data is discrete, or when the numeric variable has a date or time format.

LOG

specifies a logarithmic scale for the axis.

This axis type is never a default.

Restriction A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

Interactions Use the LOGSTYLE= option to specify the scale and format for the tick values.

Use the LOGBASE= option to specify the base value.

Use the LOGVTYPE= option to specify how the values that are provided in the VALUES= option and the MIN= and MAX= options are interpreted.

TIME
specifies a time scale for the axis.

If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

VALUES=(*values-list*) | ("*string-list*")
specifies the values for the ticks on an axis. The syntax for this option varies depending on the type of axis.

- VALUES= (*values-list*) specifies tick values for linear, time, and logarithmic axes.
- VALUES=(*"string-list"*) specifies tick values for discrete axes. The values can be character or numeric.

values-list
specifies tick values for linear, time, and logarithmic axes.

Values in *values-list* can be specified as a simple list of values in the following form:

value <...*value-n*>

The values must be listed in either ascending or descending order. The values can also be specified as a range of values of a specific increment in the following form:

value-1 TO *value-2* BY *increment-value*

By default the increment value is 1. You can use a negative integer for increment to specify a value list in descending order.

You can use both forms when specifying values:

<*value* ... *value-n* >*value-1* TO *value-2* BY *increment-value* <*value* ... *value-n*>

In all forms, multiple *n* values can be separated by blanks or commas. Here are some examples:

- Place tick marks at 2, 4, and 6 on a linear axis:

```
values=(2 4 6)
```

- Place tick marks at 6, 4, and 2 on a linear axis:

```
values=(6,4,2)
```

- Place tick marks at 25MAY08, 04JUL08, and 23AUG08 on a time axis:

```
values=("25MAY08"d "04JUL08"d "23AUG08"d)
```

- Place tick marks from 2 to 10 at an increment of 2 on a linear axis:

```
values=(2 to 10 by 2)
```

- Place tick marks from 50 to 10 at an increment of -5 on a linear axis:

```
values=(50 to 10 by -5)
```

If the specified range is not evenly divisible by the increment value, the highest value displayed on the axis is the last incremental value below the ending value for the range. For example, this value list produces a maximum axis value of 9:

```
values=(0 to 10 by 3)
```

Restrictions This option has no effect on discrete axes.

If your VALUES= option creates more than 1000 values, then the option has no effect.

Interactions For logarithmic axes, use the LOGVTYPE option to control whether the values are expanded or interpreted as exponents.

If a custom format is applied to the value, the raw value is plotted first and then formatted.

Tips Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

For graphs such as histograms, the final bin might be truncated at the maximum data value. If you need to expand the axis in order to show all of the data, use the VALUESHINT option.

"string-list"

specifies a space-separated list of string values enclosed in parentheses for a discrete axis .

Each value in the list must be enclosed in quotation marks. Numeric values must also be enclosed in quotation marks.

Only the tick values that are included in the string list are displayed on the axis. The values are displayed in the order in which they are listed. The data values that are not in the list are dropped. The list can be a subset of the data values. It can also contain values that are not included in the actual data. A tick value that is not included in the data appears on the axis, but no data is represented at its tick mark.

Note: For an example that is used in the SGPLOT procedure, see Table 10.140 on page 1547.

Restrictions This option has no effect on linear, time, or logarithmic axes.

Each value must be enclosed in quotation marks and separated from adjacent values by a blank space.

Notes If the string list contains duplicate values, the first occurrence of the duplicated value in the list is honored while the remaining instances are ignored.

The axis data can be character or numeric.

If a numeric variable has an associated format, the specified values must use the same format.

Tip Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

Examples The following example specifies the axis tick values Sedan, Sports, Wagon, and SUV:

```
values=("Sedan" "Sports" "Wagon" "SUV")
```

The following example specifies the axis tick values 10, 20, 30, and 40:

```
values=("10" "20" "30" "40")
```

The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

TIP Along the axis, numeric tick values are arranged in ascending order while character values are arranged in the order in which they are used in the graph. In some cases, the resulting tick-value order might not be desirable, especially if the graph consists of multiple plots or if the data contains missing values. This option is useful in that case. You can use this option to set the order of the axis tick values.

VALUESDISPLAY=("*value1*" "*value2*" ...)

specifies the text that is to be displayed for the tick values that are defined in the VALUES= option. The list of values must be enclosed in parentheses. Each value must be enclosed in quotation marks and separated from adjacent values by a blank space. Numeric values must also be enclosed in quotation marks.

Restriction This option applies only to discrete and linear axes.

Interaction This option should be used with the VALUES= option. The number of items in the list for this option should equal the number of items in the list for the VALUES= option.

Example The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

VALUESFORMAT=DATA | *SAS-format*

specifies how to format the values for major tick marks. This option supports discrete and logarithmic axes.

DATA

uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

SAS-format

specifies a format to apply to the major tick values.

Restriction This option currently honors most, but not every, SAS format. For more information, see "Format Support in ODS Graphics" in *SAS Graph Template Language: Reference*.

Notes If you specify a format that significantly reduces precision, then, because of tick-value rounding, the plot data elements might not align properly with the axis tick values. In that case, specify a tick-value format with a higher precision.

A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F5.1

Alias TICKVALUEFORMAT

Restriction For discrete axes, only character formats are supported.

Interactions This option is ignored when LOGSTYLE=LOGEXPONENT.

When LOGSTYLE =LOGEXPAND, this option is honored for the base 10 and base 2 logarithmic scales, and is ignored for the base E scale.

When LOGSTYLE =LINEAR, this option is honored for the base 10, base 2, and base E logarithmic scales.

VALUESHALIGN=LEFT | CENTER | RIGHT

specifies the horizontal alignment for all of the tick values that are displayed on the axis.

Default RIGHT

Interaction This option is ignored when FITPOLICY= is specified as either SPLIT or SPLITALWAYS. To align split tick mark values, use the SPLITJUSTIFY= option.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option. Tick

values specified by the VALUES= option are treated as suggestions for the tick values that actually appear, so that the actual data range of the axis is preserved. If the *values-list* contains values that are less than the data minimum, or greater than the data maximum, those values are ignored.

Interaction This option has no effect unless you also specify the VALUES= option.

ROWAXISTABLE Statement

Creates a basic plot that displays data values at specific locations along the vertical axis.

Requirements: At least one other plot statement must be used in the procedure. This other plot becomes the primary plot. The primary plot provides the default placement of values along the axis table's vertical axis.

For the axis table, you must specify at least one variable. This variable determines which values are displayed in the axis table.

Interactions: You can specify a variable for the Y= option to identify the locations along the axis where the values are displayed. If you do not specify Y=, the locations are determined by the Y variable in the primary plot. When the primary plot is a categorical chart (bar, line, or dot), the category variable from the chart is used. With the exception of that Y (or category) variable and any group variable that might be specified, axis tables are separate plots and are unaware of the options specified in the accompanying plots.

When used with bar charts, line charts, and dot plots, all axis tables must align with the category axis of the chart. If a statement uses the wrong orientation, the statement is rejected with a message in the SAS log. For example, if your procedure has a VBAR statement along with a ROWAXISTABLE statement, the ROWAXISTABLE is rejected with a message.

Axis tables cannot be used when the primary plot is one of the following plot types: BAND, BLOCK, FRINGE, REG, LOESS, and PBSPLINE. The axis table requires at least one other plot statement that is not one of these types. Otherwise, the axis table is not created and an error is written to the log.

Note: You can include multiple ROWAXISTABLE statements in an SGPANEL procedure step.

Syntax

ROWAXISTABLE *variable* <*...variable-n*> </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORGROUP=*variable*
: specifies a variable that is used to determine the color of the table values.

DROPONMISSING
: specifies that the entire axis table is dropped when all of the values are missing.

NOMISSINGCHAR
: suppresses the display of the MISSING character (.) for missing numeric values.

PAD=*numeric-value<units>* | (*pad-options*)
: specifies the amount of extra space that is added inside the table border.

POSITION=LEFT | RIGHT
: specifies the position of the axis table at the left or right side of the graph cell.

SEPARATOR
: creates a separating line between the axis table or axis tables and the plot.

TEXTGROUP=*attribute-map-group-variable*
: specifies the group variable that is used in a discrete attribute map data set to map text attributes to values for each observation.

TEXTGROUPID=*attribute-map-id*
: specifies an attribute ID for the TEXTGROUP= option.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the axis table values.

VALUEHALIGN=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the column values relative to the column width in the axis table.

VALUEJUSTIFY=LEFT | CENTER | RIGHT
: specifies the justification of the column values relative to the column width in the axis table.

Class options

CLASS=*variable*
: creates a separate axis table for each unique value of the specified variable.

CLASSDISPLAY=STACK | CLUSTER
: specifies how the class values are displayed.

CLASSORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the order in which the class values are displayed.

NOMISSINGCLASS
: specifies that missing values of the class variable are not included in the table.

Label options

LABEL

LABEL="*text-string*"

NOLABEL
specifies whether the table label is shown or hidden.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the color and font attributes of the axis table label.

LABELHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the column labels.

LABELJUSTIFY=LEFT | CENTER | RIGHT
specifies the justification of the labels, when displayed.

LABELPOS=BOTTOM | TOP
specifies the position of the labels at the bottom or top of the axis table.

STATLABEL

NOSTATLABEL
specifies whether the variable statistic is displayed in the table's label.

Plot options

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the axis table.

Y=*variable*
Specifies the Y variable to use to align the table values to the Y axis.

Plot reference options

NAME="*text-string*"
specifies a name for the plot.

Title options

TITLE="*text-string*"
specifies a title for the axis table.

TITLEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the title for the axis table.

TITLEHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the axis table title bounding box relative to the axis table width.

TITLEJUSTIFY=LEFT | CENTER | RIGHT
specifies the justification of the lines of a multiline title in the title bounding box. The justification is relative to the axis table width.

Required Argument

variable <…variable-n>
specifies one or more variables for the axis table. When multiple variables are specified, the axis tables are placed in columns. Any options that you add to the statement apply to all the variables that are specified in that statement.

Note When the variable specified is a character variable, the first value of each category is displayed in the axis table. When the variable is numeric, the axis table displays the sum statistic. You can modify the statistic using the STAT= option.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

CLASS=*variable*
creates a separate axis table for each unique value of the specified variable. This option acts as a classification variable for the axis table. Each axis table is labeled by the class value.

Restriction The CLASS option is ignored when the axis table is used with bar, line, or dot charts. If the GROUP= option is specified in the chart, that group variable is used as the CLASS variable for all axis tables.

Interaction If NOLABEL is also specified, then the class labels are removed.

Tips Use the CLASSDISPLAY= option to control whether the class values are clustered or stacked.

Use the CLASSORDER= option to control the order in which the class values are displayed.

CLASSDISPLAY=STACK | CLUSTER
specifies how the class values are displayed.

STACK
displays the class values horizontally at each midpoint value on the Y axis.

CLUSTER
displays the class values vertically at each midpoint value on the Y axis.

Restriction The CLASSDISPLAY=CLUSTER option applies only when the axis table is on a discrete axis.

Default STACK

Interaction For this option to have any effect, the CLASS= option must be specified.

CLASSORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the order in which the class values are displayed.

DATA
displays the class values in the order in which they occur in the data.

REVERSEDATA
displays the class values in the reverse order from which they occur in the data.

Tip This option is useful when the plot axis is reversed.

ASCENDING
displays the class values in ascending order.

DESCENDING
displays the class values in descending order.

Default DATA

Interactions This option is ignored when the axis table is used with DOT, HBAR, VBAR, HLINE, and VLINE statements. If the GROUPORDER= option is specified in the chart, that group order variable is used as the class order variable for all axis tables.

For this option to have any effect, the CLASS= option must be specified.

COLORGROUP=*variable*
specifies a variable that is used to determine the color of the table values. Once the variable values are found, the value colors are taken from the GraphData1–GraphData*n* style elements in the current style. The CONTRASTCOLOR attribute is used for the value text.

Restriction When used with DOT, HBAR, VBAR, HLINE, and VLINE statements, this option has no effect unless *variable* is used in the accompanying chart.

Note This option is used only to color the table values. If you want to set additional text attributes, used the TEXTGROUP= option instead.

DROPONMISSING
specifies that the entire axis table is dropped when all of the values are missing. Consider using this option if the SAS log indicates that the specified data column used for the axis table is missing all values.

LABEL
LABEL="*text-string*"
NOLABEL
specifies whether the table label is shown or hidden. If you specify LABEL=, then you can also specify a text string for the label.

Defaults LABEL

If you do not specify a text string, then the variable name is used for the label. Or, if CLASS= is also specified, then the unique values of the specified class variable are used for the labels.

Tip Use the LABELATTRS= option to modify the label text attributes. Use the LABELPOS= option to move the label.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the color and font attributes of the axis table label. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

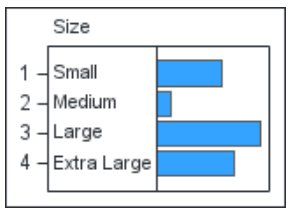
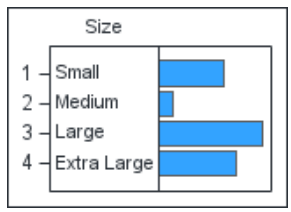
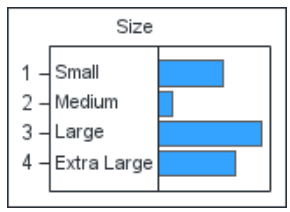
Restriction Group behavior occurs only when the CLASS= and COLORGROUP= option values are the same.

Interactions This option has no effect if NOLABEL is also specified.

If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphValueText style element.

LABELHALIGN=LEFT | CENTER | RIGHT

specifies the horizontal alignment of the column labels.

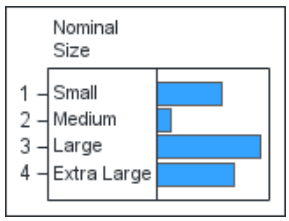
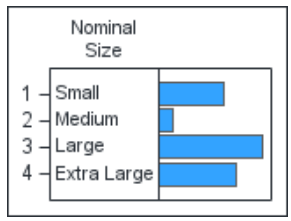
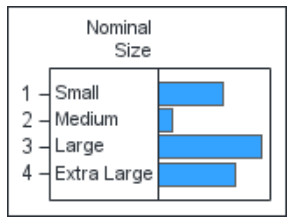
LEFT	CENTER	RIGHT
		

Default Uses the effective value of the LABELJUSTIFY= option.

Interaction This option has no effect if NOLABEL is also specified.

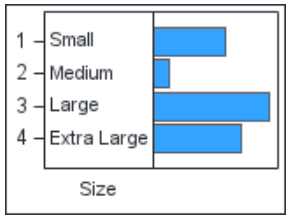
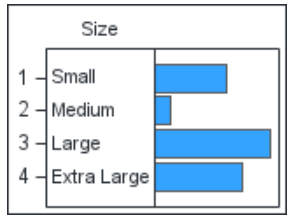
LABELJUSTIFY=LEFT | CENTER | RIGHT

specifies the justification of the labels, when displayed.

LEFT	CENTER	RIGHT
		

Default Uses LEFT for text values or RIGHT for numeric values.

LABELPOS=BOTTOM | TOP
specifies the position of the labels at the bottom or top of the axis table.

BOTTOM	TOP
	

Default BOTTOM

NAME="*text-string*"
specifies a name for the plot. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGCHAR
suppresses the display of the MISSING character (.) for missing numeric values. Missing numeric values are displayed as blanks.

NOMISSINGCLASS
specifies that missing values of the class variable are not included in the table.

Interaction For this option to have any effect, the CLASS= option must be specified.

PAD=*numeric-value<units>* | (*pad-options*)
specifies the amount of extra space that is added inside the table border. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

numeric-value<units>
specifies a dimension to use for the extra space at the table border.

(*pad-options*)
specifies a space-separated list of pad options, enclosed in parentheses.

Specify one or more of the following name-value-pair options:

LEFT=*numeric-value<units>*
specifies the amount of extra space added to the left side.

Default 4 px

RIGHT=*numeric-value<units>*
specifies the amount of extra space added to the right side.

Default 4 px

TOP=*dimension*
specifies the amount of extra space added to the top.

Default 0 px

BOTTOM=*dimension*
specifies the amount of extra space added to the bottom.

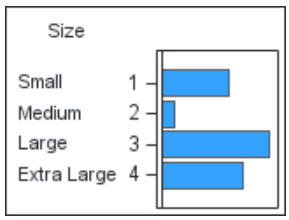
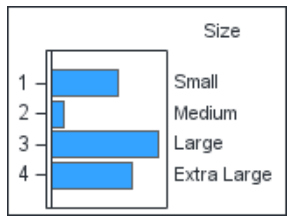
Default 0 px

Note Sides that are not assigned padding are padded with the default amount of space.

Tip Use *pad-options* to create non-uniform padding.

Note The default units for *dimension* are pixels. If you want to specify values in other units, then you must specify the desired units with the value. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

POSITION=LEFT | RIGHT
specifies the position of the axis table at the left or right side of the graph cell.

LEFT	RIGHT
	

Default RIGHT

SEPARATOR
creates a separating line between the axis table or axis tables and the plot.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the axis table. The statistic is applied only to numeric variables. For non-categorical plots, only the SUM and MEAN statistics are available.

Specify one of the following:

FREQ
the frequency of the axis table variable.

Interactions For this value to take effect, the graph must use a categorical plot (bar chart, line chart, or dot plot).

The FREQ value has a different effect when the axis table variable is the same as the categorical variable of the primary plot. In that case, the axis table variable is treated as a category (class) variable instead of a response (analysis) variable when computing the frequency. The FREQ value is the frequency of the categorical variable.

MEAN
the mean of the axis table variable.

MEDIAN
the median of the axis table variable.

Interactions For this value to take effect, the graph must use a categorical plot (bar chart, line chart, or dot plot).

This value is ignored when the axis table variable is the same as the categorical variable of the primary plot. In that case, the FREQ statistic is used and a message is written to the SAS log.

PERCENT
the percentage of the sum of the axis table variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions For this value to take effect, the graph must use a categorical plot (bar chart, line chart, or dot plot).

The PERCENT value has a different effect when the axis table variable is the same as the categorical variable of the primary plot. In that case, the axis table variable is treated as a category (class) variable instead of as a response (analysis) variable when computing the percentage. Therefore, the PERCENT value is calculated as a percentage of frequency instead of a percentage of the sum.

The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the PCTLEVEL= option in the PROC SGPANEL statement.

You can use the PCTNDEC= option in the PROC SGPANEL statement to control the number of decimals to be used when calculating the percent values. The default value is 1.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level will be zero.

SUM
the sum of the axis table variable.

Defaults SUM

When used with a categorical plot and the axis table variable is the same as the categorical variable, FREQ is the default.

Restrictions The STAT= option is applied only to numeric variables.

Only SUM and MEAN are supported for non-categorical plots.

When used with a categorical plot and the axis table variable is the same as the categorical variable, only FREQ and PERCENT are supported.

Interaction Any STAT= value specified in the primary plot has no effect on the axis table statistic.

STATLABEL | NOSTATLABEL
specifies whether the variable statistic is displayed in the table's label. STATLABEL forces the statistic to be displayed in the label. NOSTATLABEL removes the statistic from the label.

Defaults The statistic is displayed for the variable.

When a custom label is assigned to the variable, the statistic is not displayed.

Interaction This option has no effect unless the STAT= option is also specified in the axis table statement.

TEXTGROUP=*attribute-map-group-variable*
specifies the group variable that is used in a discrete attribute map data set to map text attributes to values for each observation. You specify this option only if you are using an attribute map to control visual attributes of the graph. The variable's values must correspond to the values in the VALUE variable in the attribute map data set. For more information, see Chapter 18, "Using Discrete Attribute Maps," on page 1901.

Requirement The TEXTGROUPID= option specifies the attribute ID to use for the attribute mapping. If TEXTGROUPID= is not specified, then the ATTRID= option is used. If the ATTRID= option is also not specified, then the TEXTGROUP option is ignored.

Interaction When this option is specified, the COLORGROUP= option is ignored.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

TEXTGROUPID=*attribute-map-id*
specifies an attribute ID for the TEXTGROUP= option.

Default If neither TEXTGROUPID= nor ATTRID= is specified, the TEXTGROUP= option is ignored.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

TITLE="*text-string*"
specifies a title for the axis table. It the axis table statement specifies more than one variable, the title is displayed for each variable.

Tip Use the TITLEATTRS= option to modify the title text attributes.

TITLEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the title for the axis table. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attribute of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attribute of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interactions This option has no effect unless TITLE= is also specified.

If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphDataText style element.

Examples
```
TITLEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TITLEATTRS=GraphTitleText
```

TITLEHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the axis table title bounding box relative to the axis table width. This option applies only to row (Y) axis tables.

Default LEFT

Interaction This option takes effect only when TITLE= is also specified.

Tip For a multiline title, use TITLEJUSTIFY= to justify the title text lines in the title bounding box.

TITLEJUSTIFY=LEFT | CENTER | RIGHT
specifies the justification of the lines of a multiline title in the title bounding box. The justification is relative to the axis table width. This option applies only to row (Y) axis tables.

Default LEFT

Restriction The title must wrap to multiple lines for this option to have any effect.

Interaction This option takes effect only when TITLE= is also specified.

Tip Use TITLEHALIGN= to align the title bounding box.

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis table values. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

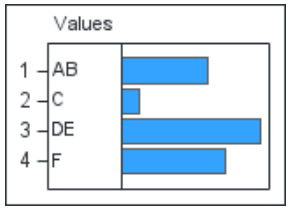
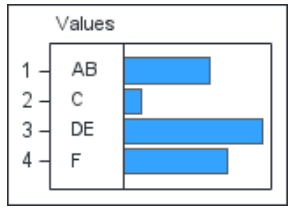
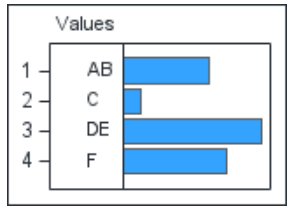
For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphDataText style element.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

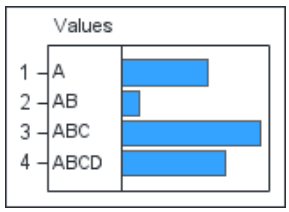
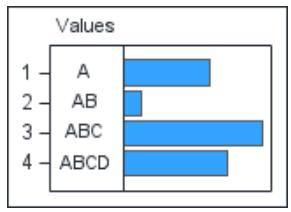
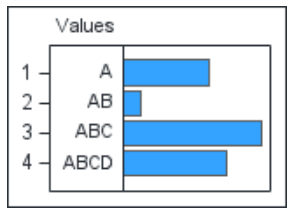
Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

VALUEHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the column values relative to the column width in the axis table. This option aligns the text within the column based on the longest string.

LEFT	CENTER	RIGHT
		

Default Uses the effective value of the VALUEJUSTIFY= option.

VALUEJUSTIFY=LEFT | CENTER | RIGHT
specifies the justification of the column values relative to the column width in the axis table. This option horizontally aligns the table values center, left, or right relative to the column width.

LEFT	CENTER	RIGHT
		

Default LEFT for text values or RIGHT for numeric values.

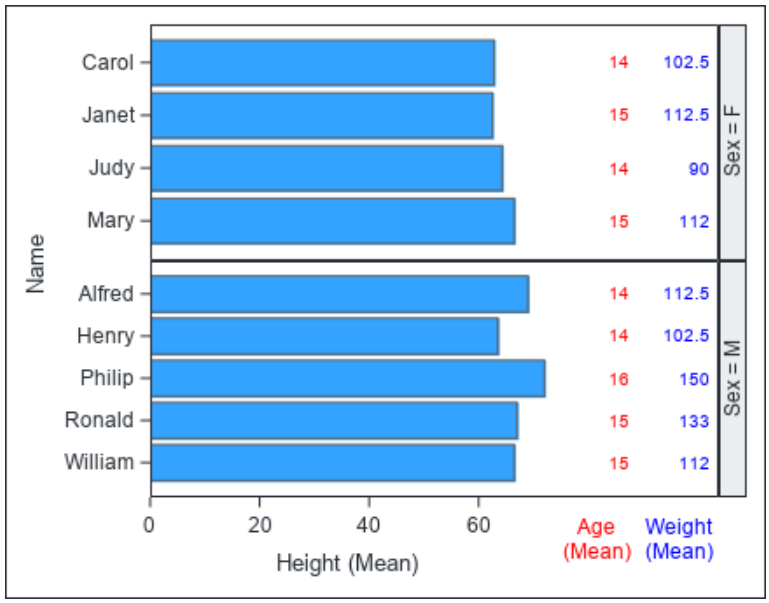
Y=*variable*
Specifies the Y variable to use to align the table values to the Y axis.

Default When this option is not specified, the Y variable of the primary plot is used by default.

Interaction This option is ignored when the axis table is used with a categorical chart (bar, line, or dot). The category variable from the chart is used instead.

Example: ROWAXISTABLE Statements

This example shows two axis tables along the Y axis of a bar chart that charts student height.



Here are the noteworthy features about the axis tables:

- The first axis table statement displays the age (in red) of each student.
- The second axis table statement specifies the weight. Both axis tables are located to the right of the Y axis by default.

Here is the SAS code for this example.

```
ods graphics / width=4.5in height=3.5in;

proc sgpanel data=sashelp.class (where=(age > 13));
  panelby sex / layout=rowlattice uniscale=column;
  hbar name / response=height stat=mean;
  rowaxistable age / stat=mean
       valueattrs=(color=red)
       labelattrs=(color=red);
  rowaxistable weight / valueattrs=(color=blue)
            stat=mean
       labelattrs=(color=blue);
run;
```

For an example that shows horizontal axis tables along the X axis, see “Example: COLAXISTABLE Statements” on page 787.

ATTRIB Statement

Associates a format, informat, label, and length with one or more variables.

Type: Procedure statement

Syntax

ATTRIB *variable-list(s) attribute-list(s)*;

Arguments

variable-list(s)
names the variables that you want to associate with the attributes.

> **TIP** List the variables in any form that SAS allows.

attribute-list(s)
specifies one or more attributes to assign to *variable-list*. Specify one or more of these attributes in the ATTRIB statement:

FORMAT=*format*
associates a format with variables in *variable-list*.

Tip The format can be either a standard SAS format or a format that is defined with the FORMAT procedure.

INFORMAT=*informat*
associates an informat with variables in *variable-list*.

Tip The informat can be either a standard SAS informat or an informat that is defined with the FORMAT procedure.

LABEL='*label*'
associates a label with variables in *variable-list*.

See Also

"ATTRIB Statement" on page 142 for more information about the ATTRIB statement.

BY Statement

Orders the output according to the BY groups.

Type: Procedure statement

Syntax

BY <DESCENDING> *variable-1*

<-<DESCENDING> *variable-n*>
<NOTSORTED>;

Required Argument

variable
specifies the variable that the procedure uses to form BY groups. You can specify more than one variable. If you do not use the NOTSORTED option in the BY statement, then either the observations in the data set must be sorted by all the variables that you specify, or they must be indexed appropriately. Variables in a BY statement are called *BY variables*.

Optional Arguments

DESCENDING
specifies that the observations are sorted in descending order by the variable that immediately follows the word DESCENDING in the BY statement.

NOTSORTED
specifies that observations are not necessarily sorted in alphabetic or numeric order. The observations are grouped in another way (for example, chronological order).

The requirement for ordering or indexing observations according to the values of BY variables is suspended for BY-group processing when you use the NOTSORTED option. In fact, the procedure does not use an index if you specify NOTSORTED. The procedure defines a BY group as a set of contiguous observations that have the same values for all BY variables. If observations with the same values for the BY variables are not contiguous, then the procedure treats each contiguous set as a separate BY group.

You cannot use the NOTSORTED option in a PROC SORT step.

Note: You cannot use the GROUPFORMAT option, which is available in the BY statement in a DATA step, in a BY statement in any PROC step.

See Also

"BY Statement" on page 143 for more information about the BY statement.

FORMAT Statement

Associates formats with variables.

Type: Procedure statement

Syntax

FORMAT *variable-1* <-*variable-n*> <*format-1*> <*variable-1* <-*variable-n*> <*format-2*> > - <DEFAULT=*default-format*>;

Required Argument

variable
names one or more variables for SAS to associate with a format. You must specify at least one *variable*.

Tip To disassociate a format from a variable, use the variable in a FORMAT statement without specifying a format in a DATA step or in PROC DATASETS. In a DATA step, place this FORMAT statement after the SET statement. See "Removing a Format" in *SAS DATA Step Statements: Reference*. You can also use PROC DATASETS.

Optional Arguments

format
specifies the format that is listed for writing the values of the variables.

Tip Formats that are associated with variables by using a FORMAT statement behave like formats that are used with a colon modifier in a subsequent PUT statement. For more information about using a colon modifier, see "PUT Statement: List" in *SAS DATA Step Statements: Reference*.

See *SAS Formats and Informats: Reference*

DEFAULT=*default-format*
specifies a temporary default format for displaying the values of variables that are not listed in the FORMAT statement. These default formats apply only to the current DATA step; they are not permanently associated with variables in the output data set.

A DEFAULT= format specification applies to

- variables that are not named in a FORMAT or ATTRIB statement
- variables that are not permanently associated with a format within a SAS data set
- variables that are not written with the explicit use of a format.

Default If you omit DEFAULT=, SAS uses BEST*w*. as the default numeric format and $*w*. as the default character format.

Restriction Use this option only in a DATA step.

Tip A DEFAULT= specification can occur anywhere in a FORMAT statement. It can specify either a numeric default, a character default, or both.

Example "Assigning Formats and Defaults" in *SAS DATA Step Statements: Reference*

See Also

"FORMAT Statement" on page 146 for more information about the FORMAT statement.

LABEL Statement

Assigns descriptive labels to variables.

Type: Procedure statement

Syntax

Form 1: Creates a descriptive label for one or more variables.

LABEL *variable-1*='*text-string*' <...*variable-n*='*text-string*>;

Form 2: Removes the label from one or more variables.

LABEL *variable-1* = ' ' < ...*variable-n*=' '>; | *variable-1* = <...*variable-n*= >

Arguments

variable

specifies the variable that you want to label. You can specify labels for multiple variables in a single LABEL statement:

```
data test;
   L = 5;
   W = 10;
   label L = 'Length' W = 'Width';
run;
```

text-string

specifies a label of up to 256 bytes.

```
data test;
   L = 5;
   label L = 'Length';
run;
```

Restrictions If the label includes a semicolon (;) or an equal sign (=), you must enclose the label in either single or double quotation marks.

```
label N = 'Number = ';
```

If the label includes single quotation marks (’), you must enclose the label in double quotation marks.

```
label Name = "Owner's Last Name";
```

There is partial support for variable name substitution with the JAVAIMG device. When the label text for the variable exceeds the space allowed, the variable name is used instead by procedures that are labeling an axis or a legend title. With the exception of the SAS/GRAPH GPLOT procedure, the variable name is not used when the JAVAIMG device is specified.

Notes

Quotation marks are not required unless the label includes a semicolon, quotation marks, or an equal sign.

The LABEL statement expects an equal sign (=) after the first variable. If any other character is found, an error occurs. The LABEL statement considers everything after the equal sign, following the first variable, as part of the label until the statement comes to another variable followed by an equal sign, regardless of quotation marks. In this example, the label for x is **`this is x y 4 = this is y`** because the next variable that is followed by an equal sign is z.

```
data test;
   x = 1;
   y = 1;
   z = 1;
   label x = 'this is x' y 4 = 'this is y' z = 'This is z';
run;
```

In this LABEL statement, the x variable has the same label:

```
label x = this is x y 4 = this is y  z = This is z;
```

See

“Character Constants and Character Variables” in *SAS Programmer’s Guide: Essentials*.

’’

removes a label from a variable. Enclose a single blank space in quotation marks to remove an existing label. You can remove labels from multiple variables in a single LABEL statement:

```
data test;
   L=5; W=10;
   label L = ' '  W = ' ';
run;
```

Alternatively, you can remove a label by specifying nothing after the equal sign:

```
data test;
   L=5; W=10;
   label L =  W = ;
run;
```

See Also

"LABEL Statement" on page 150 for more information about the LABEL statement.

WHERE Statement

Subsets the input data set by specifying certain conditions that each observation must meet before it is available for processing.

Type: Procedure statement

Syntax

WHERE *where-expression*;

Required Argument

where-expression
is an arithmetic or logical expression that generally consists of a sequence of operands and operators. For more information about where processing, see *SAS DATA Step Statements: Reference*.

Tip You can specify multiple where-expressions.

See Also

"WHERE Statement" on page 151 for more information about the WHERE statement.

Examples: SGPANEL Procedure

About the SGPANEL Procedure Examples

Examples in This Section

This section provides basic examples of how to use the SGPANEL procedure. The following examples are provided:

Example	Description
"Example 1: Creating a Panel of Graph Cells with Histograms and Density Plots" (p. 832)	Shows how to use the HISTOGRAM and DENSITY statements to create a panel of graph cells with histograms and density plots.
"Example 2: Creating a Panel of Regression Curves" (p. 833)	Shows how to use the PANELBY and REG statements to create a three-column panel of regression curves.
"Example 3: Creating a Panel of Bar Charts" (p. 835)	Shows how to use the PANELBY and HBAR statements to create a one-column panel of bar charts.
"Example 4: Creating a Panel of Line Charts" (p. 836)	Shows how to use the PANELBY and VLINE statements to create a panel of line plots with grouped data.

Additional Examples in This Book

Additional examples exist elsewhere in this book for most of the SGPANEL procedure statements. The following table lists these examples by statement.

Statement	Reference
BAND	"About Band Plots" (p. 17)
BLOCK	"About Block Plots" (p. 18)
BUBBLE	"About Bubble Plots" (p. 19)
COLAXISTABLE	"Example" (p. 787)
DENSITY	"About Density Plots" (p. 49)
DOT	"About Dot Plots" (p. 60)
DROPLINE	"About Drop Lines" on page 26 "Example" on page 251
ELLIPSE	"About Ellipse Plots" (p. 42)
FRINGE	"About Fringe Plots" (p. 20)
GRADLEGEND	"Example" (p. 276)
HBAR HBARBASIC HBARPARM	"About Bar Charts" (p. 53)
HBOX	"About Box Plots" (p. 47)
HEATMAP HEATMAPPARM	"About Heat Maps" (p. 21)
HIGHLOW	"About High-Low Plots" (p. 23)
HISTOGRAM	"About Histograms" (p. 51)
HLINE	"About Line Charts" (p. 61)
INSET	"About Text Insets" on page 38 "Example" on page 429
KEYLEGEND	"Example" (p. 438)
LEGENDITEM	"Example" (p. 443)
LINEPARM	"About Parameterized Lines" (p. 27)

Statement	Reference
LOESS	"LOESS Statement" (p. 451)
NEEDLE	"About Needle Plots" (p. 29)
PBSPLINE	"About Penalized B-Spline Plots" (p. 44)
POLYGON	"Example" (p. 512)
REFLINE	"About Reference Lines" (p. 25)
REG	"About Regression Plots" (p. 45)
ROWAXISTABLE	"Example" (p. 821)
SCATTER	"About Scatter Plots" (p. 31)
SERIES	"About Series Plots" (p. 33)
SPLINE	"About Spline Plots" (p. 34)
STEP	"About Step Plots" (p. 35)
STYLEATTRS	"Example" (p. 187)
SYMBOLCHAR	"Example 1: Simple SYMBOLCHAR Example" on page 610 "Example 2: SYMBOLCHAR Used with an SG Attribute Map" on page 611
SYMBOLIMAGE	"Example 1: Simple SYMBOLIMAGE Example" on page 616 "Example 2: SYMBOLIMAGE Used with an SG Attribute Map" on page 616
TEXT	"About Text Plots" (p. 36)
VBAR VBARBASIC VBARPARM	"About Bar Charts" (p. 53)
VBOX	"About Box Plots" (p. 47)
VECTOR	"About Vector Plots" (p. 39)
VLINE	"About Line Charts" (p. 61)

Examples Available Online

You can find more examples on the Graphically Speaking blog. On the blog page, search for SGPANEL in the **Search Graphically Speaking** search box to find the SGPANEL procedure examples.

You can also find examples on support.sas.com at the following URL: https://support.sas.com/sassamples/graphgallery/PROC_SGPANEL.html.

Example 1: Creating a Panel of Graph Cells with Histograms and Density Plots

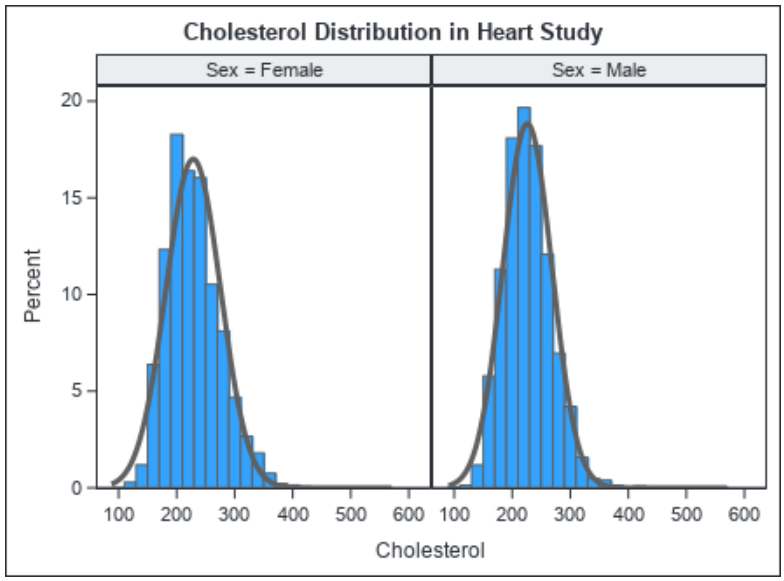
Features: HISTOGRAM statement
DENSITY statement

Sample library member: SGPNHST

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a panel of graph cells with histograms and density plots.

Output



Program

```
proc sgpanel data=sashelp.heart noautolegend;
  title "Cholesterol Distribution in Heart Study";

  panelby sex;

  histogram cholesterol;
  density cholesterol;
run;

title;
```

Program Description

Create the panel and specify the title.

```
proc sgpanel data=sashelp.heart noautolegend;
  title "Cholesterol Distribution in Heart Study";
```

Specify the classification variable for the panel.

```
  panelby sex;
```

Create the histogram and density plots.

```
  histogram cholesterol;
  density cholesterol;
run;
```

Cancel the title.

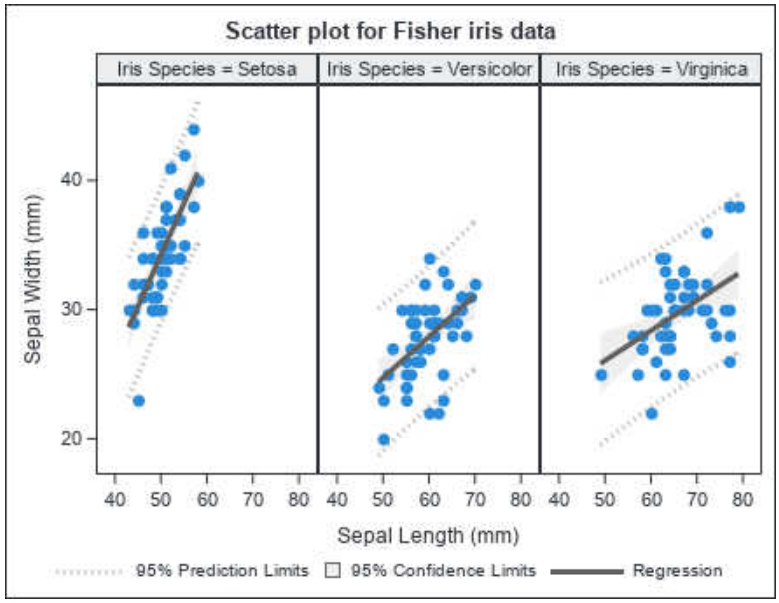
```
title;
```

Example 2: Creating a Panel of Regression Curves

Features:	REG statement
Sample library member:	SGPNREG
Note:	For information about the SAS Sample Library, see “About the SASHELP Data Library” on page 88.

This example shows a three-column panel of regression curves.

Output



Program

```
proc sgpanel data=sashelp.iris;
  title "Scatter plot for Fisher iris data";

  panelby species / columns=3;


  reg x=sepallength y=sepalwidth / cli clm;
run;

title;
```

Program Description

Create the panel and specify the title.

```
proc sgpanel data=sashelp.iris;
  title "Scatter plot for Fisher iris data";
```

Specify the classification variable for the panel. The COLUMNS= on page 170 option in the PANELBY statement specifies three columns in the panel.

```
  panelby species / columns=3;
```

Create the regression curve. The CLI on page 523 option creates individual predicted value confidence limits.The CLM on page 524 option creates mean value confidence limits.

```
  reg x=sepallength y=sepalwidth / cli clm;
run;
```

Cancel the title.

```
title;
```

Example 3: Creating a Panel of Bar Charts

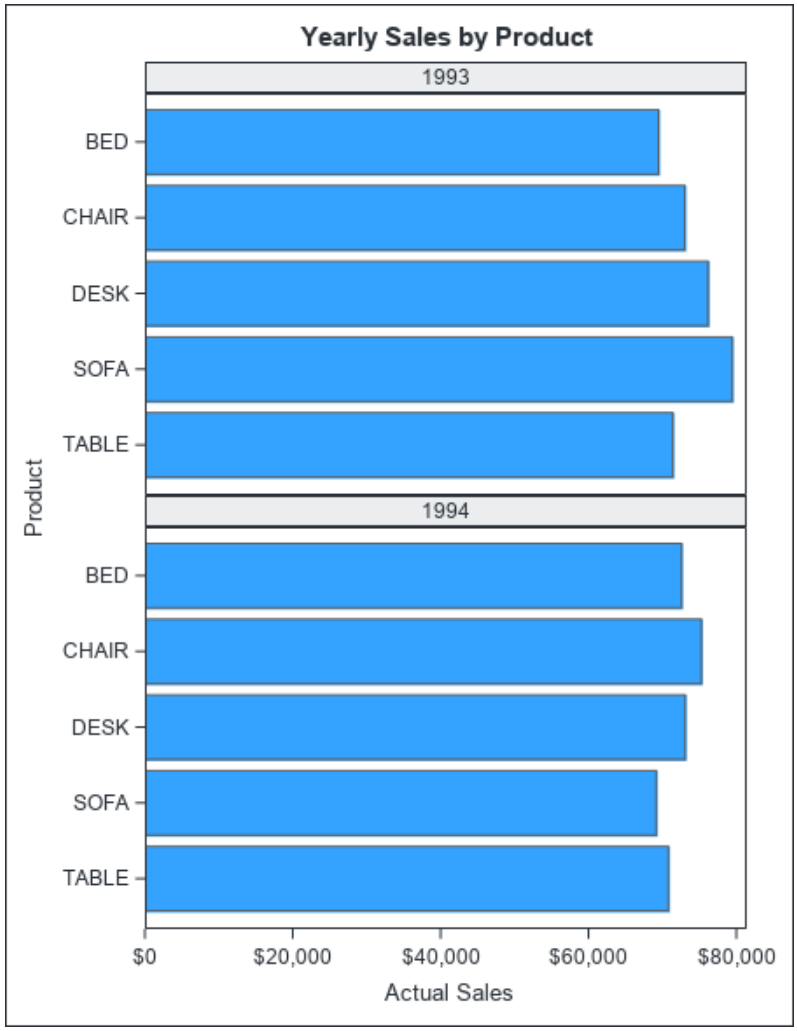
Features: HBAR statement

Sample library member: SGPNBAR

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a one-column panel of bar charts.

Output



Program

```
proc sgpanel data=sashelp.prdsale;
  title "Yearly Sales by Product";

  panelby year / novarname columns=1;

  hbar product / response=actual;
run;

title;
```

Program Description

Create the panel and set the title.

```
proc sgpanel data=sashelp.prdsale;
  title "Yearly Sales by Product";
```

Specify the classification variable for the panel. The NOVARNAME on page 174 option specifies that the variable name is not shown in the heading for each cell.The COLUMNS= on page 170 option in the PANELBY statement specifies the number of columns in the panel.

```
  panelby year / novarname columns=1;
```

Create the horizontal bar chart. The RESPONSE= on page 296 option in the HBAR statement specifies the response variable for the chart.

```
  hbar product / response=actual;
run;
```

Cancel the title.

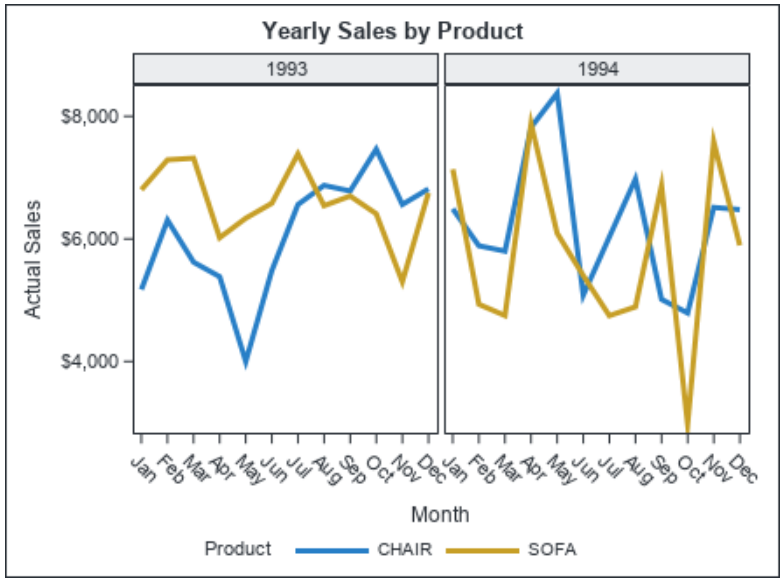
```
title;
```

Example 4: Creating a Panel of Line Charts

Features:	VLINE statement
Sample library member:	SGPNLIN
Note:	For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a panel of line plots with grouped data.

Output



Program

```
proc sgpanel data=sashelp.prdsale;
  where product in ("CHAIR" "SOFA");
  title "Yearly Sales by Product";

  panelby year / spacing=5 novarname;

  vline month / response=actual group=product;
run;

title;
```

Program Description

Create the panel and specify a title.

```
proc sgpanel data=sashelp.prdsale;
  where product in ("CHAIR" "SOFA");
  title "Yearly Sales by Product";
```

Specify the classification variable for the panel. The SPACING= on page 179 option in the PANELBY statement specifies the number of pixels between the panels in the plot.The NOVARNAME on page 174 option specifies that the classification variable name is not shown in the headings for each cell.

```
  panelby year / spacing=5 novarname;
```

Create the vertical line plot. The RESPONSE= on page 748 option in the VLINE statement specifies the response variable.The GROUP= on page 743 option specifies the group variable.

```
  vline month / response=actual group=product;
run;
```

Cancel the title.

```
title;
```

9

SGPIE Procedure: Preproduction

Overview: SGPIE Procedure: Preproduction ***839***
SGPIE Procedure Overview 840

Syntax: SGPIE Procedure: Preproduction ***841***
PROC SGPIE Statement 841
DONUT Statement 843
KEYLEGEND Statement 854
PIE Statement 859
STYLEATTRS Statement 868
ATTRIB Statement 870
BY Statement 871
FORMAT Statement 872
LABEL Statement 874
WHERE Statement 876

Examples: SGPIE Procedure: Preproduction ***877***
About the SGPIE Procedure Examples 877
Example 1: Simple Donut Chart 878
Example 2: Donut Chart with a Response Variable 878
Example 3: Simple Pie Chart 879
Example 4: Pie Chart with a Response Variable 880
Example 5: Pie Chart That Combines Categories into the Other Slice 880
Example 6: Pie Chart with Specified Background Color and Data Colors 881

Overview: SGPIE Procedure: Preproduction

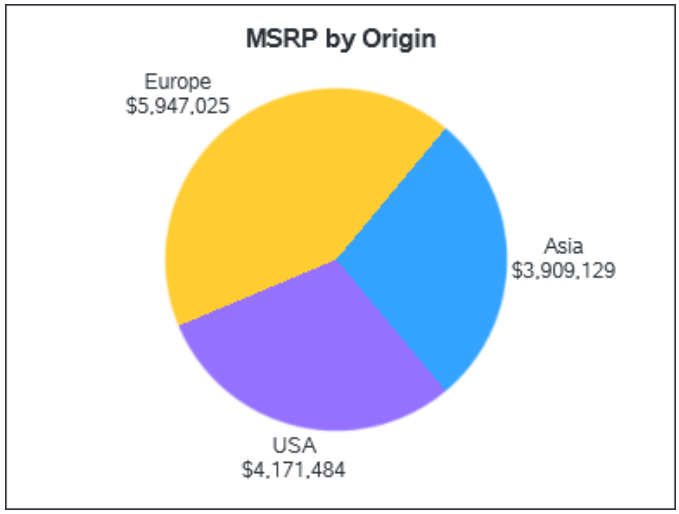
SGPIE Procedure Overview

The SGPIE procedure produces pie charts and donut charts. The angles of pie and donut slices are used to graphically represent the value of a statistic for a data range. The charts are useful for examining how the values of a variable contribute to the whole and for quickly comparing the values of several variables.

Note: The SGPIE procedure is a preproduction feature, which means that it has not been fully developed, tested, or documented.

Here is an example pie chart:

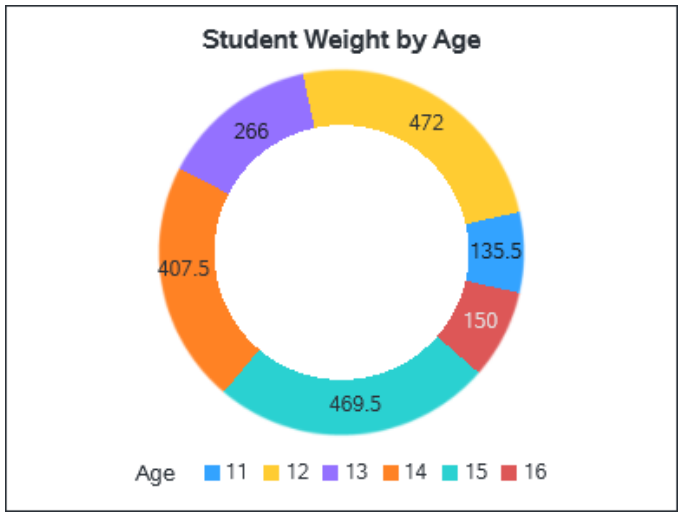
Figure 9.1 *Pie Chart*



See the code for the pie chart example.

Here is a donut chart.

Figure 9.2 Donut Chart



See the code for the donut example.

Syntax: SGPIE Procedure: Preproduction

Restrictions: This procedure is preproduction. See Overview: SGPIE Procedure on page 840.
The SGPIE procedure is supported only on 64-bit Linux systems.

Requirement: At least one plot statement is required.

PROC SGPIE <*options*>;
DONUT *category-variable* </*options*> ;
KEYLEGEND </*options*>;
PIE *category-variable*</*options*>
STYLEATTRS </*options*>
ATTRIB *variable-list(s) attribute-list(s)*;
BY *variable-1*<*...variable-N*><*options*>;
FORMAT *variable-1*<*...variable-N*<*format-1*>><*options*>;
LABEL *variable-1=label-1*<*...variable-n=label-n*>;
WHERE *where-expression*;

PROC SGPIE Statement

Identifies the data set that contains the plot variables. The statement also gives you the option to specify a description, control automatic legends, and specify whether the chart background is opaque or transparent.

Requirement: An input data set is required. If no data set is specified, the procedure uses the most recently created SAS data set.

Syntax

PROC SGPIE <*options*> ;

Summary of Optional Arguments

DATA=*input-data-set*
: specifies the SAS data set that contains the variables to process.

DATTRMAP=*discrete-attribute-map-data-set*
: specifies the discrete attribute map data set that you want to use with the procedure.

DESCRIPTION="*text-string*"
: specifies a description for the output image.

NOAUTOLEGEND
: disables automatic legends from being generated.

OPAQUE

NOOPAQUE
: specifies whether the graph background is opaque or transparent.

Optional Arguments

DATA=*input-data-set*
specifies the SAS data set that contains the variables to process. By default, the procedure uses the most recently created SAS data set.

DATTRMAP=*discrete-attribute-map-data-set*
specifies the discrete attribute map data set that you want to use with the procedure. You specify this option only if you are using a discrete attribute map to control visual attributes of the graph.

Restriction: The procedures do not support an attribute map that is stored in a table. Only standard SAS data sets are supported for attribute map data.

Requirement: The values in the DATTRMAP data set must be sorted by ID. If they are not, only the first value is found.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Example: Use a Discrete Attribute Map with the SGPIE Procedure” on page 1919

“Overview of Attribute Maps” on page 1897

DESCRIPTION="*text-string*"
specifies a description for the output image. The description identifies the image in the following locations:

- the alternate text for the image in HTML output
- the table of contents that is created by the CONTENTS option in an ODS statement

The default description is “The SGPIE Procedure”.

Alias DES=

Notes The name of the output image is specified by the IMAGENAME= option in the ODS GRAPHICS statement.

You can disable the alternate text in HTML output by specifying an empty string. That is, `DESCRIPTION=""`.

NOAUTOLEGEND
disables automatic legends from being generated. By default, legends are created automatically when the CATEGORY labels are not displayed on the slices. This option has no effect if you specify a KEYLEGEND statement.

OPAQUE | NOOPAQUE
specifies whether the graph background is opaque or transparent.

Default OPAQUE

Restriction The following output formats support transparent background (NOOPAQUE): EMF, PDF, PNG, PS, and SVG, with the following exception. The PS format does not support transparent background when your output format is not vector graphics (that is, your output renders as an image due to some graph feature or you used an OUTPUTFMT= override to an image format, including PNG).

Interaction When NOOPAQUE is specified, the background color is not used.

DONUT Statement

Creates a donut chart with slices representing counts or percents of unique category values. If a RESPONSE variable is specified, the chart summarizes the category values over corresponding values of the RESPONSE variable.

Syntax

DONUT *category-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: enhances the visual appearance of the filled donut slices.

DIRECTION=COUNTERCLOCKWISE | CLOCKWISE
: specifies whether to display the donut slices in counterclockwise or clockwise sequence.

FILLPATTERN
: enables the display of line fill patterns for the chart.

NOFILL
: turns off the display of fill colors.

OTHERFILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency of the Other slice.

RINGSIZE=*numeric-value*
: specifies the proportion of the donut radius to be occupied by the donut ring.

STARTANGLE=*degrees*
: specifies which degree between 0 and 360 serves as the starting position for the first slice.

STARTPOS=CENTER | EDGE
: specifies whether the center or the edge of the slice is positioned at the start angle.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Label options

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the color and font attributes of the slice labels.

DATALABELDISPLAY=ALL | NONE | (*content-options*)
: specifies the information to display in the slice labels.

DATALABELLOC=INSIDE | OUTSIDE | CALLOUT
: specifies whether to display the slice labels within the slices or outside of the donut circumference.

HOLELABEL

HOLELABEL="*string*"
: displays a label for the donut hole.

HOLELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies text attributes for the label text in the donut hole.

HOLEVALUE

HOLEVALUE=*number*
displays a numeric value inside the donut hole.

HOLEVALUEATTRS=*style-element*<(***options***)> | (*options*)
specifies text attributes for the value text in the donut hole.

OTHERLABEL="*string*"
specifies a label for the Other slice.

STATFMT=*format-specification*
specifies the format of the displayed statistical value.

Plot options

MAXSLICES=*positive integer*
specifies the maximum number of slices in the donut. Any remaining slices are consolidated into the Other slice.

MISSING
shows the data for missing category values.

OTHERPERCENT=*percent-of-total*
collects all category values with response values less than or equal to *percent-of-total* into the Other slice.

RESPONSE=*numeric-variable*
specifies a numeric variable for the response values.

SLICEORDER=DATA | FORMATTED | UNFORMATTED | RESPASC | RESPDESC
specifies the order of the donut slices.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Required Argument

category-variable
specifies the column for the category values. Duplicated category values are summarized into a unique value. All values are treated as discrete.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. Provide the *character-value* variable without quotation marks.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

Example “Example: Use a Discrete Attribute Map with the SGPIE Procedure” on page 1919

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the color and font attributes of the slice labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Note If data label settings result in labels that do not fit within the slices, the labels are not displayed.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELDISPLAY=ALL | NONE | (*content-options*)
specifies the information to display in the slice labels.

ALL
displays all available information

NONE
does not display slice labels

(*content-options*)
a space-separated list of one or more of the following options enclosed in parentheses:

CATEGORY
displays the category value.

RESPONSE
displays the sum of the response value for each category.

If no response variable is specified, displays the category value.

PERCENT
displays either of the following:

- when a response variable is specified, displays the response sum as a percentage.
- when no response variable is specified, displays the category frequency as a percentage.

Defaults When a response variable is specified, the default display is the category value and the response sum.

When no response variable is specified, the default display is the category value and frequency.

Notes If the category label is suppressed through this option, an automatic legend is generated with the category values.

The position of the labels is adjusted to prevent the labels from overlapping.

If data label settings result in labels that do not fit within the slices, the labels are not displayed.

DATALABELLOC=INSIDE | OUTSIDE | CALLOUT
specifies whether to display the slice labels within the slices or outside of the donut circumference.

INSIDE
locates the slice labels inside the slices.

Note If there is not sufficient space to fit all of the labels, some labels might encroach on adjacent slices and overlap with other labels.

OUTSIDE
locates the slice labels outside of the donut circumference.

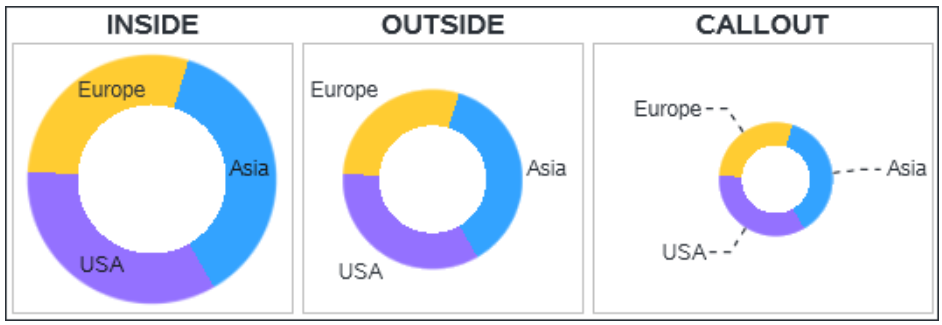
Note The diameter of the donut is reduced in order to fit the labels. In some cases, labels might overlap.

CALLOUT
locates the slice labels outside of the donut circumference and draws a line from the label to its slice.

Each callout line consists of two line segments: a 16-pixel perpendicular segment from the pie circumference, and a 12-pixel segment toward the slice label.

Note The diameter of the donut is reduced in order to fit the labels. The minimum pie size is 33 percent of the available space. If labels do not fit at the minimum size, they are truncated or dropped.

Figure 9.3 *Donut Chart Label Location Examples*



Default Each label is placed using INSIDE or CALLOUT, depending on where the label fits. If some labels do not fit when using CALLOUT, they are truncated or dropped.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
enhances the visual appearance of the filled donut slices.

Default NONE

Restriction The maximum number of skinned graphical elements is limited to 200 per plot. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

DIRECTION=COUNTERCLOCKWISE | CLOCKWISE

specifies whether to display the donut slices in counterclockwise or clockwise sequence.

Default COUNTERCLOCKWISE

Interaction The STARTANGLE= option controls the starting angle for the first donut slice.

FILLPATTERN

enables the display of line fill patterns for the chart.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Note A line fill pattern can be displayed in the OTHER slice as well. The attributes of the OTHER slice fill pattern are controlled by the current ODS style.

Accessibility notes This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the NOFILL option to suppress the chart's fill color.

You can use the DATAFILLPATTERNS= option in the STYLEATTRS statement to specify the fill pattern for all slices, except OTHER.

You can use the DATACONTRASTCOLOR= option in the STYLEATTRS statement to specify the colors used for the fill pattern for all slices, except OTHER.

HOLELABEL
HOLELABEL="*string*"

displays a label for the donut hole. If you specify a label, then that label is used for the donut hole. If you do not specify a label, then the RESPONSE variable label is used for the donut hole.

HOLELABELATTRS=*style-element*<(*options*)> | (*options*)

specifies text attributes for the label text in the donut hole. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults The Color, FontFamily, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

The font size is 60% of the hole value font size, which, by default, is determined internally based on the hole size.

Interaction This option has no effect unless HOLELABEL is also specified.

Note The font size specification in *style-element* is honored for the hole label. However, you cannot specify the hole label font size by using the SIZE= text option in (*options*) or in *style-element*(*options*).

Examples
```
HOLELABELATTRS=(Color=Green Family="Arial"
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
HOLELABELATTRS=GraphLabelText
```

HOLEVALUE
HOLEVALUE=*number*
displays a numeric value inside the donut hole.

If you specify a number, then that number is displayed inside the donut hole. If you do not specify a number, then the sum of the donut slices is displayed inside the donut hole.

Tip The label for large values uses a condensed business format to abbreviate numerical values. For example, 1,100,000,000 is displayed as 1.1B. This format can cause you to lose precision, as it rounds values to improve the display within the hole area.

HOLEVALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies text attributes for the value text in the donut hole. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults The Color, FontFamily, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

The font size is determined internally based on the hole size.

Interaction This option has no effect unless HOLEVALUE is also specified.

Note The font size specification in *style-element* is honored for the hole value. However, you cannot specify the hole value font size by using the SIZE= text option in (*options*) or in *style-element*(*options*).

Tip When the hole label is displayed, by default, the label font size is 60% of the hole value font size. If you want a different font size for the hole label, add the HOLEVALUEATTRS= option and specify a style element that has the label attributes that you want.

Examples
```
HOLEVALUEATTRS=(Color=Green Family="Arial"
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
HOLEVALUEATTRS=GraphLabelText
```

MAXSLICES=*positive integer*
specifies the maximum number of slices in the donut. Any remaining slices are consolidated into the Other slice. The Other slice is included in this count.

In the resulting chart, the largest *N*-1 slices are displayed, where *N* equals your specified *positive integer*. The remaining smaller slices are consolidated into the Other slice.

Default No maximum is set

Interaction This option does not control the drawing order of the slices. Slices are drawn as specified by the SLICEORDER= option.

MISSING
shows the data for missing category values.

Default Missing data is not shown.

NOFILL
turns off the display of fill colors.

Interaction This option takes effect only if the FILLPATTERN option is specified.

OTHERFILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency of the Other slice. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Default The Color attribute of the GraphOther:style element in the current style. Transparency is the same as the plot's transparency.

OTHERLABEL="*string*"
specifies a label for the Other slice.

Default "OTHER"

OTHERPERCENT=*percent-of-total*
collects all category values with response values less than or equal to *percent-of-total* into the Other slice.

Default Not set

Range 0 to 100

Interaction This option is ignored if MAXSLICES= is specified.

RESPONSE=*numeric-variable*
specifies a numeric variable for the response values.

RINGSIZE=*numeric-value*
specifies the proportion of the donut radius to be occupied by the donut ring.

In the following figures, the first figure shows a ring that is half the diameter of the donut. The second figure shows a much narrower ring.

Table 9.1 *Sample Ring Sizes*

RINGSIZE=0.5	RINGSIZE=0.1

Default 0.3

Range 0.0 to 1.0

Interaction The donut hole cannot be sized below a minimum proportionate value of 0.15. If changing the ring size results in a smaller donut hole than that minimum, the ring size is automatically adjusted to honor the minimum size for the donut hole.

SLICEORDER=DATA | FORMATTED | UNFORMATTED | RESPASC | RESPDESC
specifies the order of the donut slices.

DATA
draws the slices in data order.

FORMATTED
draws the slices in ascending order of the formatted category variable.

This option takes effect when a format has been applied to the category variable.

UNFORMATTED
draws the slices in ascending order of the unformatted category variable.

RESPASC
draws the slices in ascending summed order of the response variable.

RESPDESC
draws the slices in descending summed order of the response variable.

Default UNFORMATTED order

Tip By default, slices are drawn in a counterclockwise direction. You can change this using the DIRECTION= option.

STARTANGLE=*degrees*
specifies which degree between 0 and 360 serves as the starting position for the first slice.

A value of 0 degrees corresponds to the three o'clock position. The *degrees* value specified moves the starting position counterclockwise beginning with the first quadrant of the donut.

Default 0

Range 0 to 360

Interaction By default, the center, rather than the edge, of the first slice starts at the specified position. You can change this by specifying STARTPOS=EDGE.

Tips To start the first slice in the fourth quadrant of the donut, specify *degrees* to be greater than 270.

From the starting point specified by this option, the slices are drawn in the direction specified by the DIRECTION= option.

STARTPOS=CENTER | EDGE
specifies whether the center or the edge of the slice is positioned at the start angle.

Default CENTER

Tips Use the DIRECTION= option to specify the category direction. By default, the direction is counterclockwise.

Use the STARTANGLE= option to specify the start angle. By default, the start angle is zero degrees.

STATFMT=*format-specification*
specifies the format of the displayed statistical value. *format-specification* can be a SAS format or a user-defined format created with the FORMAT procedure. This format is applied to the calculated column. When no response variable is specified, the format is applied to the category frequency.

Restriction The format must be a numeric format.

Tip Use this option to change the default format that might contain decimal points, percentages, or commas.

See *SAS Formats and Informats: Reference*

Examples
```
statfmt=f8.2

statfmt=comma10.2
```

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this chart.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example http://www.sas.com

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS ON / IMAGEMAP=ON;
```

For more information, see “Using the ODS GRAPHICS Statement” on page 1886.

KEYLEGEND Statement

Creates a legend based on information from pie and donut statements. This statement is used to control the automated legend when it is displayed.

Interaction: Legends appear only when the category data label is not displayed and the NOAUTOLEGEND option is not specified in the PROC SGPIE statement. The category display is controlled by the DATALABELDISPLAY= option in the PIE and DONUT statements.

Syntax

KEYLEGEND </*options*>;

Summary of Optional Arguments

Appearance options

ACROSS=*n*
: specifies the number of columns in the legend.

BORDER

NOBORDER
: specifies whether a border is displayed around the legend.

DOWN=*n*
: specifies the number of rows in the legend.

FILLASPECT=GOLDEN | *positive-number*
: specifies an aspect ratio for the fill swatches based on their height.

FILLHEIGHT=*numeric-value*<*units*>
: specifies the height of the fill swatches.

POSITION=*position-value*
: specifies the position of the legend within the graph.

TITLE="*text-string*"
: specifies a title for the legend.

TITLEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the title.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the legend value labels.

Legend options

AUTOITEMSIZE
: specifies that all swatches in the legend are sized in proportion to the font size used for the legend value labels.

SCALE=*positive-number*
: specifies a scale factor that is to be applied to the fill swatch height.

SORTORDER=ASCENDING | DESCENDING
: specifies the sort order to use for the legend entry labels.

Optional Arguments

ACROSS=*n*

specifies the number of columns in the legend.

Default Determined automatically

Note Depending on the number of legend entries and the number of columns and rows that you specify, the legend might not fit in your graph. If your

legend does not appear, then you might need to specify a different value for the ACROSS= option.

Tip The DOWN= option specifies the number of rows in the legend.

AUTOITEMSIZE
specifies that all swatches in the legend are sized in proportion to the font size used for the legend value labels. These proportional sizes take effect regardless of the attributes that are used in the plot.

The following figures show a legend with fairly large labels. In the first figure, the color swatches are small compared to the labels. The second figure uses AUTOITEMSIZE to size the color swatches in proportion to the labels.

Default Swatch Size	AUTOITEMSIZE Used in the Statement

Tip Use the VALUEATTRS= option to control the font size for the legend value labels.

BORDER | NOBORDER
specifies whether a border is displayed around the legend. The BORDER option shows the border. The NOBORDER option hides the border.

Default BORDER

DOWN=*n*
specifies the number of rows in the legend.

Note: Depending on the number of legend entries and the number of columns and rows that you specify, the legend might not fit in your graph. If your legend does not appear, then you might need to specify a different value for the DOWN= option.

Default Determined automatically

Tip The ACROSS= option specifies the number of columns in the legend.

FILLASPECT=GOLDEN | *positive-number*
specifies an aspect ratio for the fill swatches based on their height.

GOLDEN
specifies the golden ratio of 1.618 (width = 1.618 * height).

positive-number
specifies a custom aspect ratio.

Default The system determines the aspect.

Interaction This option is ignored when AUTOITEMSIZE is specified.

Tip Use FILLHEIGHT= to specify the height.

FILLHEIGHT=*numeric-value<units>*
specifies the height of the fill swatches. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The system determines the height.

Interaction This option is ignored when AUTOITEMSIZE is specified.

Tip Use FILLASPECT= to specify the aspect ratio.

POSITION=*position-value*
specifies the position of the legend within the graph. The positions are as follows:

BOTTOM
places the legend at the bottom of the graph.

BOTTOMLEFT
places the legend at the bottom left corner of the graph.

BOTTOMRIGHT
places the legend at the bottom right corner of the graph.

LEFT
places the legend at the left side of the graph.

RIGHT
places the legend at the right side of the graph.

TOP
places the legend at the top of the graph.

TOPLEFT
places the legend at the top left corner of the graph.

TOPRIGHT
places the legend at the top right corner of the graph.

Default BOTTOM

Notes By default, if you use more than one KEYLEGEND statement, then each legend is placed in a different position.

If you specify more than one legend with the same position, then all of your legends are placed at that position.

SCALE=*positive-number*
specifies a scale factor that is to be applied to the fill swatch height. Values greater than 1 increase the height while values less than 1 reduce the height.

Default 1

Interaction This option is ignored when AUTOITEMSIZE is specified.

Tips Use FILLHEIGHT= to change the base height.

Use FILLASPECT= to specify the width.

SORTORDER=ASCENDING | DESCENDING
specifies the sort order to use for the legend entry labels.

TITLE="*text-string*"
specifies a title for the legend. The title is placed to the left of the legend body, except in the following cases:

- the legend contains two or more rows of items
- the legend title length exceeds the space that is available on the left side of the legend

In those cases, the title is placed above the legend body.

Default The category variable label, if one is assigned. If a label is not assigned, the category variable name is displayed by default.

Requirement *text-string* must be enclosed in quotation marks.

Tip To suppress the default title in an auto-generated legend, specify TITLE="" in a KEYLEGEND statement as shown in the following example:

```
proc sgpie data=sashelp.cars;
   pie type / datalabeldisplay=(percent);
   keylegend / title="";
run;
```

TITLEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the title. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Examples
```
TITLEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TITLEATTRS=GraphTitleText
```

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the legend value labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

PIE Statement

Creates a pie chart with slices representing counts or percents of unique category values. If a RESPONSE variable is specified, the chart summarizes the category values over corresponding values of the RESPONSE variable.

Examples:
"Example 3: Simple Pie Chart" on page 879
"Example 4: Pie Chart with a Response Variable" on page 880
"Example 5: Pie Chart That Combines Categories into the Other Slice" on page 880

Syntax

PIE *category-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: enhances the visual appearance of the filled pie slices.

DIRECTION=COUNTERCLOCKWISE | CLOCKWISE
: specifies whether to display the donut slices in counterclockwise or clockwise sequence.

FILLPATTERN
: enables the display of line fill patterns for the chart.

NOFILL
: turns off the display of fill colors.

OTHERFILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency of the Other slice.

STARTANGLE=*degrees*
: specifies which degree between 0 and 360 serves as the starting position for the first slice.

STARTPOS=CENTER | EDGE
: specifies whether the center or the edge of the slice is positioned at the start angle.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Label options

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the color and font attributes of the slice labels.

DATALABELDISPLAY=ALL | NONE | (*content-options*)
: specifies the information to display in the slice labels.

DATALABELLOC=INSIDE | OUTSIDE | CALLOUT
: specifies whether to display the slice labels within the pie slices or outside of the pie circumference.

OTHERLABEL="*string*"
: specifies a label for the Other slice.

STATFMT=*format-specification*
: specifies the format of the displayed statistical value.

Plot options

MAXSLICES=*positive integer*
: specifies the maximum number of slices in the pie. Any remaining slices are consolidated into the Other slice.

MISSING
: shows the data for missing category values.

OTHERPERCENT=*percent-of-total*
: collects all category values with response values less than or equal to *percent-of-total* into the Other slice.

RESPONSE=*numeric-variable*
: specifies a numeric variable for the response values.

SLICEORDER=DATA | FORMATTED | UNFORMATTED | RESPASC | RESPDESC
: specifies the order of the pie slices.

URL=*character-variable*
: specifies an HTML page to be displayed when parts of the plot are selected.

Required Argument

category-variable
: specifies the column for the category values. Duplicated category values are summarized into a unique value. All values are treated as discrete.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. Provide the *character-value* variable without quotation marks.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

Example "Example: Use a Discrete Attribute Map with the SGPIE Procedure" on page 1919

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the color and font attributes of the slice labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Note If data label settings result in labels that do not fit within the slices, the labels are not displayed.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELDISPLAY=ALL | NONE | (*content-options*)
specifies the information to display in the slice labels.

ALL
displays all available information

NONE
does not display slice labels

(*content-options*)
a space-separated list of one or more of the following options enclosed in parentheses:

CATEGORY
displays the category value.

RESPONSE
displays the sum of the response value for each category.

If no response variable is specified, displays the category value.

PERCENT
displays the response or category percentage.

The percentage displayed is as follows:

- when a response variable is specified, displays the response sum as a percentage.
- when no response variable is specified, displays the category frequency as a percentage.

Defaults When a response variable is specified, the default display is the category value and the response sum.

When no response variable is specified, the default display is the category value and frequency.

Notes If the category label is suppressed through this option, an automatic legend is generated with the category values.

The position of the labels is adjusted to prevent the labels from overlapping.

If data label settings result in labels that do not fit within the slices, the labels are not displayed.

DATALABELLOC=INSIDE | OUTSIDE | CALLOUT

specifies whether to display the slice labels within the pie slices or outside of the pie circumference.

INSIDE

locates the slice labels inside the pie slices.

Note If there is not sufficient space to fit all of the labels, some labels might encroach on adjacent slices and overlap with other labels.

OUTSIDE

locates the slice labels outside of the pie circumference.

Note The diameter of the pie is reduced in order to fit the labels. In some cases, labels might overlap.

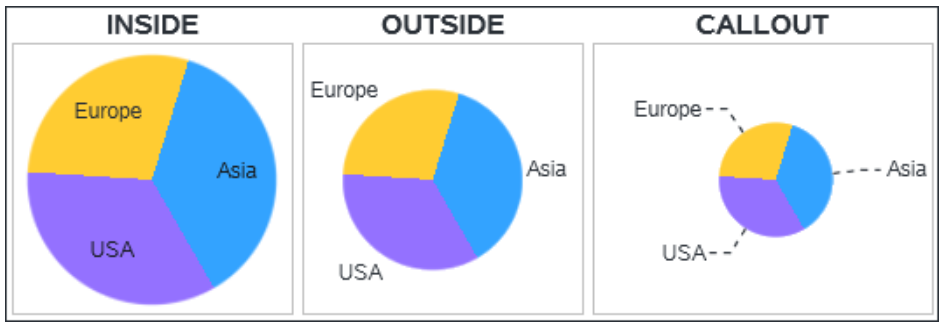
CALLOUT

locates the slice labels outside of the pie circumference and draws a line from the label to its slice.

Each callout line consists of two line segments: a 16-pixel perpendicular segment from the pie circumference, and a 12-pixel segment toward the slice label.

Note The diameter of the pie is reduced in order to fit the labels. The minimum pie size is 33 percent of the available space. If labels do not fit at the minimum size, they are truncated or dropped.

Figure 9.4 *Pie Chart Label Location Examples*



Default Each label is placed using INSIDE or CALLOUT, depending on where the label fits. If some labels do not fit when using CALLOUT, they are truncated or dropped.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

enhances the visual appearance of the filled pie slices.

Default NONE

Restriction The maximum number of skinned graphical elements is limited to 200 per plot. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

DIRECTION=COUNTERCLOCKWISE | CLOCKWISE

specifies whether to display the donut slices in counterclockwise or clockwise sequence.

Default COUNTERCLOCKWISE

Interaction The STARTANGLE= option controls the starting angle for the first donut slice.

FILLPATTERN

enables the display of line fill patterns for the chart.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Note A line fill pattern can be displayed in the OTHER slice as well. The attributes of the OTHER slice fill pattern are controlled by the current ODS style.

Accessibility notes This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the NOFILL option to suppress the chart's fill color.

You can use the DATAFILLPATTERNS= option in the STYLEATTRS statement to specify the fill pattern for all slices, except OTHER.

You can use the DATACONTRASTCOLOR= option in the STYLEATTRS statement to specify the colors used for the fill pattern for all slices, except OTHER.

MAXSLICES=*positive integer*
specifies the maximum number of slices in the pie. Any remaining slices are consolidated into the Other slice. The Other slice is included in this count.

In the resulting chart, the largest *N*-1 slices are displayed, where *N* equals your specified *positive integer*. The remaining smaller slices are consolidated into the Other slice.

Default No maximum is set

Interaction This option does not control the drawing order of the slices. Slices are drawn as specified by the SLICEORDER= option.

MISSING
shows the data for missing category values.

Default Missing data is not shown.

NOFILL
turns off the display of fill colors.

Interaction This option takes effect only if the FILLPATTERN option is specified.

OTHERFILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency of the Other slice. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Default The Color attribute of the GraphOther:style element in the current style. Transparency is the same as the plot's transparency.

OTHERLABEL="*string*"
specifies a label for the Other slice.

Default "OTHER"

OTHERPERCENT=*percent-of-total*
collects all category values with response values less than or equal to *percent-of-total* into the Other slice.

Default Not set

Range 0 to 100

Interaction This option is ignored if MAXSLICES= is specified.

RESPONSE=*numeric-variable*
specifies a numeric variable for the response values.

SLICEORDER=DATA | FORMATTED | UNFORMATTED | RESPASC | RESPDESC
specifies the order of the pie slices.

DATA
draws the slices in data order.

FORMATTED
draws the slices in ascending order of the formatted category variable.

This option takes effect when a format has been applied to the category variable.

UNFORMATTED
draws the slices in ascending order of the unformatted category variable.

RESPASC
draws the slices in ascending summed order of the response variable.

RESPDESC
draws the slices in descending summed order of the response variable.

Default UNFORMATTED order

Tip By default, slices are drawn in a counterclockwise direction. You can change this using the DIRECTION= option.

STARTANGLE=*degrees*
specifies which degree between 0 and 360 serves as the starting position for the first slice. A value of 0 degrees corresponds to the three o'clock position. The *degrees* value specified moves the starting position counterclockwise beginning with the first quadrant of the pie.

Default 0

Range 0 to 360

Interaction By default, the center, rather than the edge, of the first slice starts at the specified position. You can change this by specifying STARTPOS=EDGE.

Tips To start the first slice in the fourth quadrant of the pie, specify *degrees* to be greater than 270.

From the starting point specified by this option, the slices are drawn in the direction specified by the DIRECTION= option.

STARTPOS=CENTER | EDGE
specifies whether the center or the edge of the slice is positioned at the start angle.

Default CENTER

Tips Use the DIRECTION= option to specify the category direction. By default, the direction is counterclockwise.

Use the STARTANGLE= option to specify the start angle. By default, the start angle is zero degrees.

STATFMT=*format-specification*
specifies the format of the displayed statistical value. *format-specification* can be a SAS format or a user-defined format created with the FORMAT procedure. This format is applied to the calculated column. When no response variable is specified, the format is applied to the category frequency.

Restriction The format must be a numeric format.

Tip Use this option to change the default format that might contain decimal points, percentages, or commas.

See *SAS Formats and Informats: Reference*

Examples
```
statfmt=f8.2

statfmt=comma10.2
```

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this chart.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example http://www.sas.com

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS ON / IMAGEMAP=ON;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

STYLEATTRS Statement

Specifies attributes for a graph. The statement enables you to change the appearance of some graphics elements within the procedure.

Requirement: The procedure must include at least one chart statement.

Syntax

STYLEATTRS <*options*>;

Summary of Optional Arguments

BACKCOLOR=*color*
: specifies the background color of the graph area.

DATACOLORS=(*color-list*)
: specifies the fill colors for the graphics elements.

DATACONTRASTCOLORS=(*color-list*)
: specifies the contrast colors for the fill pattern, if used.

DATAFILLPATTERNS=(*fill-pattern-list*)
: specifies the list of graph fill patterns.

Optional Arguments

BACKCOLOR=*color*
specifies the background color of the graph area. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The Color attribute of the GraphBackground style element

Examples backcolor=CXFF0000

backcolor=light_blue

DATACOLORS=(*color-list*)
specifies the fill colors for the graphics elements. Provide a space-separated list of colors enclosed in parentheses. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The colors that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Interaction Where applicable, the COLOR= suboption of any plot option related to fill color overrides the DATACOLORS= option.

Note When this option is specified, the colors cycle through *color-list* rather than the colors that are defined in the GraphData1–GraphData*n* style elements. When the colors in *color-list* are exhausted, the colors repeat.

Example datacolors=(CXFF0000 green blue)

DATACONTRASTCOLORS=(*color-list*)
specifies the contrast colors for the fill pattern, if used. Provide a space-separated list of colors enclosed in parentheses. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The contrast colors that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Note When this option is specified, the colors cycle through *color-list* rather than the contrast colors that are defined in the GraphData1–GraphData*n* style elements. When the colors in *color-list* are exhausted, the colors repeat.

Example datacontrastcolors=(orange cyan CXFF0000)

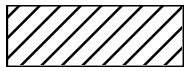
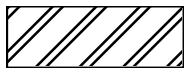
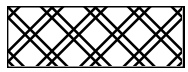
DATAFILLPATTERNS=(*fill-pattern-list*)
specifies the list of graph fill patterns.

fill-pattern-list
a space-separated list of fill patterns, enclosed in parentheses.

You can use a style attribute reference such as GraphData3:fillpattern or a two-character fill-pattern name such as L1, X5, and so on, to specify a fill pattern. The list can contain a mix of style attribute references and pattern names.

To specify a two-character fill-pattern name, specify a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5. The following table shows the patterns for each of the possible combinations.

Table 9.2 Fill Patterns

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	

Note: Fill patterns E and S are not supported by the SGPIE procedure.

When this option is specified, the fill patterns rotate through this fill-pattern list rather than through the fill patterns that are defined in the GraphData1–GraphData*n* style elements. For information about the attribute rotation patterns, see "How the Attributes Are Cycled" on page 904.

Default: The fill patterns that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Requirement: The list of fill patterns must be enclosed in parentheses.

Accessibility note: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Example:
```
datafillpatterns=(graphdata4:fillpattern L5 X2)
```

ATTRIB Statement

Associates a format, informat, label, and length with one or more variables.

Type: Procedure statement

Syntax

ATTRIB *variable-list(s) attribute-list(s)*;

Arguments

variable-list(s)
names the variables that you want to associate with the attributes.

> **TIP** List the variables in any form that SAS allows.

attribute-list(s)
specifies one or more attributes to assign to *variable-list*. Specify one or more of these attributes in the ATTRIB statement:

FORMAT=*format*
associates a format with variables in *variable-list*.

Tip The format can be either a standard SAS format or a format that is defined with the FORMAT procedure.

INFORMAT=*informat*
associates an informat with variables in *variable-list*.

Tip The informat can be either a standard SAS informat or an informat that is defined with the FORMAT procedure.

LABEL='*label*'
associates a label with variables in *variable-list*.

See Also

"ATTRIB Statement" on page 142 for more information about the ATTRIB statement.

BY Statement

Orders the output according to the BY groups.

Type: Procedure statement

Syntax

BY <DESCENDING> *variable-1*

<-<DESCENDING> *variable-n*>
<NOTSORTED>;

Required Argument

variable
specifies the variable that the procedure uses to form BY groups. You can specify more than one variable. If you do not use the NOTSORTED option in the BY statement, then either the observations in the data set must be sorted by all the variables that you specify, or they must be indexed appropriately. Variables in a BY statement are called *BY variables*.

Optional Arguments

DESCENDING
specifies that the observations are sorted in descending order by the variable that immediately follows the word DESCENDING in the BY statement.

NOTSORTED
specifies that observations are not necessarily sorted in alphabetic or numeric order. The observations are grouped in another way (for example, chronological order).

The requirement for ordering or indexing observations according to the values of BY variables is suspended for BY-group processing when you use the NOTSORTED option. In fact, the procedure does not use an index if you specify NOTSORTED. The procedure defines a BY group as a set of contiguous observations that have the same values for all BY variables. If observations with the same values for the BY variables are not contiguous, then the procedure treats each contiguous set as a separate BY group.

You cannot use the NOTSORTED option in a PROC SORT step.

Note: You cannot use the GROUPFORMAT option, which is available in the BY statement in a DATA step, in a BY statement in any PROC step.

See Also

"BY Statement" on page 143 for more information about the BY statement.

FORMAT Statement

Associates formats with variables.

Type: Procedure statement

Syntax

FORMAT *variable-1* <-*variable-n*> <*format-1*> <*variable-1* <-*variable-n*> <*format-2*> > - <DEFAULT=*default-format*>;

Required Argument

variable
names one or more variables for SAS to associate with a format. You must specify at least one *variable*.

Tip To disassociate a format from a variable, use the variable in a FORMAT statement without specifying a format in a DATA step or in PROC DATASETS. In a DATA step, place this FORMAT statement after the SET statement. See "Removing a Format" in *SAS DATA Step Statements: Reference*. You can also use PROC DATASETS.

Optional Arguments

format
specifies the format that is listed for writing the values of the variables.

Tip Formats that are associated with variables by using a FORMAT statement behave like formats that are used with a colon modifier in a subsequent PUT statement. For more information about using a colon modifier, see "PUT Statement: List" in *SAS DATA Step Statements: Reference*.

See *SAS Formats and Informats: Reference*

DEFAULT=*default-format*
specifies a temporary default format for displaying the values of variables that are not listed in the FORMAT statement. These default formats apply only to the current DATA step; they are not permanently associated with variables in the output data set.

A DEFAULT= format specification applies to

- variables that are not named in a FORMAT or ATTRIB statement
- variables that are not permanently associated with a format within a SAS data set
- variables that are not written with the explicit use of a format.

Default If you omit DEFAULT=, SAS uses BEST*w*. as the default numeric format and $*w*. as the default character format.

Restriction Use this option only in a DATA step.

Tip A DEFAULT= specification can occur anywhere in a FORMAT statement. It can specify either a numeric default, a character default, or both.

Example "Assigning Formats and Defaults" in *SAS DATA Step Statements: Reference*

See Also

"FORMAT Statement" on page 146 for more information about the FORMAT statement.

LABEL Statement

Assigns descriptive labels to variables.

Type: Procedure statement

Syntax

Form 1: Creates a descriptive label for one or more variables.

LABEL *variable-1*='*text-string*' <...*variable-n*='*text-string*>;

Form 2: Removes the label from one or more variables.

LABEL *variable-1* = ' ' < ...*variable-n*=' '>; | *variable-1* = <...*variable-n*= >

Arguments

variable

specifies the variable that you want to label. You can specify labels for multiple variables in a single LABEL statement:

```
data test;
   L = 5;
   W = 10;
   label L = 'Length' W = 'Width';
run;
```

text-string

specifies a label of up to 256 bytes.

```
data test;
   L = 5;
   label L = 'Length';
run;
```

Restrictions If the label includes a semicolon (;) or an equal sign (=), you must enclose the label in either single or double quotation marks.

```
label N = 'Number = ';
```

If the label includes single quotation marks (’), you must enclose the label in double quotation marks.

```
label Name = "Owner's Last Name";
```

There is partial support for variable name substitution with the JAVAIMG device. When the label text for the variable exceeds the space allowed, the variable name is used instead by procedures that are labeling an axis or a legend title. With the exception of the SAS/GRAPH GPLOT procedure, the variable name is not used when the JAVAIMG device is specified.

Notes

Quotation marks are not required unless the label includes a semicolon, quotation marks, or an equal sign.

The LABEL statement expects an equal sign (=) after the first variable. If any other character is found, an error occurs. The LABEL statement considers everything after the equal sign, following the first variable, as part of the label until the statement comes to another variable followed by an equal sign, regardless of quotation marks. In this example, the label for x is **`this is x y 4 = this is y`** because the next variable that is followed by an equal sign is z.

```
data test;
   x = 1;
   y = 1;
   z = 1;
   label x = 'this is x' y 4 = 'this is y' z = 'This is z';
run;
```

In this LABEL statement, the x variable has the same label:

```
label x = this is x y 4 = this is y  z = This is z;
```

See

“Character Constants and Character Variables” in *SAS Programmer’s Guide: Essentials*.

’’

removes a label from a variable. Enclose a single blank space in quotation marks to remove an existing label. You can remove labels from multiple variables in a single LABEL statement:

```
data test;
   L=5; W=10;
   label L = ' '  W = ' ';
run;
```

Alternatively, you can remove a label by specifying nothing after the equal sign:

```
data test;
   L=5; W=10;
   label L =  W = ;
run;
```

See Also

"LABEL Statement" on page 150 for more information about the LABEL statement.

WHERE Statement

Subsets the input data set by specifying certain conditions that each observation must meet before it is available for processing.

Type: Procedure statement

Syntax

WHERE *where-expression*;

Required Argument

where-expression
is an arithmetic or logical expression that generally consists of a sequence of operands and operators. For more information about where processing, see *SAS DATA Step Statements: Reference*.

Tip You can specify multiple where-expressions.

See Also

"WHERE Statement" on page 151 for more information about the WHERE statement.

Examples: SGPIE Procedure: Preproduction

About the SGPIE Procedure Examples

Examples in This Section

This section provides basic examples of how to use the SGPIE procedure. The following examples are provided:

Example	Description
"Example 1: Simple Donut Chart" (p. 878)	Shows how to use the DONUT statement to create a simple donut chart.
"Example 2: Donut Chart with a Response Variable" (p. 878)	Shows how to use the DONUT statement and the RESPONSE= and DATALABELDISPLAY= options to create a donut chart with a response variable.
"Example 3: Simple Pie Chart" (p. 879)	Shows how to use the PIE statement to create a simple pie chart.
"Example 4: Pie Chart with a Response Variable" (p. 880)	Shows how to use the PIE statement and the RESPONSE= option to create a pie chart with a response variable.
"Example 5: Pie Chart That Combines Categories into the Other Slice" (p. 880)	Shows how to create a pie chart that consolidates some category values into the other slice.
"Example 6: Pie Chart with Specified Background Color and Data Colors" (p. 881)	Shows how to create a pie chart with a specific background color and data colors.

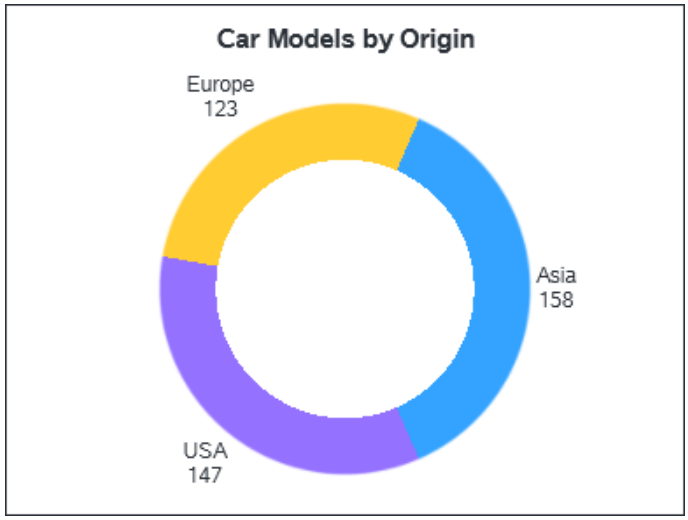
Examples Available Online

You can find more examples on the Graphically Speaking blog. On the blog page, use **Search Graphically Speaking** to search for SGPIE to find the SGPIE procedure examples.

Example 1: Simple Donut Chart

This example shows the category value and frequency numbers for cars.

Output 9.1 *Output*



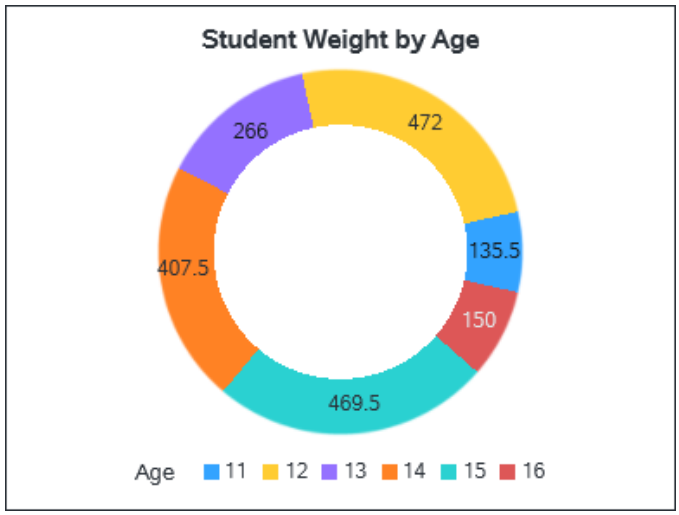
Example Code 9.1 *Program*

```
title "Car Models by Origin";
proc sgpie data=sashelp.cars;
  donut origin / datalabelloc=outside;
run;
title;
```

Example 2: Donut Chart with a Response Variable

This example displays the sum of a response variable for each value of the category variable. The DATALABELDISPLAY= on page 846 option displays only the response variable in the donut slices. The category values are displayed in the legend.

Output 9.2 *Output*



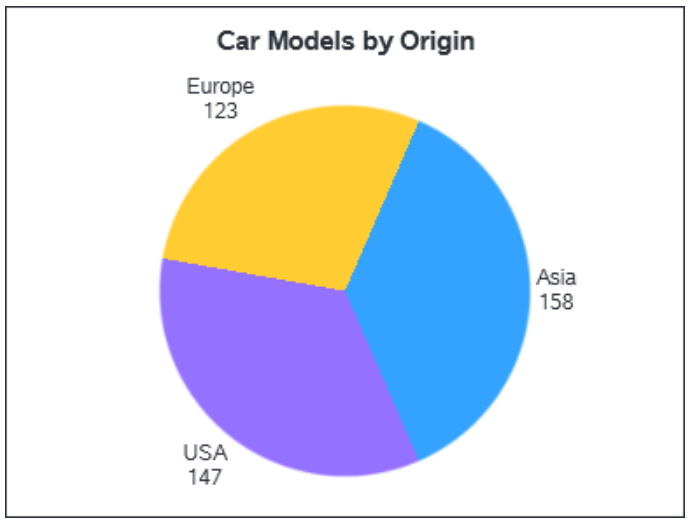
Example Code 9.2 *Program*

```
title "Student Weight by Age";
proc sgpie data=sashelp.class;
  donut age / response=weight datalabeldisplay=(response);
run;
title;
```

Example 3: Simple Pie Chart

This example shows the category value and frequency numbers for cars.

Output 9.3 *Output*



Example Code 9.3 *Program*

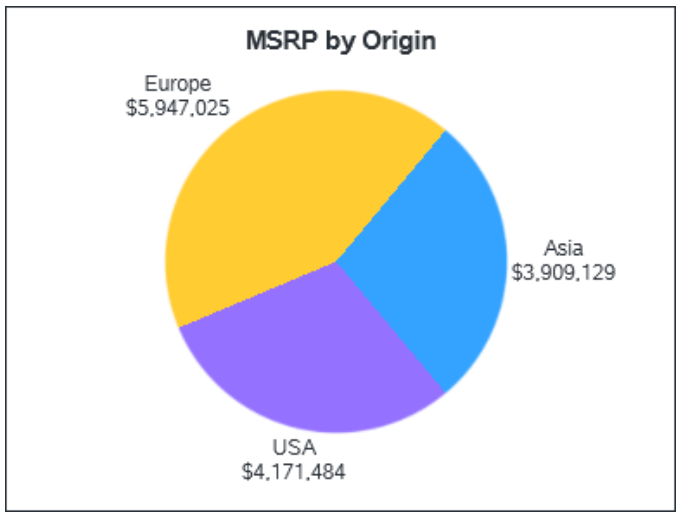
```
title "Car Models by Origin";
proc sgpie data=sashelp.cars;
```

```
  pie origin / datalabelloc=outside;
run;
title;
```

Example 4: Pie Chart with a Response Variable

This example shows the sum of the manufacturer's suggested retail price (MSRP) for each value of the category variable. The DATALABELDISPLAY= on page 861 option displays only the response variable in the pie slices. The category values are displayed in the legend.

Output 9.4 *Output*



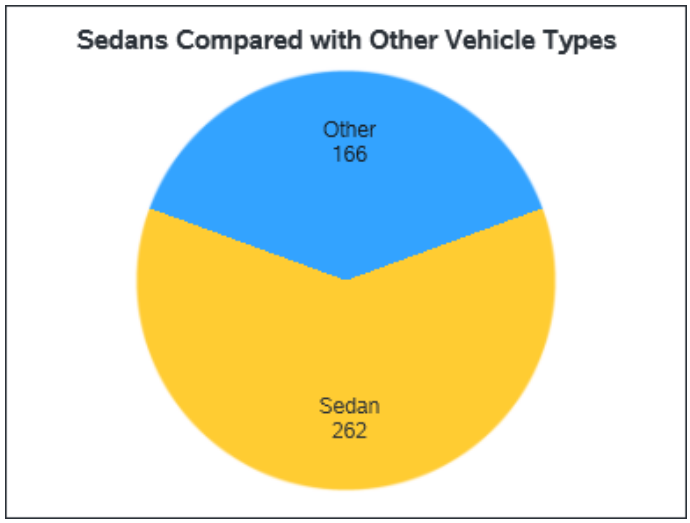
Example Code 9.4 *Program*

```
title "MSRP by Origin";
proc sgpie data=sashelp.cars;
  pie origin / response=msrp datalabelloc=outside;
run;
title;
```

Example 5: Pie Chart That Combines Categories into the Other Slice

This example produces a pie chart with only two categories of vehicles: Sedan and Other. Pie charts with only two slices are more symmetric and easier to comprehend at a glance. To accomplish this, the example defines a format that uses the Other keyword to group all vehicle types that are not sedans.

Output 9.5 *Output*



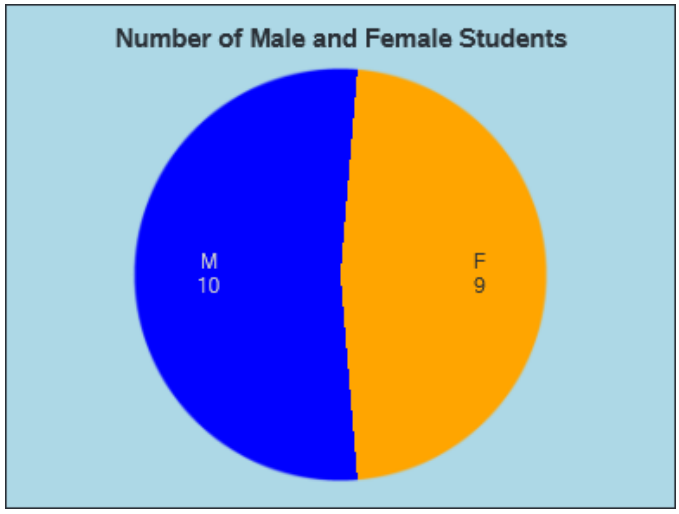
Here is the SAS code that defines the format and generates the SGPIE procedure output.

Example Code 9.5 *Program*

```
proc format;
  value $type
    'Sedan'='Sedan'
      other='Other';
run;

title "Sedans Compared with Other Vehicle Types";
proc sgpie data=sashelp.cars;
  format type $type.;
  pie type / startangle=90;
run;
title;
```

Example 6: Pie Chart with Specified Background Color and Data Colors

The following example shows a pie chart with the background color and the data colors specified.



Example Code 9.6 *Program*

```
title "Number of Male and Female Students";
proc sgpie data=sashelp.class;
  styleattrs backcolor=lightblue datacolors=(orange blue);
  pie sex ;
run;
title;
```

10

SGPLOT Procedure

Overview: SGPLOT Procedure ***883***
What Does the SGPLOT Procedure Do? 884

Concepts: SGPLOT Procedure ***886***
Overview of the SGPLOT Procedure 886
Concepts in Common with the SGPANEL Procedure 887

Syntax: SGPLOT Procedure ***888***
PROC SGPLOT Statement 890
STYLEATTRS Statement 898
BAND Statement 907
BLOCK Statement 918
BUBBLE Statement 930
DENSITY Statement 946
DOT Statement 954
DROPLINE Statement 973
ELLIPSE Statement 979
ELLIPSEPARM Statement 988
FRINGE Statement 998
GRADLEGEND Statement 1003
HBAR Statement 1008
HBARBASIC Statement 1034
HBARPARM Statement 1055
HBOX Statement 1077
HEATMAP Statement 1097
HEATMAPPARM Statement 1109
HIGHLOW Statement 1118
HISTOGRAM Statement 1136
HLINE Statement 1145
INSET Statement 1166
KEYLEGEND Statement 1170
LEGENDITEM Statement 1181
LINEPARM Statement 1188
LOESS Statement 1195
NEEDLE Statement 1210
PBSPLINE Statement 1221
POLYGON Statement 1238
REFLINE Statement 1260
REG Statement 1267
SCATTER Statement 1283

SERIES Statement 1301
SPLINE Statement 1327
STEP Statement 1343
SYMBOLCHAR Statement 1361
SYMBOLIMAGE Statement 1368
TEXT Statement 1373
VBAR Statement 1389
VBARBASIC Statement 1418
VBARPARM Statement 1438
VBOX Statement 1462
VECTOR Statement 1482
VLINE Statement 1493
WATERFALL Statement 1514
XAXIS Statement 1524
X2AXIS Statement 1550
XAXISTABLE Statement 1575
YAXIS Statement 1587
Y2AXIS Statement 1611
YAXISTABLE Statement 1635
ATTRIB Statement 1650
BY Statement 1651
FORMAT Statement 1652
LABEL Statement 1653
WHERE Statement 1655

Examples: SGPLOT Procedure ***1656***
About the SGPLOT Procedure Examples 1656
Example 1: Grouping a Scatter Plot 1660
Example 2: Clustering a Grouped Scatter Plot 1661
Example 3: Plotting Three Series 1662
Example 4: Adding Prediction and Confidence Bands to a Regression Plot 1663
Example 5: Adding a Prediction Ellipse to a Scatter Plot 1664
Example 6: Creating Lines and Bands from Pre-Computed Data 1666
Example 7: Adding Statistical Limits to a Dot Plot 1668
Example 8: Combining Histograms with Density Plots 1669
Example 9: Creating a Horizontal Box Plot 1670
Example 10: Creating a Bar-Line Chart 1671
Example 11: Creating a High-Low Chart 1672
Example 12: Series Plot with Line-Thickness Response and Arrowheads 1673

Overview: SGPLOT Procedure

What Does the SGPLOT Procedure Do?

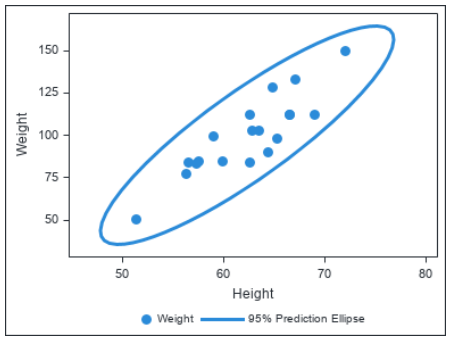
The SGPLOT procedure creates one or more plots and overlays them on a single set of axes. You can use the SGPLOT procedure to create statistical graphics such as histograms and regression plots, in addition to simple graphics such as scatter plots

and line plots. Statements and options enable you to control the appearance of your graph and add additional features such as legends and reference lines.

The SGPLOT procedure can create a wide variety of plot types, and can overlay plots together to produce many different types of graphs.

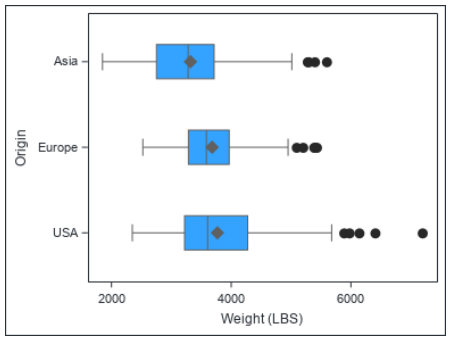
Here are some examples of graphs that the SGPLOT procedure can create.

Table 10.1 *Examples of Graphs That Can Be Generated by the SGPLOT Procedure*



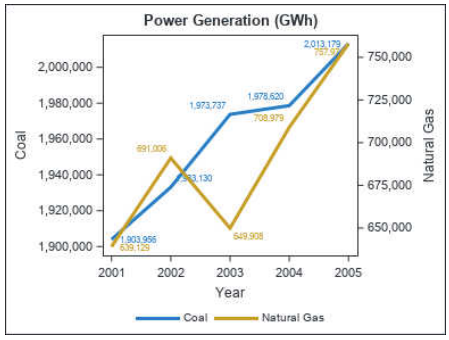
The following code creates an ellipse plot:

```
proc sgplot data=sashelp.class;
  scatter x=height y=weight;
  ellipse x=height y=weight;
run;
```



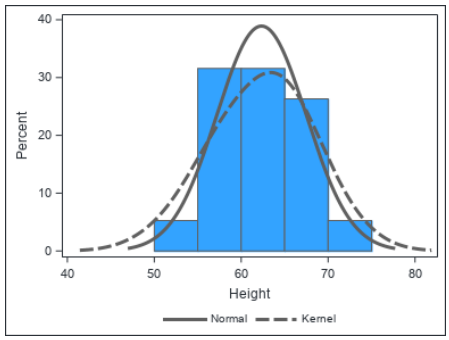
The following code creates a horizontal box plot:

```
proc sgplot data=sashelp.cars;
  hbox weight / category=origin;
run;
```



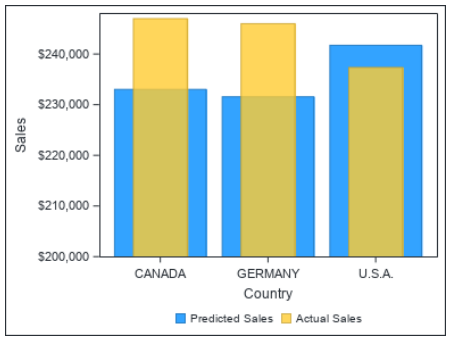
The following code creates a graph with two series plots:

```
title "Power Generation (GWh)";
proc sgplot data=sashelp.electric(where=
  (year >= 2001 and customer="Residential"));
  xaxis type=discrete;
  series x=year y=coal / datalabel;
  series x=year y=naturalgas /
      datalabel y2axis;
run;
title;
```



The following code creates a graph with a histogram, a normal density curve, and a kernel density curve:

```
proc sgplot data=sashelp.class;
  histogram height;
  density height;
  density height / type=kernel;
run;
```



The following code creates a graph with two bar charts:

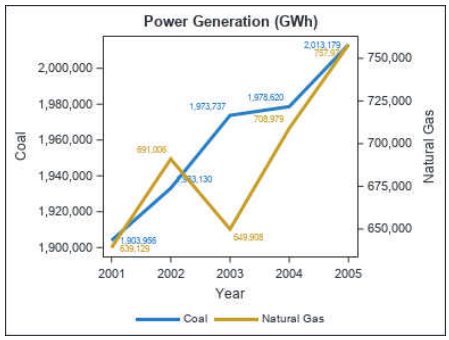
```
proc sgplot data=sashelp.prdsale;
  yaxis label="Sales" min=200000;
  vbar country / response=predict;
  vbar country / response=actual
                 barwidth=0.5
                 transparency=0.2;
run;
```

Concepts: SGPLOT Procedure

Overview of the SGPLOT Procedure

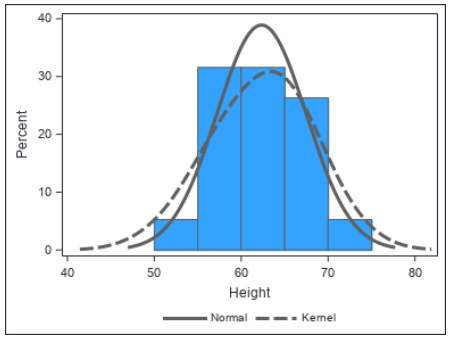
The SGPLOT procedure creates single-cell graphs with a wide range of plot types including density, dot, needle, series, bar, histograms, box, and others. The procedure can compute and display loess fits, polynomial fits, penalized B-spline fits, and ellipses. You can also add text, legends, and reference lines. Options are available for specifying colors, marker symbols, and other attributes of plot features. You can customize the axes by using axis statements such as XAXIS and YAXIS.

Plot statements can be combined to create more informative graphs. The following example shows two series plots that are overlaid in a single graph. Each plot is assigned to a different vertical axis. Data labels have been added for easy reference.




```
title "Power Generation (GWh)";
proc sgplot data=sashelp.electric(where=
  (year >= 2001 and customer="Residential"));
  xaxis type=discrete;
  series x=year y=coal / datalabel;
  series x=year y=naturalgas /
      datalabel y2axis;
run;
title;
```

The following example creates a graph with a histogram, a normal density curve, and a kernel density curve.




```
proc sgplot data=sashelp.class;
  histogram height;
  density height;
  density height / type=kernel;
run;
```

Concepts in Common with the SGPANEL Procedure

The following topics are located in the Chapter 13, "Common Concepts," on page 1777 section. These topics describe concepts that are similar between the SGPANEL and SGPLOT procedures.

Table 10.2 *Common Concepts*

"Plot Type Compatibility" (p. 1779)	explains which types of plots can be used together in a graph
"Plot Axes" (p. 1784)	describes the axis types that are supported by the procedure
"Legends" (p. 1790)	explains how legends are created automatically, and how to create customized legends
"Automatic Differentiation of Visual Attributes" (p. 1797)	explains when different style attributes are automatically assigned to plots, and how to force the procedure to use

	different style attributes if they are not automatically assigned
"Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" (p. 1802)	describes how to split the text for data labels, curve labels, and axis tick mark values when there is not enough room to display the text normally
"Marker Fills and Outlines" (p. 1805)	describes how you can change the appearance of both the marker fill and its outline for graphs that contain markers.
"Color-Naming Schemes" (p. 1808)	describes the color-naming schemes that are supported in ODS Graphics and how to specify colors in your programs.
"Fonts" (p. 1819)	describes how to determine the fonts that are available for your graphs, how to specify fonts in your programs, and how to register your own fonts in SAS.

Syntax: SGPLOT Procedure

Requirement: At least one plot statement is required.

PROC SGPLOT <*options*>;

STYLEATTRS </*options*>

BAND X=*variable* Y=*variable*
UPPER= *numeric-value* | *numeric-variable*
LOWER= *numeric-value* | *numeric-variable* </*options*>;

BLOCK X=*category-variable* BLOCK=*block-variable* </*options*>;

BUBBLE X=*variable* Y=*variable* SIZE=*numeric-variable* </*options*>;

DENSITY *response-variable* </*options*>;

DOT *category-variable* </*options*> ;

DROPLINE X=*variable x-axis-value*
Y=*variable* | *y-axis-value* </*options*>;

ELLIPSE X=*numeric-variable* Y=*numeric-variable* </*options*>;

ELLIPSEPARM SEMIMAJOR= *numeric-value* | *numeric-variable*
SEMIMINOR= *numeric-value* | *numeric-variable* </*options*>

FRINGE *numeric-variable* </*options*>;

GRADLEGEND <"*name*"> </*options*>;

HBAR *category-variable* </*options*>;

HBARBASIC *category-variable* </*options*>;

HBARPARM CATEGORY= *category-variable* RESPONSE= *numeric-variable* </*options*>;

HBOX *analysis-variable* </*options*>;

HEATMAP X=*variable* Y=*variable* </*options*>;

HEATMAPPARM X=*variable* Y=*variable*
COLORGROUP=*variable* | COLORRESPONSE=*numeric-variable* </*options*>;

HIGHLOW X=*variable* | Y=*variable*
HIGH=*numeric-variable* LOW=*numeric-variable* </option(s)>;

HISTOGRAM *response-variable* </*options*>;

HLINE *category-variable* </*options*>;

INSET *"text-string-1" ..."text-string-n"* | (*label-list*) </*options*>;

KEYLEGEND <*"name-1" ... "name-n"*> </*options*>;

LEGENDITEM TYPE=*type* NAME=*"text-string"* </*options*>;

LINEPARM X=*numeric-value* | *numeric-variable*
Y=*numeric-value* | *numeric-variable*
SLOPE=*numeric-value* | *numeric-variable*
</option(s)>;

LOESS X=*numeric-variable* Y=*numeric-variable* </*options*>;

NEEDLE X=*variable* Y=*numeric-variable* </*options*>;

PBSPLINE X=*numeric-variable* Y=*numeric-variable* </*options*>;

POLYGON X=*x-variable* Y=*y-variable* ID=*id-variable* </*options*>;

REFLINE *value(s)* </*options*>;

REG X=*numeric-variable* Y=*numeric-variable* </*options*>;

SCATTER X=*variable* Y=*variable* </*options*>;

SERIES X=*variable* Y=*variable* </*options*>;

SPLINE X=*variable* Y=*variable* </*options*>;

STEP X=*variable* Y=*variable* </*options*>;

SYMBOLCHAR NAME=*identifier*
CHAR=*"hex-string"* | *keyword* </*options*>;

SYMBOLIMAGE NAME=*identifier* IMAGE=*"image-file-specification"* </*options*>;

TEXT X=*variable* Y=*variable* TEXT=*variable* </*options*>;

VBAR *category-variable* </*options*>;

VBARBASIC *category-variable* </*options*>;

VBARPARM CATEGORY=*category-variable* RESPONSE=*numeric-variable* </*options*>;

VBOX *analysis-variable* </*options*>;

VECTOR X=*numeric-variable* Y=*numeric-variable* </*options*>;

VLINE *category-variable* </*options*>;

WATERFALL CATEGORY=*variable* RESPONSE=*numeric-variable* </*options*>;

XAXIS <*options*>;

X2AXIS <*options*>;

XAXISTABLE *variable* < ...*variable-n*> </*options*>;

YAXIS <*options*>;

Y2AXIS <*options*>;

YAXISTABLE *variable* < ...*variable-n*> </*options*>;
ATTRIB *variable-list(s) attribute-list(s)*;
BY *variable-1*<...*variable-N*><*options*>;
FORMAT *variable-1*<...*variable-N*<*format-1*>><*options*>;
LABEL *variable-1*=*label-1*<...*variable-n*=*label-n*>;
WHERE *where-expression*;

PROC SGPLOT Statement

Creates a single-cell plot from input data.

Requirement: An input data set is required.

Syntax

PROC SGPLOT <*options*> ;

Summary of Optional Arguments

ASPECT=*positive-number*
: specifies the aspect ratio of the plot's wall area.

CYCLEATTRS
NOCYCLEATTRS
: specifies whether plots are drawn with unique attributes in the graph.

DATA=*input-data-set*
: specifies the SAS data set that contains the variables to process.

DATTRMAP=*discrete-attribute-map-data-set*
: specifies the discrete attribute map data set that you want to use with the procedure.

DESCRIPTION="*text-string*"
: specifies a description for the output image.

NOAUTOLEGEND
: disables automatic legends from being generated.

NOBORDER
: turns off the display of the graph wall border.

NOSUBPIXEL
SUBPIXEL
: specifies whether subpixel rendering should be used for rendering plots and charts.

NOWALL
: turns off the display of the graph wall's fill.

OPAQUE

NOOPAQUE
specifies whether the graph background is opaque or transparent.

PAD=*dimension<units>* | (*pad-options*)
specifies the amount of extra space to add to the left, right, top, and bottom of the graph.

PCTLEVEL=BY | GRAPH | GROUP
specifies the scope of graph data that is calculated in percentages.

PCTNDEC=*numeric-value*
specifies the number of decimal spaces to be used to calculate the percent values.

RATTRMAP=*range-attribute-map-data-set*
specifies the range attribute map data set that you want to use with the procedure.

SGANNO=*annotation-data-set*
specifies the SG annotation data set that you want to use.

TMPLOUT="*filename*"
writes the Graph Template Language code for your graph to a file.

UNIFORM=GROUP | SCALE | ALL | XSCALE | YSCALE | XSCALEGROUP | YSCALEGROUP
specifies how to control axis scaling and legends when you use a BY statement.

Optional Arguments

ASPECT=*positive-number*
specifies the aspect ratio of the plot's wall area. The ratio is expressed as a positive decimal fraction representing *wall-height* divided by *wall-width*. For example, 0.75 is a 3/4 aspect ratio, and 1.0 is a square aspect ratio.

Small numbers, such as 0.01, produce a short, wide rectangular area. Larger numbers yield a taller, narrower rectangular area.

Default The wall area is sized to the maximum area that can fill the available space.

CYCLEATTRS | NOCYCLEATTRS
specifies whether plots are drawn with unique attributes in the graph. By default, the SGPLOT procedure automatically assigns unique attributes in many situations, depending on the types of plots that you specify. If the plots do not have unique attributes by default, then the CYCLEATTRS option assigns unique attributes to each plot in the graph. The NOCYCLEATTRS option prevents the procedure from assigning unique attributes.

For example, if you specify the CYCLEATTRS option and you create a graph with a SERIES statement and a SCATTER statement, then the two plots have different colors.

If you specify the NOCYCLEATTRS option, then plots have the same attributes unless you specify appearance options such as the LINEATTRS= option.

See "Automatic Differentiation of Visual Attributes" on page 1797

DATA=*input-data-set*
specifies the SAS data set that contains the variables to process. By default, the procedure uses the most recently created SAS data set.

DATTRMAP=*discrete-attribute-map-data-set*
specifies the discrete attribute map data set that you want to use with the procedure. You specify this option only if you are using a discrete attribute map to control visual attributes of the graph.

Restriction The procedures do not support an attribute map that is stored in a table. Only standard SAS data sets are supported for attribute map data.

Requirement The values in the DATTRMAP data set must be sorted by ID. If they are not, only the first value is found.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

DESCRIPTION="*text-string*"
specifies a description for the output image. The description identifies the image in the following locations:

- the alternate text for the image in HTML output
- the table of contents that is created by the CONTENTS option in an ODS statement

The default description is "The SGPLOT Procedure".

Alias DES=

Notes The name of the output image is specified by the IMAGENAME= option in the ODS GRAPHICS statement.

You can disable the alternate text in HTML output by specifying an empty string. That is, `DESCRIPTION=""`.

NOAUTOLEGEND
disables automatic legends from being generated. By default, legends are created automatically for some plots, depending on their content. This option has no effect if you specify a KEYLEGEND statement.

NOBORDER
turns off the display of the graph wall border.

Tips To turn off the display of the wall's fill, use the NOWALL option.

To turn off the display of the graph's border, use the NOBORDER option in the ODS GRAPHICS statement.

NOSUBPIXEL | SUBPIXEL
specifies whether subpixel rendering should be used for rendering plots and charts. Subpixel rendering produces smoother curves and more precise bar spacing.

Defaults When neither option is specified, the system applies SUBPIXEL when it makes sense for the graph.

Subpixel rendering is always enabled for vector-graphics output.

Requirement Antialiasing must be enabled for the SUBPIXEL option to have any effect. Antialiasing is enabled by default. To re-enable antialiasing, use the ANTIALIAS=ON option in the ODS GRAPHICS statement.

Interaction If the SUBPIXEL option is explicitly set in the ODS GRAPHICS statement, that setting is used.

Tip For a large amount of data, antialiasing is disabled when the number of observations exceeds the default maximum of 4000 observations. In that case, subpixel rendering is also disabled. To increase the maximum, use the ANTIALIASMAX= option in the ODS GRAPHICS statement.

See "Subpixel Rendering" on page 1857

"ODS GRAPHICS Statement" on page 102 for information about the ANTIALIAS= and ANTIALIASMAX= options.

NOWALL
turns off the display of the graph wall's fill. This option might be useful when your graph contains an annotation, and the wall color interferes with that annotation.

Also, for many styles, the wall fill color is the same as the graph background. However, if that is not the case with the style that you use for a graph, then you might want to use this option.

Tip To turn off the display of the wall's border, use the NOBORDER option.

OPAQUE | NOOPAQUE
specifies whether the graph background is opaque or transparent.

Default OPAQUE

Restriction The following output formats support transparent background (NOOPAQUE): EMF, PDF, PNG, PS, and SVG, with the following exception. The PS format does not support transparent background when your output format is not vector graphics (that is, your output renders as an image due to some graph feature or you used an OUTPUTFMT= override to an image format, including PNG).

Interaction When NOOPAQUE is specified, the background color is not used.

PAD=*dimension<units>* | (*pad-options*)
specifies the amount of extra space to add to the left, right, top, and bottom of the graph. The extra space can help make the spacing more aesthetic, or provide blank space for custom annotations.

You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see Chapter 20, "Annotating ODS Graphics," on page 1935.

This option creates margins around the graph for company logos, annotated notes, and so on. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Use pad options to create non-uniform padding. Edges that are not assigned padding are padded with the default amount.

pad-options can be one or more of the following:

LEFT= ***numeric-value<units>***
specifies the amount of extra space to add to the left edge.

RIGHT= ***numeric-value<units>***
specifies the amount of extra space to add to the right edge.

TOP= ***numeric-value<units>***
specifies the amount of extra space to add to the top edge.

BOTTOM= ***numeric-value<units>***
specifies the amount of extra space to add to the bottom edge.

PCTLEVEL=BY | GRAPH | GROUP
specifies the scope of graph data that is calculated in percentages. When you calculate percentages using the STAT=PERCENT option, the calculation can be performed at different levels in the graph. The percentages within the selected level attempt to round up to 100%.

BY
the percentages within each BY-group round up to 100%

Interaction For this value to take effect, a BY-group must be specified in the procedure.

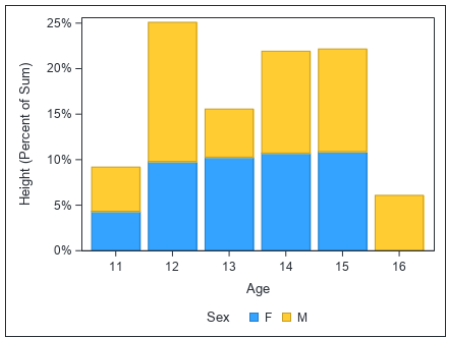
GRAPH
the percentages across the entire graph round up to 100%

GROUP
the percentages across groups within a category round up to 100%.

Interaction For this value to take effect, the GROUP= option must be specified in the plot statement.

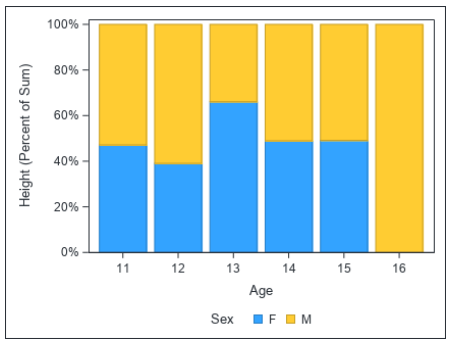
In the following examples, the first example specifies a value of GRAPH (the default), and the second example specifies a value of GROUP.

```
ods graphics on / reset=all;
ods graphics on / width=4.5in;
```



PCTLEVEL=GRAPH

```
proc sgplot data=sashelp.class pctlevel=graph;
  vbar age / response=height stat=percent
     group=sex;
run;
```



PCTLEVEL=GROUP

```
proc sgplot data=sashelp.class pctlevel=group;
  vbar age / response=height stat=percent
     group=sex;
run;
```

In the graph that specifies PCTLEVEL=GRAPH, the bars collectively add up to 100%. Each bar represents a fraction of the total.

In the graph that specifies PCTLEVEL=GROUP, each bar adds up to 100% of the category data represented by the bar. In this case, the categories are age.

Default GRAPH

Interaction For this option to take effect, STAT=PERCENT must be specified for a plot in the procedure.

Tip You can use the PCTNDEC= option in the SGPLOT procedure statement to control the number of decimals to be used when calculating the percent values. The default value is 1.

PCTNDEC=*numeric-value*

specifies the number of decimal spaces to be used to calculate the percent values.

Default The default number of decimals is based on the magnitude of the largest percentage value.

10% to 100% = 1	.1% to < 1% = 3
1% to < 10% = 2	< .1% = 4

Interaction For this option to take effect, STAT=PERCENT must be specified for a plot in the procedure.

Tip You can use the PCTLEVEL= option in the PROC SGPLOT statement to control the scope of the percent calculations.

RATTRMAP=*range-attribute-map-data-set*
specifies the range attribute map data set that you want to use with the procedure. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

Restriction The procedures do not support an attribute map that is stored in a table. Only SAS data sets are supported for attribute map data.

Requirement The values in the RATTRMAP data set must be sorted by ID. If they are not, only the first value is found.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SGANNO=*annotation-data-set*
specifies the SG annotation data set that you want to use. You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see Chapter 20, "Annotating ODS Graphics," on page 1935.

Note: The procedures do not support annotation data that is stored in a table. Only standard SAS data sets are supported for annotation data.

TMPLOUT="*filename*"
writes the Graph Template Language code for your graph to a file. Writing your template code to a file can be useful for building larger Graph Template Language definitions.

Restriction Graphs that contain summarized data (for example, bar, line, and dot) do not generate a useable template. For summarized data, the procedure sets internal values for one or more variables, and these internal values do not exist in the data set that is used with the procedure.

UNIFORM=GROUP | SCALE | ALL | XSCALE | YSCALE | XSCALEGROUP | YSCALEGROUP
specifies how to control axis scaling and legends when you use a BY statement.

Specify one of the following values:

ALL
specifies that both the legend group values and the axis scaling are common among all of the levels of the BY variable or variables.

GROUP
specifies that each group value in the legend uses the same visual attributes across all levels of the BY variable or variables.

This option is applicable only when the GROUP= option is specified for the plot.

SCALE
specifies that the axis scaling is shared among all of the levels of the BY variable or variables.

XSCALE
specifies that the X-axis scaling is shared among all of the levels of the BY variable or variables.

The Y axes are unaffected.

YSCALE
specifies that the Y-axis scaling is shared among all of the levels of the BY variable or variables.

The X axes are unaffected.

XSCALEGROUP
specifies that both the legend group values and the X-axis scaling are shared among all of the levels of the BY variable or variables.

The Y axes are unaffected. This option is applicable only when the GROUP= option is specified for the plot.

YSCALEGROUP
specifies that both the legend group values and the Y-axis scaling are shared among all of the levels of the BY variable or variables.

The X axes are unaffected. This option is applicable only when the GROUP= option is specified for the plot.

Default ALL

Restriction Basic bar charts (HBARBASIC and VBARBASIC) do not support uniform axis scaling. However, they do support group uniformity.

Interactions This option has no effect if you do not use a BY statement.

This option has no effect when the CATEGORYORDER= option is specified in the plot statement.

Uniform axis scaling does not apply to the gradient legend that is displayed when the COLORRESPONSE= option is specified in the plot statement. For a workaround that involves defining a range attribute map, see SAS Note SN-055810.

Note The options that affect legend group values do not fully populate every legend with all possible legend values. See the following tip.

Tip You can fully populate a legend by using a discrete attribute map. In the attribute map, specify the SHOW reserved attribute map variable. For more information, see Chapter 18, “Using Discrete Attribute Maps,” on page 1901 and “Reserved Discrete Attribute Map Variables” on page 1905.

STYLEATTRS Statement

Specifies visual attributes for a graph such as colors, markers, and so on, within the procedure step, without having to change the ODS style template.

Requirement: The procedure must include at least one plot statement.

Interaction: The following options apply only when the GraphData1–GraphData*n* style elements are used: DATACOLORS=, DATACONTRASTCOLORS=, DATAFILLPATTERNS=, DATALINEPATTERNS=, and DATASYMBOLS=. For more information, see "When the Style Options Take Effect" on page 904.

Tip: When specifying data attributes such as colors, symbols, and line patterns for multiple plots, the attributes might not be assigned consistently in some cases. To ensure consistent attribute assignments, specify the option NOCYCLEATTRS on page 891 in the SGPLOT procedure statement.

Syntax

STYLEATTRS <*options*>;

Summary of Optional Arguments

Axis options

AXISBREAK=BRACKET | NOTCH | SLANTEDLEFT | SLANTEDRIGHT | SQUIGGLE | SPARK | Z
: specifies a symbol to use on the axis lines to indicate a break in the axis.

AXISEXTENT=FULL | DATA
: specifies the extent of the axis line for all axes.

Background and wall options

BACKCOLOR=*color*
: specifies the background color of the graph area.

WALLCOLOR=*color*
: specifies the color of the plot wall area.

Style options

DATACOLORS=(*color-list*)
: specifies the fill colors for the graphics elements.

DATACONTRASTCOLORS=(*color-list*)
: specifies the contrast colors for the graphics elements, such as lines and markers.

DATAFILLPATTERNS=(*fill-pattern-list*)
: specifies the list of graph fill patterns.

DATALINEPATTERNS=(*line-pattern-list*)
specifies the list of line patterns for the graph data lines.

DATASYMBOLS=(*marker-symbol-list*)
specifies the list of marker symbols for the graph data.

Optional Arguments

AXISBREAK=BRACKET | NOTCH | SLANTEDLEFT | SLANTEDRIGHT | SQUIGGLE | SPARK | Z
specifies a symbol to use on the axis lines to indicate a break in the axis. The following figure shows an example of each symbol on a horizontal linear axis for ranges 1–4 and 6–10.

Figure 10.1 *Axis Break Symbols*

Default — When this option is not specified, the default break is a sine wave across the plot area.

Restriction — This option applies to linear and time axes only.

Requirements — For this option to have any effect, the DISPLAY= option for the axis must include the axis line.

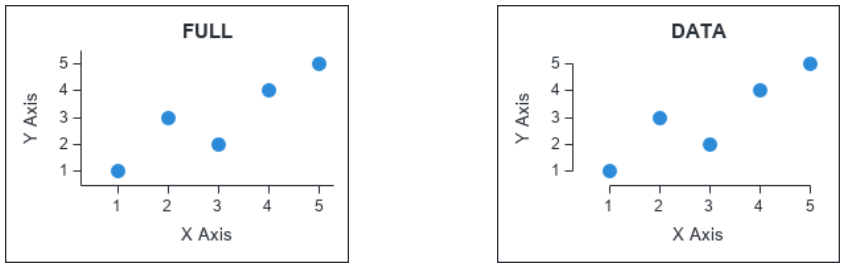
For this option to have any effect, the RANGES= option for the axis must be used to specify ranges for a broken axis.

AXISEXTENT=FULL | DATA
specifies the extent of the axis line for all axes. This option also turns off the wall border so that the axis lines can be clearly seen.

FULL
the axis lines extend along the entire length of the axis.

DATA
the axis lines extend through the data range from the minimum offset to the maximum offset.

The following figure shows a simple example of each value for the X- and Y-axis lines. Both axes specify a small minimum offset. For the DATA value, the axis lines extend through the data range from the minimum offset. No maximum offset was used in the example.



Default FULL

Tips Use the OFFSETMIN and OFFSETMAX axis options to specify the offsets.

Use the THRESHOLDMIN and THRESHOLDMAX axis options to specify a threshold for displaying one more tick mark at the low or high end of the axis.

BACKCOLOR=*color*

specifies the background color of the graph area. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The Color attribute of the GraphBackground style element

Examples backcolor=CXFF0000

backcolor=light_blue

DATACOLORS=(*color-list*)

specifies the fill colors for the graphics elements. The graphics elements can be in grouped plots or in overlaid multiple plots with the CYCLEATTRS feature in effect.

Provide a space-separated list of colors enclosed in parentheses. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The colors that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Requirement For this option to have an effect, at least one plot statement in the procedure must specify the GROUP= option.

Interaction Where applicable, the COLOR= suboption of any plot option related to fill color overrides the DATACOLORS= option.

Note When this option is specified, the colors cycle through *color-list* rather than the colors that are defined in the GraphData1–GraphData*n* style elements. When the colors in *color-list* are exhausted, the colors repeat.

Example datacolors=(CXFF0000 green blue)

DATACONTRASTCOLORS=(*color-list*)
specifies the contrast colors for the graphics elements, such as lines and markers. The lines and markers can be in grouped plots or in overlaid multiple plots with the CYCLEATTRS feature in effect.

Provide a space-separated list of colors enclosed in parentheses. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default — The contrast colors that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Requirement — For this option to have an effect, at least one plot statement in the procedure must specify the GROUP= option.

Interaction — Where applicable, the COLOR= suboption of any plot option related to a marker or line color overrides the DATACONTRASTCOLORS= option.

Note — When this option is specified, the colors cycle through *color-list* rather than the contrast colors that are defined in the GraphData1–GraphData*n* style elements. When the colors in *color-list* are exhausted, the colors repeat.

Example — datacontrastcolors=(orange cyan CXFF0000)

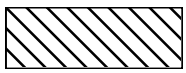
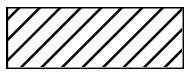
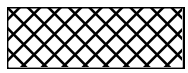
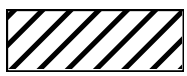
DATAFILLPATTERNS=(*fill-pattern-list*)
specifies the list of graph fill patterns.

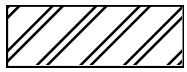
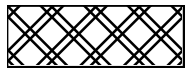
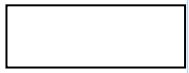
fill-pattern-list
a space-separated list of fill patterns, enclosed in parentheses.

You can use a style attribute reference such as GraphData3:fillpattern or a two-character fill-pattern name such as L1, X5, and so on, to specify a fill pattern. The list can contain a mix of style attribute references and pattern names.

To specify a two-character fill-pattern name, specify a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5. The following table shows the patterns for each of the possible combinations.

Table 10.3 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

When this option is specified, the fill patterns rotate through this fill-pattern list rather than through the fill patterns that are defined in the GraphData1–GraphData*n* style elements. For information about the attribute rotation patterns, see "How the Attributes Are Cycled" on page 904..

Default The fill patterns that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Requirement The list of fill patterns must be enclosed in parentheses.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Example
```
datafillpatterns=(graphdata4:fillpattern L5 X2)
```

DATALINEPATTERNS=(*line-pattern-list*)
specifies the list of line patterns for the graph data lines. Provide a space-separated list of line patterns enclosed in parentheses. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default The line patterns that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Requirement For this option to have an effect, at least one plot statement in the procedure must specify the GROUP= option.

Interaction Where applicable, the PATTERN= suboption of any plot option related to line attributes overrides the DATALINEPATTERNS= option.

Note When this option is specified, the line patterns cycle through *line-pattern-list* rather than the line patterns that are defined in the GraphData1–GraphData*n* style elements. When the patterns in *line-pattern-list* are exhausted, the patterns repeat.

Example datalinepatterns=(dot solid longdash 26)

DATASYMBOLS=(*marker-symbol-list*)
specifies the list of marker symbols for the graph data. Provide a space-separated list of symbols enclosed in parentheses. See marker symbols on page 1834.

Default The symbols that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Requirement For this option to have an effect, at least one plot statement in the procedure must specify the GROUP= option.

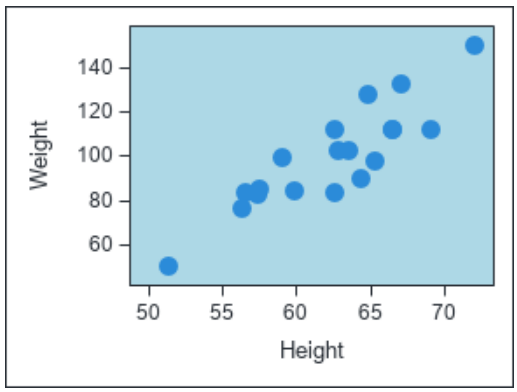
Interaction Where applicable, the SYMBOL= suboption of the MARKERATTRS= option overrides the DATASYMBOLS= option.

Note When this option is specified, the symbols cycle through *marker-symbol-list* rather than the symbols that are defined in the GraphData1–GraphData*n* style elements. When the symbols in *marker-symbol-list* are exhausted, the symbols repeat.

Example datasymbols=(circle square triangle star)

WALLCOLOR=*color*
specifies the color of the plot wall area. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

The following figure shows a light blue wall color.



Default The Color attribute of the GraphWalls style element

Examples wallcolor=CXFF0000

wallcolor=light_blue

Details

When the Style Options Take Effect

The following style options apply only when the GraphData1–GraphData*n* style elements are used: DATACOLORS=, DATACONTRASTCOLORS=, DATAFILLPATTERNS=, DATALINEPATTERNS=, and DATASYMBOLS=.

The corresponding attributes for the style options take effect when any one of the following is true:

- a GROUP= option is specified in at least one plot statement.
- the procedure contains at least two plot statements that cycle the respective attribute.
- a plot statement specifies a GraphData*n* style reference for the respective attribute (COLOR=). Example:

```
proc sgplot data=sashelp.class;
   styleattrs datacolors=(orange);
   vbar age / fillattrs=GraphData1;
run;
```

  In this case, DATACOLORS= overrides GraphData1:Color.

For more information, see the following blogs:

- Getting Started with SGPLOT: Style Attributes
- Attrs, attrs, everywhere: The interaction between ATTRPRIORITY, CYCLEATTRS, and STYLEATTRS in ODS Graphics

How the Attributes Are Cycled

The STYLEATTRS feature cycles the attributes in the order specified in your attribute list until the group values are exhausted. If the specified attribute list is exhausted first, then the list is repeated. If you specify multiple options, the STYLEATTRS feature combines the options as it cycles through.

In the following example, the DATACONTRASTCOLORS= option specifies three contrast colors used for lines. The DATALINEPATTERNS= option specifies two line patterns, and the DATASYMBOLS= option specifies two marker symbols.

```
proc sort data=sashelp.class out=class;
   by age;
run;
proc sgplot data=class;
   styleattrs
      datacontrastcolors=(red green blue)
      datalinepatterns=(dot solid)
      datasymbols=(circlefilled squarefilled);
   needle x=height y=weight / group=age markers;
   keylegend / across=3;
run;
```

In this example, the Illuminate style is used, so the procedure uses a color-priority rotation pattern.

With the color-priority rotation pattern, marker symbols and line patterns are held constant while each color in the list is applied to the marker symbol and line.

In the example, the dotted line pattern is held constant while the procedure applies red, green, and blue colors to the dotted lines and circlefilled marker symbols for the consecutive group values. If there are more group values, the options apply the red, green, and blue colors to solid lines and squarefilled markers.

The following figure shows an example rotation for an age grouping.

You can change this rotation pattern by using the ATTRPRIORITY= option in the ODS GRAPHICS statement. The ATTRPRIORITY= option controls the rotation pattern for the attributes derived from the GraphData1–GraphData*n* style elements for the style that is in effect.

Here is the general syntax for the option.

ATTRPRIORITY=COLOR | NONE

The default behavior for this option depends on the ODS style that is in use. For styles such as Illuminate and HTMLBlue, ATTRPRIORITY in the style is set to COLOR. For some of the other styles, ATTRPRIORITY in the style is set to NONE.

You can use the ATTRPRIORITY= option to control the rotation pattern for the attributes that you specify with the STYLEATTRS feature.

To change the rotation pattern in the previous example, specify the following before you invoke the procedure:

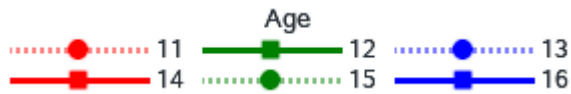
```
ods graphics / attrpriority=none;
```

Now, the following options in the STYLEATTRS statement are applied in alternating order:

```
proc sgplot data=class;
   styleattrs
      datacontrastcolors=(red green blue)
      datalinepatterns=(dot solid)
      datasymbols=(circlefilled squarefilled);
   needle x=height y=weight / group=age markers;
   keylegend / across=3;
run;
```

In the output, a red dotted line pattern and a red circlefilled marker are applied for the first group crossing, a solid green line pattern and a green squarefilled marker are applied for the second, a blue dotted line pattern and a blue circlefilled marker are applied for the third, and so on.

The following figure shows the rotation for the age grouping with no priority rotation.



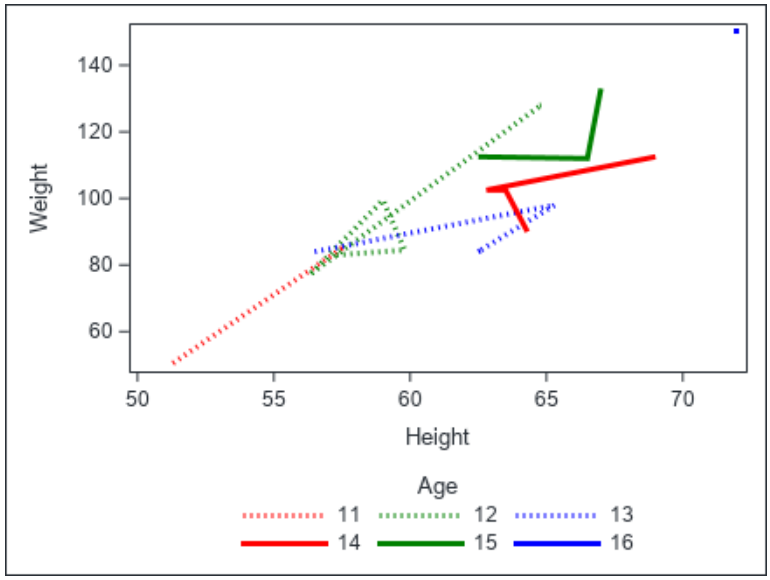
See Also

"ODS GRAPHICS Statement" on page 102

Example: Sorted Data with Contrast Colors and Line Patterns Specified

This example has the following features:

- The input data is sorted by the group variable.
- The ODS style is Illuminate. For the Illuminate style, attribute priority is color by default. Therefore, the procedure uses a color-priority rotation pattern to determine the output colors and line patterns.




```
/* Sort the data set by the group variable */
proc sort data=sashelp.class out=class;
  by age;
run;

/* Generate the graph using the sorted data */
proc sgplot data=class;
  styleattrs
     datacontrastcolors=(red green blue)
     datalinepatterns=(dot solid);
  series x=height y=weight / group=age;
  keylegend;
run;
```

BAND Statement

Creates a band that highlights part of a plot.

Restrictions: The axis that the UPPER and LOWER values are placed on cannot be a discrete axis. For example, if you specify a variable for Y, the plot cannot use a discrete horizontal axis.

The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

Note: The input data should be sorted by the X or Y variable. If the data is not sorted, the graph might produce unpredictable results.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Examples: "About Band Plots" on page 17

"Example 6: Creating Lines and Bands from Pre-Computed Data" on page 1666

Syntax

BAND X=*variable* | Y=*variable*
UPPER=*numeric-value* | *numeric-variable*
LOWER=*numeric-value* | *numeric-variable*
</option(s)>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

FILL

NOFILL
: specifies whether the area fill is visible.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

LINEATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the outlines for the band.

NOEXTEND
specifies that the band does not extend beyond the first and last data points in the plot.

OUTLINE

NOOUTLINE
specifies whether the outlines of the band are visible.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

TYPE=SERIES | STEP
specifies how the data points for the lower and upper band boundaries are connected.

Axes options

X2AXIS
assigns the variable that is assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS
assigns the variable that is assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABELATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the labels in the plot when you specify a curve label.

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELLOWER="*text-string*"
adds a label to the lower edge of the band.

CURVELABELPOS=AUTO | MIN | MAX | START | END
specifies the location of the curve label.

CURVELABELUPPER="*text-string*"
adds a label to the upper edge of the band.

LEGENDLABEL="*text-string*"
specifies a label that identifies the elements from the band plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text used for curve labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot reference options

MODELNAME="*plot-name*"
specifies the name of a plot that is used as a model for the interpolation for the band.

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

X=*variable*
Y=*variable*
specifies a variable that is used to plot the band along the X or Y axis.

LOWER=*numeric-value* | *numeric-variable*
specifies the lower value for the band. You can specify either a constant numeric value or a numeric variable.

UPPER=*numeric-value* | *numeric-variable*
specifies the upper value for the band. You can specify either a constant numeric value or a numeric variable.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you specify a curve label. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Interaction This option has no effect unless CURVELABELLOWER or CURVELABELUPPER is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family=Arial Size=8
     Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default INSIDE

CURVELABELLOWER="*text-string*"
adds a label to the lower edge of the band. Specify the label text.

Restriction This option cannot be used with the GROUP= option. The label specified is not drawn when the GROUP= option is in effect.

Interaction If the plot uses a broken linear or time axis, the curve label might not align properly with the band in some cases.

CURVELABELPOS=AUTO | MIN | MAX | START | END
specifies the location of the curve label. Specify one of the following values:

AUTO
places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction This value takes effect only when CURVELABELLOC=OUTSIDE.

MIN
places the label at the part of the curve closest to the minimum X-axis value.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

START
places the curve label at the first point on the curve.

END
places the curve label at the last point on the curve.

Default END

Interactions This option has no effect unless the CURVELABELLOWER= or CURVELABELUPPER= option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

CURVELABELUPPER="*text-string*"
adds a label to the upper edge of the band. Specify the label text.

Restriction This option cannot be used with the GROUP= option. The label specified is not drawn when the GROUP= option is in effect.

Interaction If the plot uses a broken linear or time axis, the curve label might not align properly with the band in some cases.

FILL | NOFILL
specifies whether the area fill is visible. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default The default status of the area fill is specified by the DisplayOpts attribute of the GraphBand style element in the current style.

Interactions Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips: Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

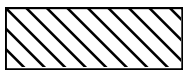
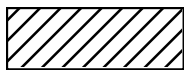
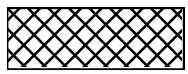
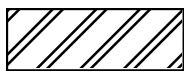
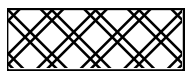
COLOR=*color*
specifies a color to use for the fill lines.

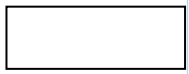
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.4 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

GROUP=*variable*
specifies a variable that is used to group the data. A separate band is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

LEGENDLABEL="*text-string*"
specifies a label that identifies the elements from the band plot in the legend. By default, the label “band“ is used for ungrouped data, and the group values are used for grouped data.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the outlines for the band. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see “Line Attributes and Patterns” on page 1828.

Defaults For ungrouped data, the ContrastColor and LineStyle, and LineThickness attributes of the GraphConfidence style element in the current style.

For grouped data, the lineThickness attribute of the GraphConfidence style element, and the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless you also specify the OUTLINES option.

MODELNAME="*plot-name*"
specifies the name of a plot that is used as a model for the interpolation for the band.

Default If you do not specify this option, then the band is interpolated in the same way as a series plot.

Requirement *plot-name* must be the name that has been assigned with the associated plot's NAME= option.

See "Details" on page 918

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOEXTEND
specifies that the band does not extend beyond the first and last data points in the plot. By default, the band extends to the edges of the plot area.

Interaction This option has no effect if you do not specify numeric values for the UPPER= and LOWER= options.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

OUTLINE | NOOUTLINE
specifies whether the outlines of the band are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default The default status of the band outlines is specified by the DisplayOpts attribute of the GraphBand.

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text used for curve labels is always split. The text is split at every occurrence of the specified split character or characters. This option affects both the upper and lower curve labels if they are displayed.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Interactions This option has no effect unless either CURVELABELLOWER or CURVELABELUPPER is also specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See “Fit Policies for Axis Tick Values, Curve Labels, and Data Labels” on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See “Fit Policies for Axis Tick Values, Curve Labels, and Data Labels” on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

TYPE=SERIES | STEP
specifies how the data points for the lower and upper band boundaries are connected. You can specify one of the following:

SERIES
the data points are connected directly using line segments, as in a series plot.

STEP
the data points are connected using a step function, as in a step plot.

Default SERIES

X2AXIS
assigns the variable that is assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS
assigns the variable that is assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

Details

The MODELNAME= option fits a band to another plot. This is particularly useful for plots that use a special interpolation such as step plots.

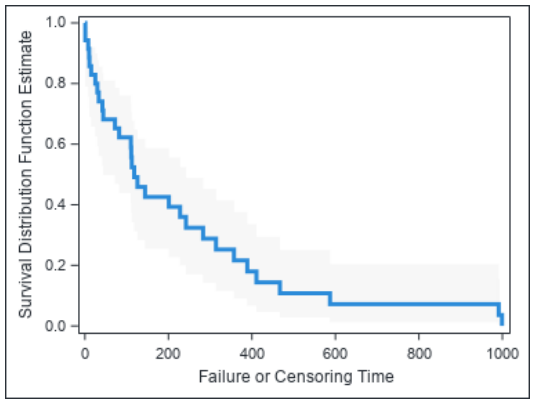
The following code fits a band to a step plot:

```
proc lifetest data=VALung outsurv=survival maxtime=600
   plots=none noprint notable;
   time SurvTime*Censor(1);
   id Therapy;
   strata Cell;
   test Age;
run;

proc sgplot data=survival NOAUTOLEGEND;
   band x=SurvTime upper=sdf_ucl lower=sdf_lcl / modelname="myname"
transparency=.5;
   step x=SurvTime y=survival / name="myname";
   where cell eq "squamous";
run;
```

Note: For information about the LIFETEST procedure and data set VALung, see The LIFETEST Procedure and example Product-Limit Estimates and Tests of Association in *SAS/STAT User's Guide*.

Here is the output.

Figure 10.2 *Fitted Band Plot Example*



BLOCK Statement

Creates one or more rectangular blocks containing text values.

Notes: The block widths are determined by a specified block variable.

Each block represents a block variable value and spans the data that has the same block variable value.

When data labels are used, the axis offsets are extended to accommodate the length of the longest data label. In that case, the blocks at each end of the block plot are extended to include the additional axis offset. This behavior is expected.

The BLOCK statement is typically used with another plot statement, which establishes the axis type for the Y axis. However, the BLOCK statement can be used by itself. In that case, the block plot has no Y axis.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Block Plots" on page 18

Syntax

BLOCK X=*category-variable* BLOCK=*block-variable* </*option(s)*>;

Summary of Optional Arguments

Appearance options

ALTFILLATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of alternate fills for the blocks.

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

FILL

NOFILL
: specifies whether the blocks are filled.

FILLATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the fill for the blocks.

FILLTYPE=MULTICOLOR | ALTERNATE
: specifies how the blocks are filled.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the block outlines.

OUTLINE

NOOUTLINE
: specifies whether the blocks have outlines.

TRANSPARENCY=*numeric-value*
: specifies the degree of transparency for the blocks.

Axis options

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

Block options

POSITION=BOTTOM | CENTER | TOP
: positions the block plot at the bottom, center, or top of the graph.

Block text options

BLOCKLABEL=*variable*
: specifies a column to use for alternative text in the blocks.

NOVALUES

VALUES
: specifies whether the block values are displayed or hidden.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the block text values are split according to the fit policy that is in effect.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the block text values.

VALUEFITPOLICY=NONE | SHRINK | SPLIT | SPLITALWAYS | TRUNCATE
: specifies how text values are adjusted to fit within the containing block.

VALUEHALIGN=LEFT | CENTER | RIGHT | START
: specifies the horizontal alignment of the value text within the blocks.

VALUEVALIGN=TOP | CENTER | BOTTOM
: specifies the vertical alignment of the value text within the blocks.

Label options

LABEL

LABEL="*text-string*"

NOLABEL
: specifies an external label for a single block plot.

LABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the color and font attributes of the external block label(s).

LABELPOS=BOTTOM | LEFT | RIGHT | TOP
: specifies the position for the block label for a single block plot.

Plot options

CLASS=*variable*
: creates a stack of block plots, with one block plot for each unique value of the specified variable.

EXTENDMISSING
: extends the previous block value if the current value is missing.

NOMISSINGCLASS
: suppresses blocks that correspond to missing values of the CLASS= value.

Plot reference options

NAME="*text-string*"
: assigns a name to a plot statement.

Required Arguments

X=*category-variable*
specifies X-axis positions. When the X axis is numeric and the specified variable is numeric, values are expected to be in sorted, ascending order. If the X axis is discrete and the specified column is numeric, values are treated as numeric-discrete.

BLOCK=*block-variable*
specifies the variable that classifies the observations into distinct subsets.

Optional Arguments

ALTFILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of alternate fills for the blocks. This option in conjunction with the FILLATTRS= option controls fill appearance when FILLTYPE=ALTERNATE. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Requirement FILLTYPE=ALTERNATE must be specified for this option to have any effect.

Interaction This option has no effect if you specify the NOFILL option.

Note The TRANSPARENCY= attribute affects only the alternate fill blocks. To set the same transparency for both the fill blocks and the alternate fill blocks, use the TRANSPARENCY= option in the BLOCK statement.

Tips The FILLATTRS= option controls the fill color of non-alternate blocks.

To make all block fill areas the same color, set the FILLATTRS= and ALTFILLATTRS= options to the same value.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BLOCKLABEL=*variable*
specifies a column to use for alternative text in the blocks.

Interaction This option overrides the default text for the blocks.

Tip The font and color attributes for the alternative text are specified by the VALUEATTRS= option.

CLASS=*variable*
creates a stack of block plots, with one block plot for each unique value of the specified variable.

Interaction To label the blocks by the class values, specify the BLOCKLABEL= option using the same class variable.

Note For local data, the visual attributes for each class value are assigned in data order. However, if you are using data that is stored in a table, the visual attributes for each class value are assigned in ascending order of the class variable character values or of unformatted numeric values.

EXTENDMISSING
extends the previous block value if the current value is missing.

FILL | NOFILL
specifies whether the blocks are filled.

Default FILL

Interaction The NOFILL option can be used with the NOOUTLINE option to hide both the fill and the outline.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fill for the blocks. This option in conjunction with the ALTFILLATTRS= option controls fill appearance when FILLTYPE=ALTERNATE. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Requirement FILLTYPE=ALTERNATE must be specified for this option to have any effect.

Interaction This option has no effect if you specify the NOFILL option.

Note The TRANSPARENCY= attribute affects only the fill blocks. To set the same transparency for both the fill blocks and the alternate fill blocks, use the TRANSPARENCY= option in the BLOCK statement.

Tips The ALTFILLATTRS= option controls the alternate fill color.

To make all block fill areas the same color, set the FILLATTRS= and ALTFILLATTRS= options to the same value.

FILLTYPE=MULTICOLOR | ALTERNATE
specifies how the blocks are filled.

MULTICOLOR
Blocks are filled with the COLOR attribute of the GraphData1–GraphData*n* style elements.

ALTERNATE
Blocks are filled alternating between the colors specified by the FILLATTRS= and ALTFILLATTRS= options.

Default MULTICOLOR

Interaction This option has no effect if NOFILL is also specified.

LABEL
LABEL="*text-string*"
NOLABEL
specifies an external label for a single block plot. If you specify the LABEL option, the procedure displays the name of the block variable. You can override that label by providing your own text.

Default The label text is the variable label of the BLOCK= variable. If there is no variable label, the variable name is used.

Interactions By default, the label appears to the left of the plot. You can specify a different position for the label using the LABELPOS= option.

You can specify the font and color attributes for the label using the LABELATTRS= option.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the color and font attributes of the external block label(s). You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LABELPOS=BOTTOM | LEFT | RIGHT | TOP
specifies the position for the block label for a single block plot.

Default LEFT

Interaction This option has no effect if NOLABEL is also specified.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the block outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interactions This option has no effect if the NOOUTLINE option is also specified.

The external labels are also outlined when LABEL= is TOP or BOTTOM. In that case, the LINEATTRS= option also controls the appearance of the external label outlines.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGCLASS
suppresses blocks that correspond to missing values of the CLASS= value.

OUTLINE | NOOUTLINE
specifies whether the blocks have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

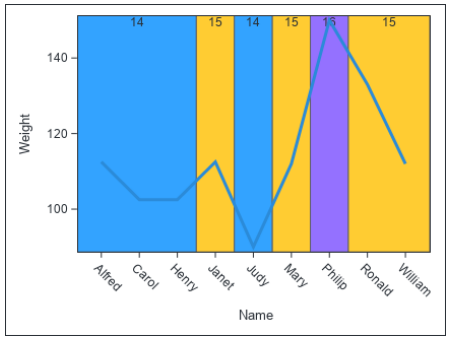
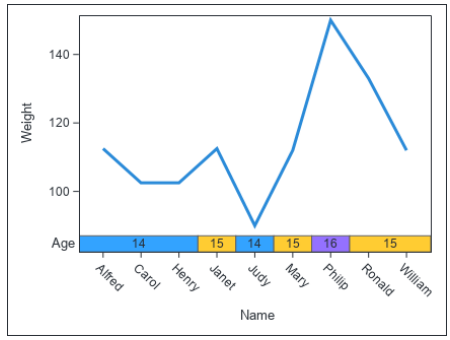
Interaction The NOOUTLINE option can be used with the NOFILL option to hide both the outline and the fill.

NOVALUES | VALUES
specifies whether the block values are displayed or hidden.

Default VALUES

Tip The font and color attributes for the text values are specified by the VALUEATTRS= option.

POSITION=BOTTOM | CENTER | TOP
positions the block plot at the bottom, center, or top of the graph. This option is useful when the graph contains more than one plot.

POSITION=CENTER	POSITION=BOTTOM
	

Default CENTER

SPLITCHAR="*character-list*"

specifies one or more characters on which the block text values are split according to the fit policy that is in effect. When VALUEFITPOLICY=SPLIT is in effect, the block text values are split on the specified character or characters where required to fit the block text within the containing block. When VALUEFITPOLICY=SPLITALWAYS is in effect, the block text values are split on every occurrence of the specified character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur. In that case, if the value does not fit the available space, it might collide with the values in the adjacent blocks.

Default The default split character is a space.

Interactions This option has no effect unless VALUEFITPOLICY= specifies SPLIT or SPLITALWAYS. The default value for VALUEFITPOLICY= is SPLIT.

When the text value is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the blocks. Transparency affects both the fill and the alternate fills, if enabled. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default 0.0

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the block text values. This option affects the default values that are associated with the BLOCK= argument. If BLOCKLABEL= is specified, then the option affects those values instead.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

VALUEFITPOLICY=NONE | SHRINK | SPLIT | SPLITALWAYS | TRUNCATE
specifies how text values are adjusted to fit within the containing block. Select one of the following values:

NONE
No attempt is made to fit values that collide with the text values in adjacent blocks.

SHRINK
All values are reduced in font size until they all fit.

SPLIT
If a value does not fit within the containing block, it is split at a split character.

No split occurs at split characters that occur where a split is not needed. In that case, the split character is displayed with the text value.

If the value does not contain any of the specified split characters, a split does not occur. In that case, if the value does not fit the available space, it might collide with the adjoining values.

Default The default split character is a space.

Tip Use the SPLITCHAR= option to specify a split character.

SPLITALWAYS
Text values are split at a split character in all blocks.

Default The default split character is a space.

Tip Use the SPLITCHAR= option to specify a split character.

TRUNCATE
Any value that does not fit is truncated.

For a numeric column, an asterisk (*) is substituted for the entire value whenever truncation occurs. For a character column, the truncated portion of the text is replaced by an ellipsis (...).

Default SPLIT

VALUEHALIGN=LEFT | CENTER | RIGHT | START
specifies the horizontal alignment of the value text within the blocks. This option changes the text alignment regardless of whether you split values or specify the VALUEFITPOLICY= option.

Select one of the following values:

LEFT
left-aligned within the block

CENTER
center-aligned within the block

RIGHT
right-aligned within the block

START
center-aligned at the starting value of the block

Default CENTER

Interaction This option has no effect if you also specify the NOVALUES option.

VALUEVALIGN=TOP | CENTER | BOTTOM
specifies the vertical alignment of the value text within the blocks.

Default TOP

Interaction This option has no effect if you also specify the NOVALUES option.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Details

Changing the Appearance of Block Text Values

By default, values for the variable that is specified in the BLOCK= argument appear as text in the blocks. You have several options for changing the text or its appearance.

The following table lists the tasks that you can perform along with the options that are involved in performing the task:

Table 10.5 *Tasks for Changing the Appearance of Block Text*

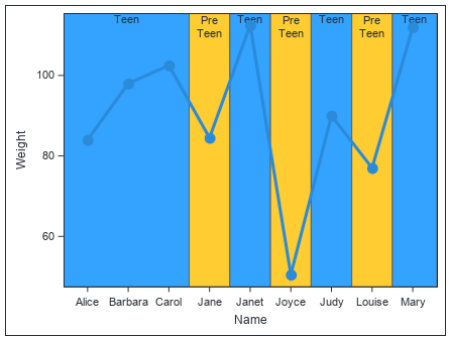
Task	Options
hide or show the values	NOVALUES or VALUES
use a different variable for alternative text in the blocks	BLOCKLABEL=
change the font characteristics of the text	VALUEATTRS=
align the text horizontally	VALUEHALIGN=
align the text vertically	VALUEVALIGN=
specify how text values are adjusted to fit within the containing block	VALUEFITPOLICY=
for text that is split into multiple lines within the block, specify either or both of the following: ■ one or more characters used to split the text value ■ that the split characters are included in the displayed value	SPLITCHAR= SPLITCHARNODROP

The following DATA step creates a SPEC column with two values: Teen and Pre-Teen.

```
data class1;
 set sashelp.class;
 where sex="F";
 length spec $10;
 if age > 12 then spec='Teen';
 else spec='Pre-Teen';
run;
```

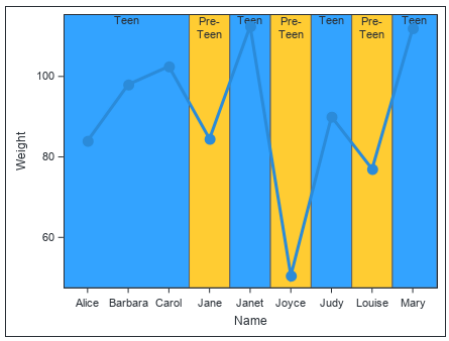
The SPEC column is used for the blocks in the examples below. The value fit policy is specified as VALUEFITPOLICY=SPLIT. Based on this policy, the text values split if they do not fit within their block. The examples split the block text on the “-” character.

In the first example, the split character does not appear in the blocks.




```
proc sgplot data=class1;
  block x=name block=spec /
    valuefitpolicy=split
    splitchar="-";
  series x=name y=weight / markers
    markerattrs=(size=10pt);
run;
```

The next example specifies the SPLITCHARNODROP option. This option forces the split character to appear in the graph.




```
proc sgplot data=class1;
  block x=name block=spec /
    valuefitpolicy=split
    splitchar="-"
    splitcharnodrop;
  series x=name y=weight / markers
    markerattrs=(size=10pt);
run;
```

Notice that the series plot does not contrast well against the block plots. There are two simple changes that you can make to improve the graph:

- Position the block plot above or below the series plot. The POSITION= option positions the block plot at the bottom, center, or top of the graph. This option is available only with the SGPLOT procedure.
- Change the display attributes of the block plot or the series plot. The following section provides more information about block plot display attributes.

Changing the Appearance of Block Fills

If you want to change the appearance of the fills, you have several options.

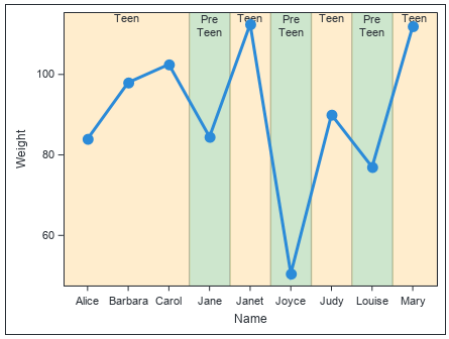
The following table lists the tasks that you can perform along with the options that are involved in performing the task:

Table 10.6 *Tasks for Changing the Appearance of Block Fills*

Task	Option
change the transparency of the fills for all blocks	TRANSPARENCY=
change the color or transparency for the fills when you are alternating fill colors	FILLTYPE= FILLATTRS=

Task	Option
	ALTFILLATTRS=

For example, the previous example in "Changing the Appearance of Block Text Values" on page 928 showed a block plot overlaid on a series plot. For that example, you might want the block fills to provide more contrast with the series plot. The following example builds on the previous plot and specifies colors and transparency for alternating fills.




```
proc sgplot data=class1;
 block x=name block=spec /
   valuefitpolicy=split
   splitchar="-"
   filltype=alternate
   fillattrs=(color=orange
              transparency=0.75)
   altfillattrs=(color=green
                 transparency=0.75);
 series x=name y=weight / markers
   markerattrs=(size=10pt);
run;
```

The FILLTYPE=ALTERNATE option alternates the block fills. You can then specify a fill color for the primary and alternate fills using the FILLATTRS= option and the ALTFILLATTRS= option, respectively. Finally, the example changes the transparency of the block plot to 80%.

BUBBLE Statement

Creates a bubble plot in which two variables determine the location of the bubble centers and a third variable controls the size of the bubble.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Syntax

BUBBLE X=*variable* Y=*variable* SIZE=*numeric-variable* </option(s)>;

Summary of Optional Arguments

Appearance options

ABSSCALE

specifies that the SIZE= column values are interpreted in the same units as the axes rather than as relative values.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

BRADIUSMAX=*numeric-value<units>*
specifies the size of the radius of the largest bubble.

BRADIUSMIN=*numeric-value<units>*
specifies the size of the radius of the smallest bubble.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DRAWORDER=SIZE | DATA
specifies whether the bubbles are drawn according to bubble size or according to data order.

FILL

NOFILL
specifies whether the bubbles are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency.

FILLPATTERN
enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern.

LINEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the outlines for the bubbles.

OUTLINE

NOOUTLINE
specifies whether the outlines of the bubbles are visible.

PROPORTIONAL
specifies that proportional scaling be applied to the SIZE= column values.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

X2AXIS
assigns the variables that are assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS
assigns the variables that are assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
: specifies a variable that is used to group the data.

NOMISSINGGROUP
: specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL

DATALABEL=*variable*
: displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
: specifies the location of each data label with respect to its data point.

LEGENDLABEL="*text-string*"
: specifies the label that identifies the bubble plot in the legend.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

Plot options

URL=*character-variable*
: specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
: assigns a name to a plot statement.

Required Arguments

X=*variable*
: specifies the variable for the X axis.

Y=*variable*
specifies the variable for the Y axis.

SIZE=*numeric-variable*
specifies the variable that controls the size of the bubbles. The minimum and maximum values automatically provide the range that is used to determine bubble sizes. You can control this range manually by using the BRADIUSMAX and BRADIUSMIN options.

Optional Arguments

ABSSCALE
specifies that the SIZE= column values are interpreted in the same units as the axes rather than as relative values. By default, the bubble sizes are scaled to represent the value range of the SIZE= column. For example, if only two bubbles are drawn with sizes of 2 and 4, then they appear the same as only two bubbles with sizes of 4000 and 8000. By contrast, when ABSSCALE is specified, the size values are interpreted in the same units as the axes.

Restriction This option is ignored if the X or Y axis is discrete.

Interaction When this option is used, the BRADIUSMAX= and BRADIUSMIN= options are ignored.

Note The bubbles might be drawn as ellipses if the X and Y axes are scaled differently.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BRADIUSMAX=*numeric-value<units>*
specifies the size of the radius of the largest bubble. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Restriction The BRADIUSMAX= value must be greater than the BRADIUSMIN= value if one is specified. If BRADIUSMAX is not greater, an error is generated and a message is written to the SAS log.

Note If you specify the maximum size as a percentage, this is interpreted as a percentage of the graph's height.

BRADIUSMIN=*numeric-value<units>*
specifies the size of the radius of the smallest bubble. You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Restriction The BRADIUSMIN= value must be smaller than the BRADIUSMAX= value if one is specified. If BRADIUSMIN is not smaller, an error is generated and a message is written to the SAS log.

Note If you specify the minimum size as a percentage, this is interpreted as a percentage of the graph's height.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorRamp style element for filled bubbles. The ThreeColorAltRamp style element for unfilled bubbles.

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on the plot. The data skin affects all bubbles. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.7 *DATASKIN Options for Bubbles*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DRAWORDER=SIZE | DATA

specifies whether the bubbles are drawn according to bubble size or according to data order.

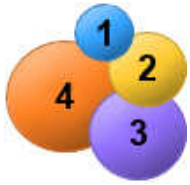
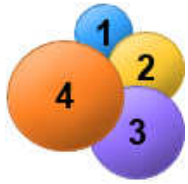
SIZE

draws the bubbles according to bubble size, from the largest to the smallest.

DATA

draws the bubbles according to data order.

The following figure shows the effect of SIZE and DATA on four bubbles. The bubble labels indicate the data order, and the bubble sizes increase linearly starting with 1.

DRAWORDER=SIZE	DRAWORDER=DATA
	

Default SIZE

FILL | NOFILL
specifies whether the bubbles are filled. The FILL option shows the fill color. The NOFILL option hides the fill color.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips: Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

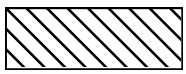
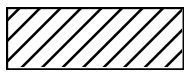
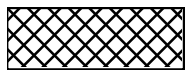
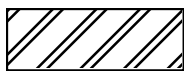
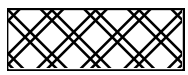
COLOR=*color*
specifies a color to use for the fill lines.

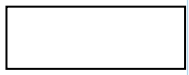
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.8 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

GROUP=*variable*
specifies a variable that is used to group the data. The bubbles for each group value are automatically distinguished by different colors.

When this option is used, the bubble colors are taken from the GraphData1... GraphData*n* style elements. If the bubbles are filled, then the COLOR attribute is used for bubble fill and CONTRASTCOLOR is used for the bubble outline. If the bubbles are not filled, then the CONTRASTCOLOR and PATTERN are used for the bubble outlines.

When this option is used and the value is a variable associated with an ATTRID= option, the attribute mapping defined by the associated attribute map is used.

Interaction This option is ignored if the COLORRESPONSE= option is also used.

Tip When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bubble plot in the legend.

Default By default, the label for the Y variable is used for ungrouped data, and the group values are used for grouped data.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the outlines for the bubbles. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the GraphOutlines style element in the current style.

For grouped data:

- if only outlines are displayed, LineThickness from GraphOutlines, and ContrastColor and LineStyle from a GraphData1–GraphData*n* style element in the current style
- if outlines and fill are displayed, LineThickness and LineStyle from GraphOutlines, and ContrastColor from a GraphData1–GraphData*n* style element in the current style

Interactions This option has no effect if you also specify the NOOUTLINE option.

This option is ignored if the DATASKIN= option is also specified.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OUTLINE | NOOUTLINE
specifies whether the outlines of the bubbles are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

PROPORTIONAL
specifies that proportional scaling be applied to the SIZE= column values. By default, scaling is linear, which increases the size of the bubbles in linear proportion to the range of the SIZE= column values. For example, if only two bubbles are drawn with sizes of 2 and 4, they appear the same as only two bubbles with sizes of 4000 and 8000. When PROPORTIONAL is specified, each bubble is sized in direct proportion to its corresponding SIZE= column value. For example, if only two bubbles are drawn with sizes of 50 and 100, then the SIZE=50 bubble is drawn to half the size of the SIZE=100 bubble.

Default linear

Interactions If all the values for the SIZE= column are negative, then PROPORTIONAL is ignored and the scaling is linear.

When PROPORTIONAL and BRADIUSMIN= are specified, the BRADIUSMIN= option specifies the minimum bubble size. In that case, when a SIZE= column value results in a bubble of a size that is less than the BRADIUSMIN= value, the bubble size for that value is changed to the BRADIUSMIN= value.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, “Using Range Attribute Maps,” on page 1921

“Overview of Attribute Maps” on page 1897

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the variables that are assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS
assigns the variables that are assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

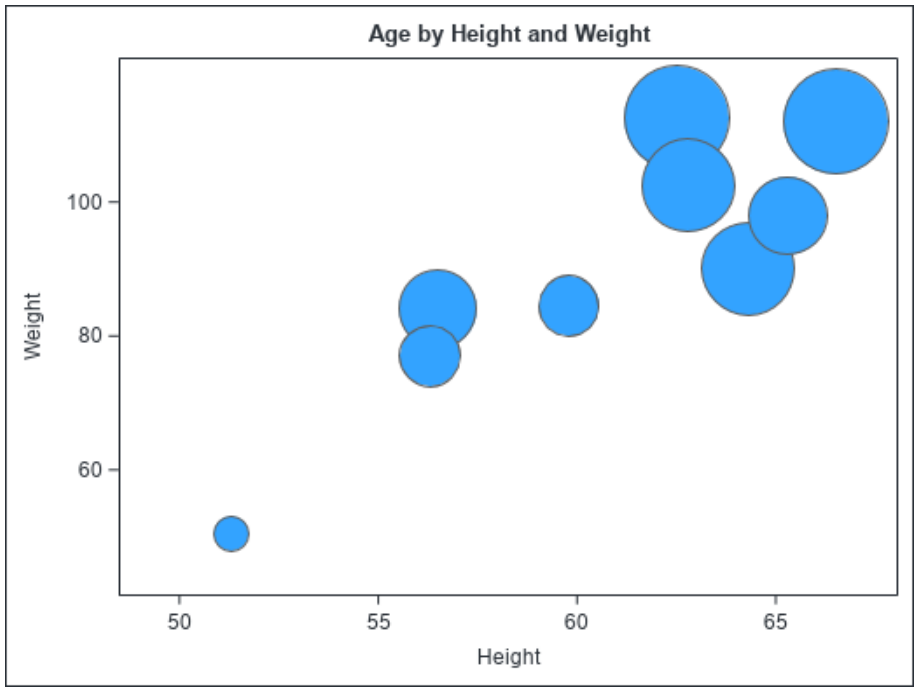
Examples

Example 1: Creating a Basic Bubble Plot

This example creates a simple bubble plot of weight and height where age is represented by bubble size.

```
title "Age by Height and Weight";
proc sgplot data=sashelp.class(where=(sex='F'));
   bubble x=height y=weight size=age;
run;
```

Here is the result.



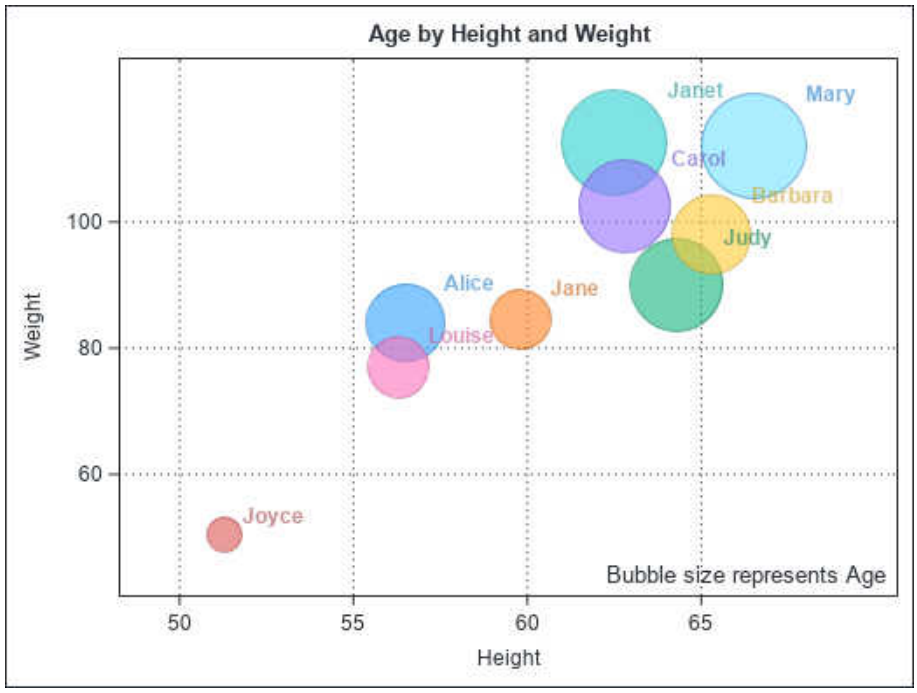
Example 2: Enhancing a Bubble Plot

This example enhances the previous basic bubble plot by adding colors, labels, an inset, and other visual features.

```
title 'Age by Height and Weight';
proc sgplot data=sashelp.class(where=(sex='F'))
   noautolegend;                              /* 1 */
   bubble x=height y=weight size=age /        /* 2 */
      group=name
      datalabel=name
      transparency=0.4
      datalabelattrs=(size=9 weight=bold);
   inset "Bubble size represents Age" /       /* 3 */
      position=bottomright
      textattrs=(size=11);
   yaxis grid;                                /* 4 */
   xaxis grid;
run;
```

1. The NOAUTOLEGEND on page 892 option in the PROC SGPLOT statement suppresses the automatically generated legend.
2. In the BUBBLE statement:
   - The GROUP= on page 939 option assigns a color to the bubble for each name.
   - The DATALABEL= on page 935 option specifies the value of variable Name as the label for each bubble.
   - The TRANSPARENCY= on page 944 option specifies the transparency for the bubble fill.
   - The DATALABELATTRS= on page 935 option specifies the attributes for the bubble labels.
3. The INSET on page 1166 statement defines an inset for the graph. The POSITION= on page 1169 option specifies the bottom right corner as the position. The TEXTATTRS= on page 1169 option specifies the size of the inset text.
4. The YAXIS on page 1587 and XAXIS on page 1524 statements specify the axis options. The GRID option specifies grid lines at the major tick marks on each axis.

Here is the result.



DENSITY Statement

Creates a density curve that shows the distribution of values for a numeric variable.

Restrictions: The DENSITY statement cannot be used with discrete axes.

The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

Interaction: The DENSITY statement can be combined only with DENSITY and HISTOGRAM statements in the SGPLOT procedure. See "Plot Compatibility" on page 1780.

Examples: "About Density Plots" on page 49

"Example 8: Combining Histograms with Density Plots" on page 1669

Syntax

DENSITY *response-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the density line.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

TYPE=NORMAL < (***normal-opts***)> | KERNEL < (***kernel-opts***)>
: specifies the type of distribution curve that is used for the density plot.

Axis options

SCALE=COUNT | DENSITY | PERCENT | PROPORTION
: specifies the scaling that is used for the response axis.

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

Group options

GROUP=*variable*
: specifies a variable that is used to group the data.

Label options

CURVELABEL="*text-string*"
: adds a label for the density curve.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC=OUTSIDE | INSIDE

specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label.

LEGENDLABEL="*text-string*"
specifies a label that identifies the density plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations.

Y2AXIS
assigns the calculated result to the secondary (right) vertical axis.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Argument

response-variable
specifies the variable for the X axis. The variable must be numeric.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CURVELABEL="*text-string*"
adds a label for the density curve.

Restriction This option is ignored when the GROUP= option is specified.

Interaction If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default INSIDE

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label. Specify one of the following values:

AUTO
places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction This value takes effect only when CURVELABELLOC=OUTSIDE.

END
places the curve label at the last point on the curve.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

MIN
places the label at the part of the curve closest to the minimum X-axis value.

START
places the curve label at the first point on the curve.

Defaults END if CURVELABELLOC=INSIDE

AUTO if CURVELABELLOC=OUTSIDE

Interactions This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

LEGENDLABEL="*text-string*"
specifies a label that identifies the density plot in the legend. By default, the label identifies the type of density curve. If you specify TYPE=NORMAL, then the default label is “Normal.“ If you specify TYPE=KERNEL, then the default label is “Kernel.“

Note User-specified parameters from the TYPE= option are included in the label by default.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the density line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see “Line Attributes and Patterns” on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphFit style element in the current style.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

SCALE=COUNT | DENSITY | PERCENT | PROPORTION
specifies the scaling that is used for the response axis. Specify one of the following values:

COUNT
the axis displays the frequency count.

DENSITY
the axis displays the density estimate values.

PERCENT
the axis displays values as a percentage of the total.

PROPORTION
the axis displays values in proportion to the total.

Defaults When the DENSITY statement follows a HISTOGRAM statement, the density plot uses the histogram's response-axis scaling.

When the DENSITY statement does not follow a HISTOGRAM statement, the density plot uses DENSITY response-axis scaling.

Interaction This option is ignored when the DENSITY statement follows a HISTOGRAM statement. In that case, the density plot uses the histogram's response-axis scaling instead of the scaling specified in this option. See SCALE= on page 1145 for the HISTOGRAM statement.

Note The PROPORTION scale can be used only when you combine a density plot and a histogram.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless CURVELABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

TYPE=NORMAL < (*normal-opts*)> | KERNEL < (*kernel-opts*)>
specifies the type of distribution curve that is used for the density plot. Specify one of the following keywords:

NORMAL < (*normal-opts*)>
specifies a normal density estimate, with a mean and a standard deviation.

normal-opts can be one or more of the following values:

MU=*numeric-value*
specifies the mean value that is used in the density function equation.

By default, the mean value is calculated from the data.

SIGMA=*numeric-value*
specifies the standard deviation value that is used in the density function equation.

The value that you specify for the SIGMA= suboption must be a positive number. By default, the standard deviation value is calculated from the data.

KERNEL < (*kernel-opts*)>
specifies a nonparametric kernel density estimate.

kernel-opts can be:

C=*numeric-value*
specifies the standardized bandwidth for a number that is greater than 0 and less than or equal to 100.

The value that you specify for the C= suboption affects the value of λ as shown in the following equation:

$$\lambda = cQn^{-\frac{1}{5}}$$

In this equation *c* is the standardized bandwidth, *Q* is the interquartile range, and *n* is the sample size.

WEIGHT=NORMAL | QUADRATIC | TRIANGULAR
specifies the weight function.

You can specify either normal, quadratic, or triangular weight function.

Default NORMAL

Note The kernel density function uses the same estimation that is used by the UNIVARIATE procedure. This estimation minimizes the Asymptotic Mean Integrated Square Error (AMISE) for the kernel density function.

Default NORMAL

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the calculated result to the secondary (right) vertical axis.

Details

Normal Density Function

When the type of the density curve is NORMAL, the fitted density function equation is as follows.

$$p(x) = \frac{100h\,\%}{\sigma\sqrt{2\,\pi}}\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \text{ for } -\infty < x < \infty$$

In the equation, μ is the mean, and σ is the standard deviation. You can specify μ by using the MU= suboption and σ by using the SIGMA= suboption.

Kernel Density Function

When the TYPE of the density curve is KERNEL, the general form of the kernel density estimator is as follows.

$$\widehat{f_\lambda}(x) = \frac{100h\,\%}{n\lambda}\sum_{i=1}^{n} K_0\left(\frac{x-x_i}{\lambda}\right)$$

In the equation, $K_0(\cdot)$ is the weight function, λ is the bandwidth, n is the sample size, and x_i is the ith observation. You can use the C= suboption to specify the bandwidth and the WEIGHT= suboption to specify the weight function $K_0(\cdot)$.

Kernel Density Weight Functions

The formulas for the weight functions are as follows.

NORMAL

$$K_0(t) = \frac{1}{\sqrt{2\,\pi}}\exp\left(-\frac{1}{2}t^2\right) \text{ for } -\infty < t < \infty$$

QUADRATIC

$$K_0(t) = \frac{3}{4}\left(1-t^2\right) \text{ for } |t| \le 1$$

TRIANGULAR

$$K_0(t) = 1 - |t| \text{ for } -|t| \le 1$$

DOT Statement

Creates a dot plot that summarizes the values of a category variable.

Restriction: The DOT statement can be combined only with other horizontal categorization plot statements. When category plots are overlaid, all of the plots must use the same category variable. See "Plot Compatibility" on page 1780.

See: "Marker Fills and Outlines" on page 1805

Examples: "About Dot Plots" on page 60
"Example 7: Adding Statistical Limits to a Dot Plot" on page 1668

Syntax

DOT *category-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all dots from discrete category values.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
: specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
: specifies how to display grouped dots.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
: displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=DATA | LEFT | RIGHT
: specifies the location of the data label.

LEGENDLABEL="*text-string*"
: specifies the label that identifies the dot plot in the legend.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

STATLABEL

NOSTATLABEL
: specifies whether the response variable statistic is displayed in the axis and legend labels.

Limit options

LIMITATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
: specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITS=BOTH | LOWER | UPPER
: specifies which limit lines to display.

LIMITSTAT=CLM | STDDEV | STDERR
: specifies the statistic for the limit lines.

NOLIMITCAPS
: suppresses the serif cap on limit lines, if limit lines are displayed.

NUMSTD=*n*
: specifies the number of standard units for the limit lines when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR.

Marker options

FILLEDOUTLINEDMARKERS
: specifies that markers have a fill and an outline.

MARKERATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
: specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the marker outlines.

Plot options

ALPHA=*numeric-value*
: specifies the confidence level for the confidence limits.

CATEGORYORDER=RESPASC | RESPDESC
: specifies the order in which the categories are arranged.

FREQ=*numeric-variable*
: specifies a variable for the frequency count for each observation in the input data.

MISSING
: accepts a missing value as a valid category value.

RESPONSE=*response-variable*
: specifies a numeric response variable for the plot.

URL=*character-variable*
: specifies an HTML page to be displayed when parts of the plot are selected.

WEIGHT=*numeric-variable*
: specifies a variable that contains values to be used as weights for the calculations.

Plot reference options

NAME="*text-string*"
: assigns a name to a plot statement.

Statistics options

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
: specifies the statistic to use for computing the response colors.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
: specifies the statistic for the horizontal axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the dots.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default .05

Interactions This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA= value that you specify is used for all of the plots.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged. By default, the plot is sorted in ascending order based on the category values. You can change the order so that it is based on the response values.

Specify one of the following values:

RESPASC
sorts by the response values in ascending order.

RESPDESC
sorts by the response values in descending order.

Default The plot is sorted in ascending order based on the category values.

Restrictions This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Uniform scaling and response sorting cannot occur on the same axis. If the UNIFORM= option is used in the SGPLOT statement, the UNIFORM option is ignored for the sorted response axis and a note is generated in the log. The UNIFORM= option is applied to the other axes and groups if requested. Note that the UNIFORM= option can selectively apply scaling to only the X or Y axis

Interactions When a group variable is used with the CATEGORYORDER= option, the category order is not affected by the value of the groups. The categories are always sorted by the response statistic at a category level.

When this option and the GROUPORDER= option are both specified, the GROUPORDER= option is ignored.

Notes If CATEGORYORDER= is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

When two or more observations have the same response values, there is no guarantee that the tied values will be sub-sorted.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interaction This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorAltRamp style element

Requirement The COLORRESPONSE= option must also be specified, or COLORSTAT= must be FREQ or PCT. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

When COLORRESPONSE= is specified, the following values are valid:

SUM
sum values for the color response

MEAN
mean values for the color response

MEDIAN
median values for the color response

Default SUM when option COLORRESPONSE= is also specified.

Requirement The COLORRESPONSE= option must also be specified. If COLORRESPONSE= is not specified, COLORSTAT= is ignored.

Note This option is independent of the STAT= and RESPONSE= options.

DATALABEL
DATALABEL=*variable*
displays a label for each data point.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITS or LIMITSTAT, then the limit labels are also shown. Limit labels are shown when you use DATALABEL (without a variable).

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element. When DATALABELPOS=DATA is in effect, the marker color. Otherwise, the Color attribute of the GraphDataText style element.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=DATA | LEFT | RIGHT
specifies the location of the data label. Specify one of the following values:

DATA
places the label on the data primitives (at the right edge of the dots).

LEFT
places the label to the left of the dots on the left edge of the data area.

RIGHT
places the label to the right of the dots on the right edge of the data area.

Default DATA

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays limit information when limits are specified.

This option displays group values for each category when GROUP= is also specified.

This option displays response values for each overlaid chart.

When you specify a value of LEFT or RIGHT for this option, the splitting of data labels (using the SPLITCHAR= option) is not supported.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all marker symbols. If the marker symbol is not filled, then the data skin is applied to the outlines. Specify one of the following:

Table 10.9 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	**PRESSED**	**SHEEN**

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interaction You can use the MARKERATTRS= option to specify a filled marker symbol.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all dots from discrete category values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the category axis is discrete.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Tip Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interactions If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

When FREQ= and MISSING are both specified, the MISSING option has no effect.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped dots.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interaction This option is ignored unless GROUP= is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default ASCENDING

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the dot plot in the legend. By default, the label of the response variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, then the legend label is “Frequency”.

Interaction The LEGENDLABEL=option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

For grouped data, the LineStyle and LineThickness attributes attributes of the GraphError style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITS= or LIMITSTAT= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITS=BOTH | LOWER | UPPER
specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending horizontally from each dot. Upper limits extend to the right of the dot and lower limits extend to the left of the dot. By default, no limits are displayed unless you specify either the LIMITS= or LIMITSTAT= option. Specify one of the following values:

BOTH
adds lower and upper limit lines to the plot.

LOWER
adds lower limit lines to the plot.

UPPER
adds upper limit lines to the plot.

Default By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interaction Limit lines are displayed only when you specify STAT=MEAN.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines. Specify one of the following statistics:

CLM
confidence limits

STDDEV
standard deviation

STDERR
standard error

Default If you specify the LIMITS= option, then the default for LIMITSTAT= is CLM. Otherwise, there is no default.

Interactions If you specify the LIMITSTAT= option only, then the default value for the LIMITS= option is BOTH.

Limits lines are displayed only when you specify STAT=MEAN.

Tip To display limit labels, specify DATALABEL (without a variable).

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interaction Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOLIMITCAPS
suppresses the serif cap on limit lines, if limit lines are displayed.

Interaction The appearance of the limit lines is controlled by the LIMITATTRS= option.

NUMSTD=*n*
specifies the number of standard units for the limit lines when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR. You can specify any positive number, including decimals.

Default 1

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable for each category value are displayed on the horizontal axis.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the horizontal axis. Specify one of the following:

FREQ
the frequencies of the response or category variable.

The frequencies are calculated as follows:

- If you specify the RESPONSE= option, FREQ calculates the frequency of the response variable.
- If you do not specify the RESPONSE= option, FREQ calculates the frequency of the category variable.

MEAN
the mean of the response variable.

Interaction For STAT=MEAN to take effect, you must also specify the RESPONSE= option.

MEDIAN
the median of the response variable.

Interaction For STAT=MEDIAN to take effect, you must also specify the RESPONSE= option.

PERCENT
the percentage of the response or category variable.

The percentage is calculated as follows:

- If you specify the RESPONSE= option, PERCENT calculates the percentage of the sum of the response variable.
- If you do not specify the RESPONSE= option, PERCENT calculates the percentage of the frequency of the category variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the =PCTLEVEL= option in the PROC SGPLOT statement.

You can use the PCTNDEC= option in the SGPLOT procedure statement to control the number of decimals to be used when calculating the percent values.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level are zero.

SUM
the sum of the response variable.

This is the default value when you specify the RESPONSE= option.

Interaction For this value to take effect, you must also specify the RESPONSE= option.

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Restriction If you do not also specify the RESPONSE= option, then only the FREQ or PERCENT statistic is calculated (FREQ is the default). If you specify RESPONSE=, then you can use any of the statistics.

Interaction When the graph is generated, the statistic is appended to the variable name in the axis label and the legend (if it is created). However, if a label has been assigned to the variable, then the label appears in the axis label and legend instead of the statistic.

STATLABEL | NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels. STATLABEL forces the statistic to be displayed. NOSTATLABEL removes the statistic from the axis and legend labels.

Normally, the procedure displays the statistic along with the name of the response variable. However, when a custom label is assigned to the response variable, the procedure does not display the statistic. In each case, you can control whether the statistic is displayed.

Defaults The statistic is displayed for the response variable.

When a custom label is assigned to the response variable, the statistic is not displayed.

Interactions This option has no effect unless the RESPONSE= option is specified.

This option has no effect if you specify the axis label using the LABEL= option in an AXIS statement.

TIP=(*variable-list*) | NONE

specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)

a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE

suppresses the data tips from this plot.

Requirement: You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction: This option replaces all of the information that is displayed by default.

Tip: Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example: tip=(age weight)

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default: The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement: A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction: This option has no effect unless TIP= is also specified.

Note: A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip: Use the TIPLABEL= option to assign labels to the list of variables.

See: *SAS Formats and Informats: Reference*

Example: tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see “Using the ODS GRAPHICS Statement” on page 1886.

If your categorical chart is overlaid with other categorical charts, each chart can specify its own URL variable. The URL variable

specified for each chart overlay is honored. This capability affects the following charts: DOT, HBAR, HLINE, VBAR, VLINE.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction If your plot is overlaid with other categorization plots that also specify WEIGHT=, then the first WEIGHT variable that you specified is used for all of the plots.

DROPLINE Statement

Creates one or more drop lines from data point(s) to one or both axes. The line(s) can be horizontal, vertical, or both.

Restriction: When the DROPLINE variable is the same as the response variable of a categorical chart that is specified in the procedure, the DROPLINE statement is ignored.

Requirement: The DROPLINE statement must be overlaid with another plot that is derived from data values that provide boundaries for the axis area.

See: "About Drop Lines" on page 26

"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Syntax

DROPLINE X=*variable* | *x-axis-value*
Y=*variable* | *y-axis-value* <*/options*>;

Summary of Optional Arguments

Appearance options

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all drop lines from discrete X or Y values.

LINEATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the drop line.

NOCLIP
extends the plot axes to contain the drop lines.

TRANSPARENCY=*value*
specifies the degree of transparency for the drop line.

Axis options

DROPTO=BOTH | X | Y
specifies the axis to which the line is dropped.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Label options

LABEL=*variable* | "*text-string*"
creates labels for each drop line.

LABELATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the labels.

LEGENDLABEL="*text-string*"
specifies a label that identifies the plot in the legend.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

X=*variable* | *x-axis-value*
specifies the X coordinate of the drop line(s). If you specify an *x-axis-value* that is a text string, enclose the string in quotation marks.

Requirements Values must agree in type with the X-axis data type. For example, you should use numeric SAS date or time values (or SAS date/time constants) for a time axis.

Unformatted numeric values do not map to a formatted discrete axis. If a drop line is drawn to a point on a formatted discrete X axis, the DROPLINE X= value must be a formatted value that appears on the X axis.

Y=*variable* | *y-axis-value*
specifies the Y coordinate of the drop line(s). If you specify a *y-axis-value* that is a text string, enclose the string in quotation marks.

Requirements Values must agree in type with the Y-axis data type.

Unformatted numeric values do not map to a formatted discrete axis. If a drop line is drawn to a point on a formatted discrete Y

axis, the DROPLINE Y= value must be a formatted value that appears on the Y axis.

Optional Arguments

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 10.10 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all drop lines from discrete X or Y values.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset) where 0.5 represents half the distance between discrete ticks.

Requirement This option is applicable only when the X or Y axis is discrete.

DROPTO=BOTH | X | Y
specifies the axis to which the line is dropped.

BOTH
draws dropline(s) to both axes.

X
draws dropline(s) to the X axis, or to the X2 axis if specified.

Y
draws dropline(s) to the Y axis, or to the Y2 axis if specified.

Default X

LABEL=*variable* | "*text-string*"
creates labels for each drop line.

variable
a variable for the label value.

Requirement In order to use a variable to label your drop lines, you must specify a variable in X=, Y=, or both in the DROPLINE statement. If you specify scalar values in X= and Y=, the specified label variable is ignored and a message is written to the log.

"text-string"
a text string for the label value.

Requirement In order to use a text string to label your drop line, you must specify scalar values in X= and Y=. If you specify a variable in X=, Y=, or both in the DROPLINE statement, the specified text string is ignored and a message is written to the log.

Default No label is displayed.

Note Space is reserved at the maximum end of the X axis to accommodate the length of the labels regardless of where the labels appear in the plot.

Tip You can use the OFFSETMAX= axis option to adjust the amount of space that is reserved on the X axis for the labels.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the LABEL option is also specified.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LEGENDLABEL="*text-string*"
specifies a label that identifies the plot in the legend. By default, the label “drop” is used.

Interaction This option has no effect unless you also specify the NAME= option.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the drop line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see “Line Attributes and Patterns” on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphReference style element in the current style.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOCLIP
extends the plot axes to contain the drop lines.

By default, if a drop line extends beyond the data range, the drop line is not drawn. When NOCLIP is in effect, the axis ranges are extended, if necessary, to accommodate the longest drop lines.

TRANSPARENCY=*value*
specifies the degree of transparency for the drop line.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Note This option does not affect the drop-line label.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Details

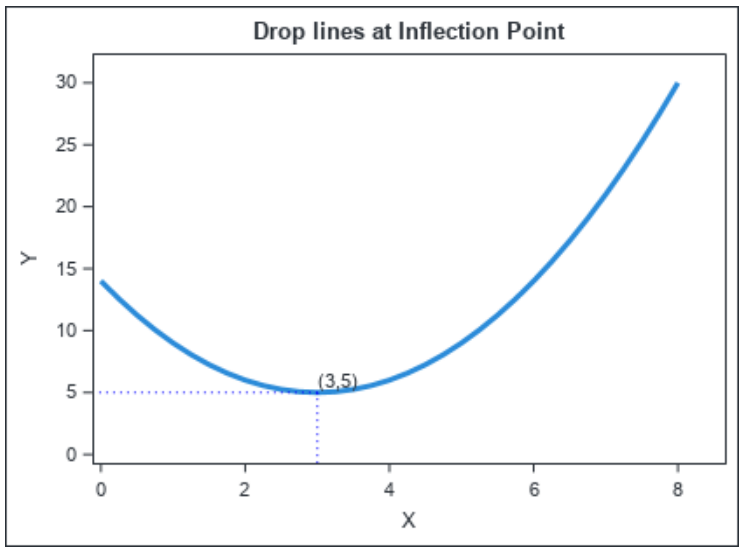
A drop line is always drawn perpendicular from the specified point to the X or Y axis. Axis offsets do not apply to drop lines, so they always meet the axis line.

The DROPTO= option controls whether a horizontal or vertical drop line is created. DROPTO=X specifies the horizontal axis for a vertical drop line, and DROPTO=Y specifies the vertical axis for a horizontal drop line. DROPTO=BOTH specifies both axes.

A DROPLINE statement must be used with another plot statement that establishes the axis scale for the DROPTO= points. For example, a drop line can be used with a scatter plot or a histogram. You can generate multiple drop lines by specifying a column for X and Y. The column type (numeric or string) must agree with the type of data presented on the axis.

Example

This example shows a DROPLINE statement originating from the point (X=3, Y=5) and dropping to both axes.




```
/* Create data set with X and Y variables
   that plot a parabola.*/
data test;
  do X=0 to 8 by 0.25;
    Y=(x-3)*(x-3) + 5;
      output;
  end;
run;

/* Create the plot and drop lines.
   Specify a label and line attributes for the drop lines.*/
title "Drop lines at Inflection Point";
proc sgplot data=test;
  series x=x y=y;
  dropline x=3 y=5 / dropto=both label="(3,5)"
      lineattrs=(color=blue pattern=dot);
  yaxis min=0;
run;
title;
```

ELLIPSE Statement

Adds a confidence or prediction ellipse to another plot.

Restriction: In order to produce useful output, the ELLIPSE statement should be used with another plot statement that uses numeric axes. See "Plot Compatibility" on page 1780.

See: "About Ellipse Plots" on page 42

Syntax

ELLIPSE X=*numeric-variable* Y=*numeric-variable* </option(s)>;

Summary of Optional Arguments

Appearance options

FILL

NOFILL
: specifies whether the area fill is visible.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

LEGENDLABEL="*text-string*"
: specifies a label that identifies the ellipse in the legend.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the outlines for the ellipse.

OUTLINE

NOOUTLINE
: specifies whether the outlines of the ellipse are visible.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
: assigns the Y variable to the secondary (right) vertical axis.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Plot options

ALPHA=*numeric-value*
specifies the confidence level for the ellipse.

CLIP
specifies that the data for the ellipse is ignored when determining the data ranges for the axes.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

TYPE=MEAN | PREDICTED
specifies the type of ellipse.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

X=*numeric-variable*
specifies a numeric variable for the X axis.

Y=*numeric-variable*
specifies a numeric variable for the Y axis.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the ellipse.

Default .05

Range 0.01 (complete confidence) to 0.99 (no confidence)

CLIP
specifies that the data for the ellipse is ignored when determining the data ranges for the axes. By default, the data for the ellipse is considered when determining the data ranges for the axes.

Interaction This option is automatically set if you specify UNIFORM=SCALE or UNIFORM=ALL option in the PROC SGPLOT statement.

Note This option is not available in the SGPANEL procedure. Clipping is always true for SGPANEL.

FILL | NOFILL
specifies whether the area fill is visible. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default: The default status of the area fill is specified by the DisplayOpts attribute of the GraphEllipse style element in the current style.

Interactions: Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults: For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interaction: This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction: If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips: Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

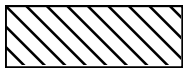
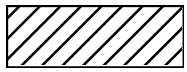
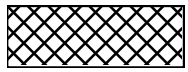
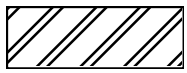
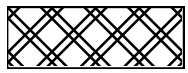
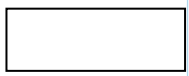
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.11 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FREQ=*numeric-variable*

specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

GROUP=*variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

You can use the NOMISSINGGROUP option to omit missing values of the group variable from the plot.

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the ellipses, but the PATTERN= suboption of the OUTLINEATTRS= option could be used to assign the same line pattern to all ellipse outlines, letting outline color distinguish group values.

LEGENDLABEL="*text-string*"

specifies a label that identifies the ellipse in the legend. By default, the label describes the confidence value of the ellipse and the type of ellipse. For example, "95% Prediction Ellipse."

LINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the outlines for the ellipse. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OUTLINE | NOOUTLINE
specifies whether the outlines of the ellipse are visible. The OUTLINE option shows the outlines and hides the fill. The NOOUTLINE option hides the outlines.

Default The default status of the outlines is specified by the DisplayOpts attribute of the GraphEllipse style element in the current style.

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

TYPE=MEAN | PREDICTED
specifies the type of ellipse. MEAN specifies a confidence ellipse for the population mean. PREDICTED specifies a prediction ellipse for a new observation. Both ellipse types assume a bivariate normal distribution.

Default PREDICTED

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Details

Statement Description

The ELLIPSE statement plots an ellipse with specified semimajor and semiminor axis lengths, a specified slope for the major axis, and the ellipse center. The ellipse center is the point of intersection of the semimajor and semiminor axes. The ELLIPSE statement computes an ellipse for a set of points that is specified by the X= on page 980 and Y= on page 980 columns and a confidence level that is specified by the ALPHA= option.

You can generate one or more ellipses by specifying a numeric column for any or all of options SEMIMAJOR=, SEMIMINOR=, XORIGIN=, and YORIGIN=. If any of the specified columns contain a missing value, then no ellipse is drawn. To request a vertical major axis, specify SLOPE=. (missing value) as a column value.

Confidence and Prediction Ellipses

Two types of ellipses can be computed for the input data (where observations correspond to points in a scatter plot). One is a confidence ellipse for the population mean (TYPE=MEAN), and the other is a prediction ellipse for a new observation (TYPE=PREDICT). Both assume a bivariate normal distribution.

Let $\overline{Z}$ and S be the sample mean and sample covariance matrix of a random sample of size n from a bivariate normal distribution with mean μ and covariance matrix Σ. The variable $\overline{Z} - \mu$ is distributed as a bivariate normal variate with mean zero and covariance $(\frac{1}{n})\Sigma$, and it is independent of S. Using Hotelling's T^2 statistic, which is defined as follows:

$$T^2 = n(\overline{Z} - \mu)' S^{-1} (\overline{Z} - \mu)$$

A $100(1-\alpha)\,\%$ confidence ellipse for μ is computed from the following equation:

$$\frac{n}{n-1}(\overline{Z} - \mu)' S^{-1} (\overline{Z} - \mu) = \frac{2}{n-2} F_{2,n-2}(1-\alpha)$$

where $F_{2,n-2}(1-\alpha)$ is the $(1-\alpha)$ critical value of an F distribution with degrees of freedom 2 and $n-2$.

A prediction ellipse is a region for predicting a new observation in the population. It also approximates a region containing a specified percentage of the population.

Denote a new observation as the bivariate random variable Z_{new}. The following variable:

$$Z_{new} - \overline{Z} = (Z_{new} - \mu) - (\overline{Z} - \mu)$$

is distributed as a bivariate normal variate with mean zero (the zero vector) and covariance $(1 + \frac{1}{n})\Sigma$, and it is independent of S. A $100(1-\alpha)\,\%$ prediction ellipse is then given by the following equation:

$$\frac{n}{n-1}(\overline{Z} - \mu)' S^{-1} (\overline{Z} - \mu) = \frac{2(n+1)}{n-2} F_{2,n-2}(1-\alpha)$$

The family of ellipses generated by different critical values of the F distribution has a common center (the sample mean) and common major and minor axis directions.

The shape of an ellipse depends on the aspect ratio of the plot. The ellipse indicates the correlation between the two variables if the variables are standardized (by dividing the variables by their respective standard deviations). In this situation, the ratio between the major and minor axis lengths is:

$$\sqrt{\frac{1+|r|}{1-|r|}}$$

In particular, if $r = 0$, then the ratio is 1, which corresponds to a circular confidence contour and indicates that the variables are uncorrelated. A larger value of the ratio indicates a larger positive or negative correlation between the variables.

ELLIPSE Default Data Tip

An ellipse generated by the ELLIPSE statement can display a default data tip in HTML output. Data tips are not enabled by default. To enable data tips, specify the IMAGEMAP option in an ODS GRAPHICS statement. The data tip displays the following roles:

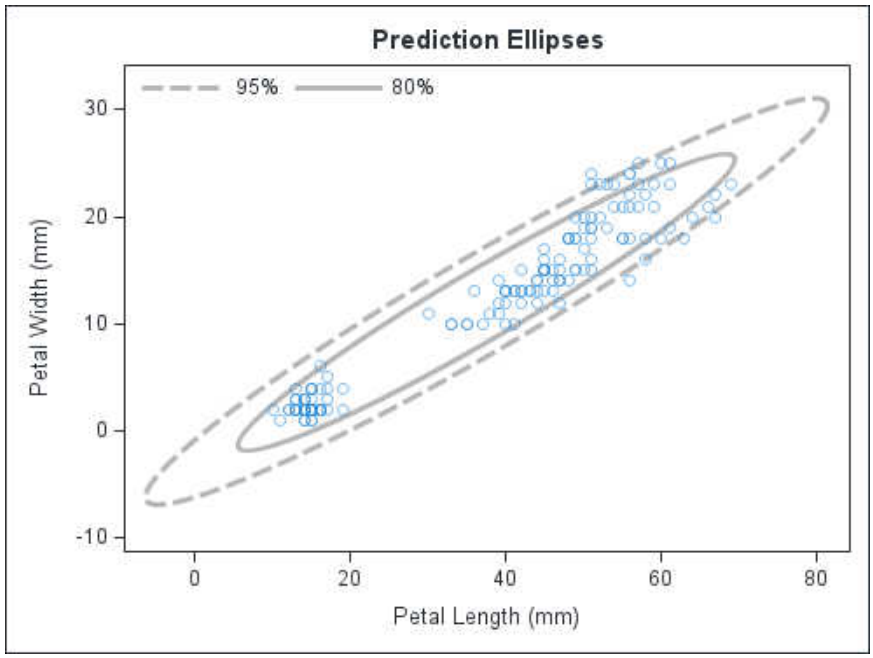
- GROUP= (if specified)
- XORIGIN=
- YORIGIN=
- SEMIMAJOR=
- SEMIMINOR=
- SLOPE=

The ELLIPSE statement does not support data tip options that can be used to customize the data tip.

Examples

Example 1: Prediction Ellipses

The following graph was generated by the "Example Program" in *SAS Graph Template Language: Reference*:



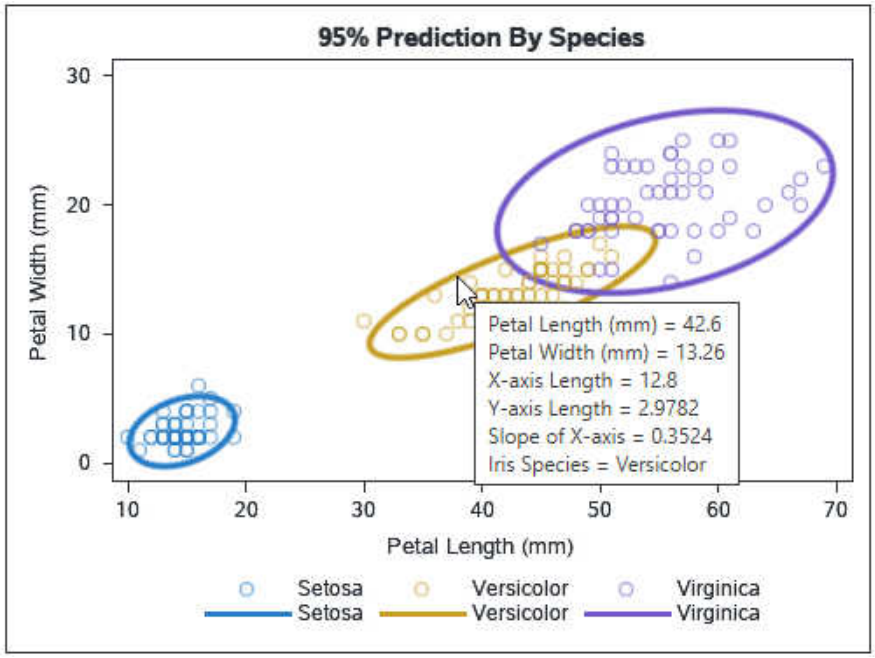
Example Program

This example overlays two ELLIPSE statements on a scatter plot of the same data. Both ELLIPSE statements use TYPE= PREDICTED. One ELLIPSE statement uses ALPHA= 0.05, and the other uses ALPHA=0.2.

```
title "Prediction Ellipses";
proc sgplot data=sashelp.iris;
   ellipse x=petallength y=petalwidth /
      type=predicted  alpha=0.05
      name="p95" legendlabel="95%"
      lineattrs=graphconfidence2;
   ellipse x=petallength y=petalwidth /
      type=predicted  alpha=0.2
      name="p80" legendlabel="80%"
      lineattrs=graphconfidence;
   scatter x=petallength y=petalwidth /
      transparency=0.5
      markerattrs=(symbol=circle size=6pt);
   keylegend "p95" "p80" /
      location=inside position=topleft;
run;
title;
```

Example 2: 95% Confidence Ellipses by Plant Species

This example creates an ellipse plot that is grouped by plant species from the data in Sashelp.Iris with data tips enabled. The following image shows the output for "Example Program" in *SAS Graph Template Language: Reference*. It also shows the data tip for the Versicolor ellipse.



Example Program

```
ods graphics / imagemap;                              /* 1 */

title "95% Prediction By Species";
proc sgplot data=sashelp.iris;
   ellipse x=petallength y=petalwidth /
      group=species                                   /* 2 */
      type=predicted alpha=0.05 name="p95";
   scatter x=petallength y=petalwidth /
      name="sp"
      group=species                                   /* 3 */
      markerattrs=(symbol=circle size=6pt);
   keylegend "sp" "p95" / across=3;
run;

ods graphics / reset=imagemap;
title;
```

1 The IMAGEMAP option generates an image map in the HTML output, which enables the data tips. See "ELLIPSE Default Data Tip" on page 986.

2 The GROUP= option groups the ellipses by column Species.

3 The GROUP= option groups the scatter plot by column Species.

ELLIPSEPARM Statement

Adds an ellipse to the primary plot, specified by axis, slope, and origin parameters.

Restriction: The ELLIPSEPARM statement must be overlaid with another plot that is derived from data values that provide boundaries for the axis area. It is typically overlaid with a scatter plot. See "Plot Compatibility" on page 1780.

Requirement: You must specify the axis parameters.

See: "About Ellipse Plots" on page 42

Syntax

ELLIPSEPARM SEMIMAJOR=*numeric-variable* | *positive-number*
SEMIMINOR=*numeric-variable* | *positive-number* </option(s)>;

Summary of Optional Arguments

Appearance options

FILL

NOFILL
: specifies whether to display a filled ellipse.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

LEGENDLABEL="*text-string*"
: specifies a label that identifies the ellipse in the legend.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the outlines for the ellipse.

OUTLINE

NOOUTLINE
: specifies whether the outlines of the ellipse are visible.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
: assigns the Y variable to the secondary (right) vertical axis.

Group options

GROUP=*variable*
: specifies a variable that is used to group the data.

NOMISSINGGROUP

specifies that missing values of the group variable are not included in the plot.

Plot options

CLIP
: specifies that the data for the ellipse is ignored when determining the data ranges for the axes.

SLOPE=*numeric-value* | *numeric-variable*
: specifies the slope of the major axis for the ellipse.

XORIGIN=*numeric-value* | *numeric-variable*
: specifies the X coordinate of the center of the ellipse.

YORIGIN=*numeric-value* | *numeric-variable*
: specifies the Y coordinate of the center of the ellipse.

Plot reference options

NAME="*text-string*"
: assigns a name to a plot statement.

Required Arguments

SEMIMAJOR=*numeric-variable* | *positive-number*
specifies half the length of the major axis for the ellipse. Values are in the same units as the data. The SEMIMAJOR value can be greater than, smaller than, or equal to the SEMIMINOR value.

Restrictions
: The semimajor value must be positive. If a numeric variable is specified and the variable contains negative values, then no ellipses are drawn.

  If the variable contains a missing value, the ellipse for that observation is not drawn. However, the remaining ellipses that have nonmissing values for this argument and for SEMIMINOR= are drawn.

SEMIMINOR=*numeric-variable* | *positive-number*
specifies half the length of the minor axis for the ellipse. Values are in the same units as the data. The SEMIMINOR value can be greater than, smaller than, or equal to the SEMIMAJOR value.

Restrictions
: The semiminor value must be positive. If a numeric variable is specified and the variable contains negative values, then no ellipses are drawn.

  If the variable contains a missing value, the ellipse for that observation is not drawn. However, the remaining ellipses that have nonmissing values for this argument and for SEMIMAJOR= are drawn.

Optional Arguments

CLIP
specifies that the data for the ellipse is ignored when determining the data ranges for the axes. By default, the data for the ellipse is considered when determining the data ranges for the axes.

Interaction This option is automatically set if you specify UNIFORM=SCALE or UNIFORM=ALL option in the PROC SGPLOT statement.

FILL | NOFILL
specifies whether to display a filled ellipse. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default The default is specified by the DisplayOpts attribute of the GraphEllipse style element in the current style.

Interactions Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interaction This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips

Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)

specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element

specifies the name of a style element.

options

specifies fill-pattern style options.

Specify one or both of the following options:

COLOR=*color*

specifies a color to use for the fill lines.

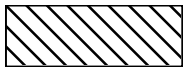
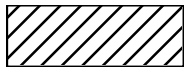
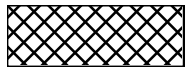
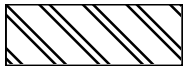
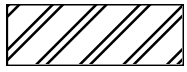
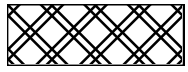
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

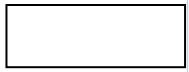
PATTERN=*line-pattern*

specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.12 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction The group value can be a character or numeric column. The GROUP= option can be used only when a numeric column or value is specified for the SEMIMAJOR= and SEMIMINOR= arguments, and when used, the SLOPE=, XORGIN= and YORIGIN= options.

Note If a constant value is specified for SEMIMAJOR=, SEMIMINOR= XORIGIN=, YORIGIN=, and SLOPE=, and GROUP= is also specified, an ellipse is drawn for each group value, but all of the ellipses overlap.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

You can use the NOMISSINGGROUP option to omit missing values of the group variable from the plot.

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the ellipses, but the PATTERN= suboption of the OUTLINEATTRS= option could be used to assign the same line pattern to all ellipse outlines, letting outline color distinguish group values.

LEGENDLABEL="*text-string*"
specifies a label that identifies the ellipse in the legend.

Default EllipseParm

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the outlines for the ellipse. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OUTLINE | NOOUTLINE
specifies whether the outlines of the ellipse are visible. The OUTLINE option shows the outlines and hides the fill. The NOOUTLINE option hides the outlines.

Default The default status of the outlines is specified by the DisplayOpts attribute of the GraphEllipse style element in the current style.

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

SLOPE=*numeric-value* | *numeric-variable*
specifies the slope of the major axis for the ellipse. The slope can be positive or negative.

Default 0

Note The slope value is in the data space and might not always be maintained in the screen space. Thus, setting SLOPE=1 does not always generate a 45-degree line on the screen.

Tips SLOPE=0 creates a major axis parallel to the X axis.

SLOPE= . (missing value) creates a major axis parallel to the Y axis.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

XORIGIN=*numeric-value* | *numeric-variable*
specifies the X coordinate of the center of the ellipse. Values are in the units of the data.

Default 0

Tips By default, if the value specified for the XORIGIN= option is outside of the X-axis data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP option.

You can use the YORIGIN= option to specify the Y coordinate of the center of the ellipse.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

YORIGIN=*numeric-value* | *numeric-variable*
specifies the Y coordinate of the center of the ellipse. Values are in the units of the data.

Default 0

Tips By default, if the value specified for the YORIGIN= option is outside of the Y-axis data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP option.

You can use the XORIGIN= option to specify the X coordinate of the center of the ellipse.

Details

Statement Description

The ELLIPSEPARM statement plots an ellipse with specified semimajor and semiminor axis lengths, a specified slope for the major axis, and the ellipse center. The ellipse center is the point of intersection of the semimajor and semiminor axes. The ELLIPSEPARM statement does not perform computations on input data to derive the location and shape of the ellipse. Another plot that is derived from data values that provide boundaries for the axis area must be included in the layout. The ellipse is typically overlaid with a scatter plot. See ELLIPSE Statement for information about computed ellipses.

You can generate a single ellipse by specifying a constant for each of the following arguments: SEMIMAJOR=, SEMIMINOR=, SLOPE=, XORIGIN=, YORIGIN=. The SLOPE=, XORIGIN=, and YORIGIN= arguments are set to a constant by default.

You can generate multiple ellipses by specifying a numeric column for any or all of those arguments. If any of the SEMIMAJOR=, SEMIMINOR=, XORIGIN=, or YORIGIN= constants or columns contains a missing value, then no ellipse is drawn. To request a vertical major axis, specify SLOPE=. (missing value) as a constant or column value.

ELLIPSEPARM Default Data Tip

An ellipse generated by the ELLIPSEPARM statement can display a default data tip in HTML output. Data tips are not enabled by default. To enable data tips, specify the IMAGEMAP option in an ODS GRAPHICS statement. The data tip displays the following roles:

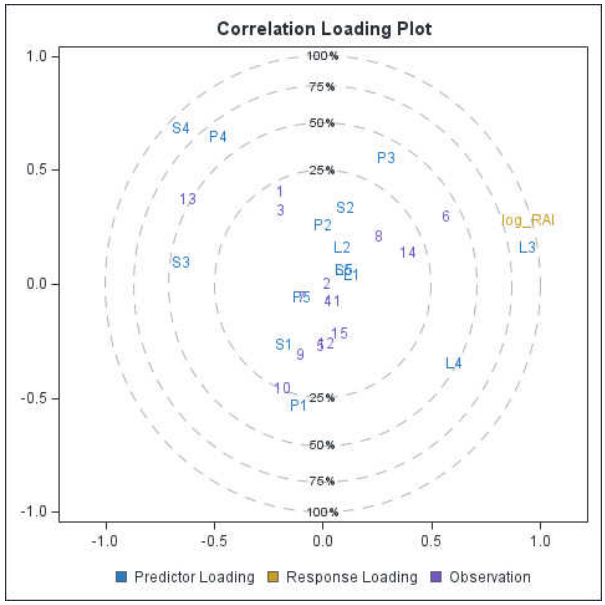
- GROUP= (if specified)
- XORIGIN=
- YORIGIN=
- SEMIMAJOR=
- SEMIMINOR=
- SLOPE=

The ELLIPSEPARM statement does not support data tip options that can be used to customize the data tip.

Example: Simple Correlation Loading Plot

This example resembles a correlation loading plot that is produced using the PLS procedure in the SAS/STAT product. The example consists of overlaid scatter plots of the scores of the first two factors, the loadings of the model effects, and the loadings of the dependent variables. The loadings are scaled so that the amount of variation in the variables that is explained by the model is proportional to the distance from the origin; circles indicating various levels of explained variation are also overlaid.

The circles are drawn with ELLIPSEPARM statements by specifying the same value for SEMIMAJOR= on page 990 and SEMIMINOR= on page 990. All circles are concentric because they share the same origin (0,0). The input data shown is representative of that computed by the PLS procedure for the Correlation Loadings Plot. For more details, see PLS procedure example Examining Model Details in *SAS/STAT User's Guide*.




```
data corrplot;
infile cards missover dlm="#";
length CorrGroup $ 18 CorrLabel CorrLabel CircleLabel $ 8;
input Corr1 Corr2 CorrGroup CorrLabel xCircleLabel
yCircleLabel CircleLabel;
datalines;
-0.179 # -0.268 # Predictor Loading # S1 # 0 # 0.5    # 25%
0.105  # 0.332  # Predictor Loading # S2 # 0 # -0.5   # 25%
-0.654 # 0.094  # Predictor Loading # S3 # 0 # 0.707  # 50%
-0.653 # 0.685  # Predictor Loading # S4 # 0 # -0.707 # 50%
0.096  # 0.059  # Predictor Loading # S5 # 0 # 0.866  # 75%
0.132  # 0.036  # Predictor Loading # L1 # 0 # -0.866 # 75%
0.087  # 0.156  # Predictor Loading # L2 # 0 # 1      # 100%
0.940  # 0.160  # Predictor Loading # L3 # 0 # -1     # 100%
0.607  # -0.350 # Predictor Loading # L4
0.096  # 0.059  # Predictor Loading # L5
-0.111 # -0.534 # Predictor Loading # P1
0.003  # 0.256  # Predictor Loading # P2
0.293  # 0.551  # Predictor Loading # P3
-0.480 # 0.643  # Predictor Loading # P4
-0.096 # -0.059 # Predictor Loading # P5
0.946  # 0.279  # Response Loading  # log_RAI
-0.196 # 0.403  # Observation       # 1
0.020  # -0.001 # Observation       # 2
-0.195 # 0.324  # Observation       # 3
0.021  # -0.079 # Observation       # 4
-0.009 # -0.274 # Observation       # 5
0.567  # 0.294  # Observation       # 6
-0.096 # -0.059 # Observation       # 7
0.258  # 0.210  # Observation       # 8
```

```
-0.104 # -0.309 # Observation       # 9
-0.187 # -0.458 # Observation       # 10
0.051  # -0.078 # Observation       # 11
0.017  # -0.260 # Observation       # 12
-0.621 # 0.372  # Observation       # 13
0.392  # 0.138  # Observation       # 14
0.080  # -0.221 # Observation       # 15
;
run;

ods graphics / width=480px height=480px;

title "Correlation Loading Plot";
proc sgplot data=corrplot;
  ellipseparm semimajor=1 semiminor=1 /
     slope=0 xorigin=0 yorigin=0 clip
     lineattrs=(pattern=dash color=black thickness=1px)
     transparency=0.7;
  ellipseparm semimajor=0.87 semiminor=0.87 /
     slope=0 xorigin=0 yorigin=0 clip
     lineattrs=(pattern=dash color=black thickness=1px)
     transparency=0.7;
  ellipseparm semimajor=0.71 semiminor=0.71 /
     slope=0 xorigin=0 yorigin=0 clip
     lineattrs=(pattern=dash color=black thickness=1px)
     transparency=0.7;
  ellipseparm semimajor=0.5 semiminor=0.5 /
     slope=0 xorigin=0 yorigin=0 clip
     lineattrs=(pattern=dash color=black thickness=1px)
     transparency=0.7;
  scatter x=xcirclelabel y=ycirclelabel /
     markerchar=circlelabel;
  scatter x=corr1 y=corr2 / name="ScatterVars"
     group=corrgroup markerchar=corrlabel
     markercharattrs=(size=9pt);
  xaxis values=(-1.0 to 1.0 by 0.5) display=(nolabel);
  yaxis values=(-1.0 to 1.0 by 0.5) display=(nolabel);
  keylegend "ScatterVars";
run;

ods graphics / reset=width reset=height;
title;
```

FRINGE Statement

Creates a fringe plot on the X axis of an X-Y plot.

Interaction: Fringe plots can be overlaid with all plots except with box plots and categorical charts (bar charts, line plots, and dot plots). See “Plot Compatibility” on page 1780.

Example: “About Fringe Plots” on page 20

Syntax

FRINGE *numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the fringe lines.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
: specifies a variable that is used to group the data.

HEIGHT=*dimension*<**units**>
: specifies the height of the fringe lines.

NOMISSINGGROUP
: specifies that missing values of the group variable are not included in the plot.

Label options

LEGENDLABEL="*text-string*"
: specifies a label that identifies the needle plot in the legend.

Plot reference options

NAME="*text-string*"
: assigns a name to a plot statement.

Required Argument

numeric-variable
specifies the variable that provides the X coordinates of the data values.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes. Each distinct group value is represented in the graph by a different line color. Line patterns are not changed across groups.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

You can use the NOMISSINGGROUP option to omit missing values of the group variable from the plot.

HEIGHT=*dimension*<units>
specifies the height of the fringe lines. You can also specify the unit of measurement. The default unit is pixels.

The following table contains the units that are available:

Table 10.13 *Measurement Units*

Unit	Description
CM	centimeters
IN	inches
MM	millimeters
PCT or %	percentage

Unit	Description
PT	point size, calculated at 72 dots per inch
PX	pixels

Default 10px

LEGENDLABEL="*text-string*"
specifies a label that identifies the needle plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fringe lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

GRADLEGEND Statement

Generates a gradient legend that maps the data range of a response variable to a range of colors.

Notes: You can use up to four GRADLEGEND statements in a procedure.

The GRADLEGEND statement is used in conjunction with the COLORRESPONSE= option. (The COLORRESPONSE= option is specified in the plot statement and indicates the response variable that is used to map the colors.)

See: "Using Gradient Color Legends" on page 1793

Syntax

GRADLEGEND <"*name*"> <*/options*>;

Summary of Optional Arguments

Appearance options

BORDER
NOBORDER

specifies whether a border is visible around the legend.

INTEGER
specifies that integers are used for the gradient legend.

NOTITLE
removes the default title. The default title is the name of the response variable.

OUTERPAD=*numeric-value<units>* | (*pad-options*)
specifies the amount of extra space that is added outside the legend border.

POSITION=TOP | BOTTOM | LEFT | RIGHT
specifies the position of the legend within the graph.

TITLE="*text-string*"
specifies a title for the legend.

TITLEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the title.

Legend options

"name"
specifies the name of the plot that you want to include in the legend.

Scale options

EXTRACTSCALE

EXTRACTSCALE=DEFAULT | SCIENTIFIC
extracts a scale factor from the tick values and uses it to reduce the tick value width.

Optional Arguments

"name"
specifies the name of the plot that you want to include in the legend. The name that you specify must correspond to a value that you entered for the NAME= option in a plot statement. The plot statement must also specify the COLORRESPONSE= option.

Default If no name is specified, the legend references whichever plot statement specifies the COLORRESPONSE= option. If the procedure contains more than one plot with a COLORRESPONSE= option and you do not specify a plot name, then the legend attempts to reference both or all of these plots. The resulting legend might be hard to read.

Restriction Only one name can be specified. If you want a continuous legend for more than one plot, you can use multiple GRADLEGEND statements. You can use up to four GRADLEGEND statements in a procedure.

BORDER | NOBORDER
specifies whether a border is visible around the legend.

Default NOBORDER

EXTRACTSCALE
EXTRACTSCALE=DEFAULT | SCIENTIFIC

extracts a scale factor from the tick values and uses it to reduce the tick value width. The scale used is appended to the legend title as shown in the following example.

```
Total Sales (millions)
```

For long legend titles, if the scale does not fit the available space, then the title is truncated, and the scale is appended to the truncated title. Ellipses indicate that the label was truncated as shown in the following example.

```
Total Sales for the Fourth Quarter Of ... (millions)
```

In extreme cases where the title does not fit even with truncation, the title is dropped.

You can also specify whether to use a named scale or a scientific-notation scale.

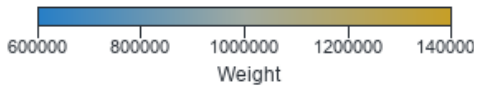
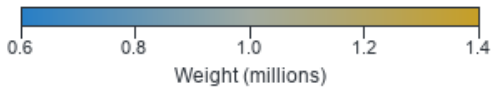
DEFAULT
extracts a named scale.

A named scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as 10^*n*) for values over 999 trillion. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of 1/10 (10^–*n*) for values less than 1 trillionth.

SCIENTIFIC
extracts a scientific-notation scale.

A scientific-notation scale is a multiple of 10 expressed as 10^*n* for values greater than 1, or a multiple of 1/10 expressed as 10^–*n* for values less than 1.

The following examples show a gradient legend before and after EXTRACTSCALE= is specified:

Default Scale	EXTRACTSCALE Specified
	

Default DEFAULT

Restriction The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale.

INTEGER
specifies that integers are used for the gradient legend.

NOTITLE
removes the default title. The default title is the name of the response variable.

Default The default title is displayed.

OUTERPAD=*numeric-value<units>* | (*pad-options*)
specifies the amount of extra space that is added outside the legend border.

numeric-value<units>
specifies a dimension to use for the extra space at the left, right, top, and bottom of the legend border.

The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

(*pad-options*)
a space-separated list of one or more of the following name-value pair options, enclosed in parentheses:

LEFT=*numeric-value<units>*	specifies the amount of extra space added to the left side.
RIGHT=*numeric-value<units>*	specifies the amount of extra space added to the right side.
TOP=*numeric-value<units>*	specifies the amount of extra space added to the top.
BOTTOM=*numeric-value<units>*	specifies the amount of extra space added to the bottom.

Note Sides that are not assigned padding are padded with the default amount.

Tip Use *pad-options* to create non-uniform padding.

Default No padding

Note The default units for *dimension* are pixels. If you want to specify values in other units, then you must specify the units with the value. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Example "Example: Gradient Legend That Uses the OUTERPAD= Option" on page 1007

POSITION=TOP | BOTTOM | LEFT | RIGHT
specifies the position of the legend within the graph.

Default RIGHT

Notes By default, if you use more than one GRADLEGEND statement, then each legend is placed in a different position.

If you specify more than one legend with the same position, then those legends are placed at that position.

TITLE="*text-string*"
specifies a title for the legend.

Default If you do not specify this option, then the name of the COLORRESPONSE= variable is displayed as the title.

TITLEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the title. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

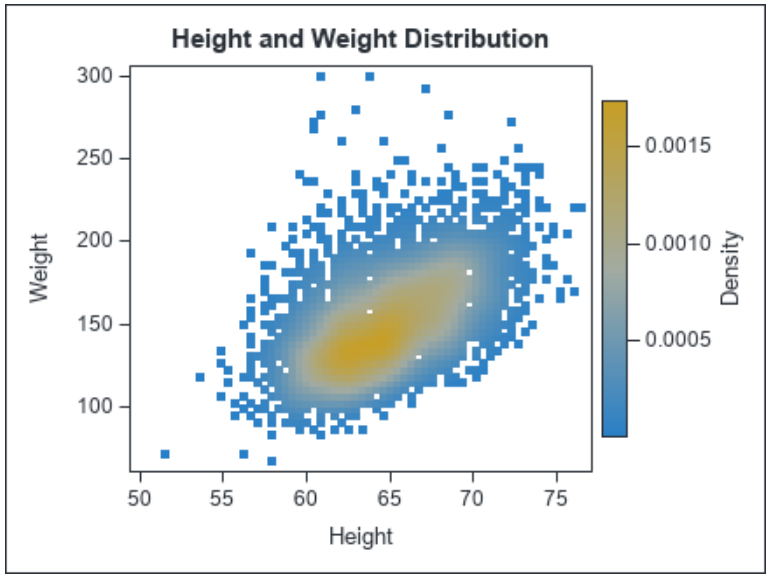
For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
TITLEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TITLEATTRS=GraphTitleText
```

Example: Gradient Legend That Uses the OUTERPAD= Option

The following example shows a gradient legend that maps a density scale to a color gradient. The OUTERPAD= on page 1174 option adds padding to the top and bottom of the gradient.




```
title "Height and Weight Distribution";
proc sgplot data=sashelp.gridded(where=(count>0));
  scatter x=height y=weight /
       colorresponse=density  name="scatter"
       markerattrs=(symbol=squarefilled size=6px);
```

```
  gradlegend "scatter" /
      outerpad=(top=20px bottom=20px);
run;
title;
```

HBAR Statement

Creates a horizontal bar chart that summarizes the values of a category variable.

Restrictions: When category plots are overlaid, all of the plots must use the same category variable.

The HBAR statement can be combined only with other categorization plot statements in the SGPLOT procedure. See "Plot Compatibility" on page 1780.

Tips: The HBAR statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

Bar charts can be combined with basic plot types using the HBARBASIC and VBARBASIC statements.

Example: "About Bar Charts" on page 53

Syntax

HBAR *category-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed.

FILL

NOFILL
specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
specifies the thickness of the bars when the category data is numeric.

NOZEROBARS
suppresses zero-length bars.

OUTLINE

NOOUTLINE
specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

X2AXIS
assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the category variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
: specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
: specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
: displays a label for each bar.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
: specifies the fit policy for the bar labels.

DATALABELPOS=DATA | LEFT | RIGHT
: specifies the location of the data label.

LEGENDLABEL="*text-string*"
: specifies the label that identifies the bar chart in the legend.

SEGLABEL
: displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
: specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*
: specifies the text format used to display the bar segment labels.

STATLABEL

NOSTATLABEL
: specifies whether the response variable statistic is displayed in the axis and legend labels.

Limit options

LIMITATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
: specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITS=BOTH | LOWER | UPPER
: specifies which limit lines to display.

LIMITSTAT=CLM | STDDEV | STDERR
: specifies the statistic for the limit lines.

NUMSTD=*n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR.

Plot options

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

MISSING
accepts a missing value as a valid category value.

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Statistics options

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the horizontal axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the bars. The variable generates the midpoints to which each observation in the data set contributes.

Interval bar charts are supported when the category axis is set to TYPE=LINEAR.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default .05

Interactions This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA= value that you specify is used for all of the plots.

ATTRID=*character-value*

specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BARWIDTH=*numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Defaults .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

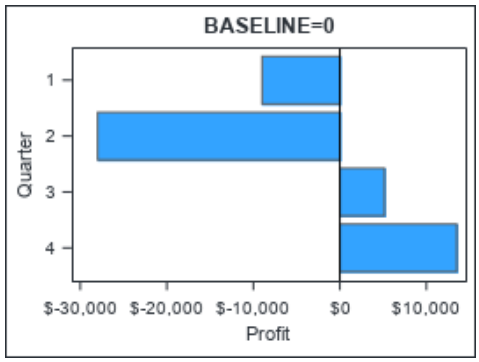
Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*

specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interactions If GROUPDISPLAY=STACK is specified, this option is ignored.

When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged. By default, the plot is sorted in ascending order based on the category values. You can change the order so that it is based on the response values.

Specify one of the following values:

RESPASC
sorts by the response values in ascending order.

RESPDESC
sorts by the response values in descending order.

Default The plot is sorted in ascending order based on the category values.

Restrictions This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Uniform scaling and response sorting cannot occur on the same axis. If the UNIFORM= option is used in the SGPLOT statement, the UNIFORM option is ignored for the sorted response axis and a note is generated in the log. The UNIFORM= option is applied to the other axes and groups if requested. Note that the UNIFORM= option can selectively apply scaling to only the X or Y axis

Interactions When a group variable is used with the CATEGORYORDER= option, the category order is not affected by the value of the groups. The categories are always sorted by the response statistic at a category level.

When this option and the GROUPORDER= option are both specified, the GROUPORDER= option is ignored.

Notes If CATEGORYORDER= is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

When two or more observations have the same response values, there is no guarantee that the tied values will be sub-sorted.

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified and when GROUPDISPLAY=CLUSTER.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Requirement The COLORRESPONSE= option must also be specified, or COLORSTAT= must be FREQ or PCT. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

When COLORRESPONSE= is specified, the following values are valid:

SUM
sum values for the color response

MEAN
mean values for the color response

MEDIAN
median values for the color response

Default SUM when option COLORRESPONSE= is also specified.

Requirement The COLORRESPONSE= option must also be specified. If COLORRESPONSE= is not specified, COLORSTAT= is ignored.

Note This option is independent of the STAT= and RESPONSE= options.

DATALABEL
DATALABEL=*variable*
displays a label for each bar.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITS or LIMITSTAT, then the limit labels are also shown. Limit labels are also shown when you use DATALABEL without a variable.

This option is supported with the GROUP= option only when GROUPDISPLAY=CLUSTER is specified. Otherwise, the label is not shown. The labels are shown if you also specify the DATALABELPOS= option.

By default, the data label fit policy is to show the labels unless they collide. As a result, the labels sometimes might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
specifies the fit policy for the bar labels. Specify one of the following values:

NONE
specifies that no fit policy is implemented for the bar labels.

By default, the fit policy is to show the labels unless they collide. As a result, the labels might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

INSIDEPREFERRED
places bar labels inside the bars, when possible.

Labels are placed outside the bars in the following cases:

- If a label cannot fit within the length of its respective bar, then that label is moved outside the bar.
- If the font height for any label exceeds the bar width, then all labels are positioned outside the bars.
- If SEGLABEL is specified, the INSIDEPREFERRED policy is ignored and the labels are positioned outside the bars.

Default Show the labels unless they collide.

Interaction This option has no effect unless DATALABEL= is also specified.

DATALABELPOS=DATA | LEFT | RIGHT
specifies the location of the data label. Specify one of the following values:

DATA
places the label on the data primitives (at the right edge of the bars).

LEFT
places the label to the left of the bars on the left edge of the data area.

RIGHT
places the label to the right of the bars on the right edge of the data area.

Default DATA

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays limit information when limits are specified.

This option displays group values for each category when GROUP= is also specified.

This option displays response values for each overlaid chart.

This option does not support the splitting or rotation of data labels.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.14 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

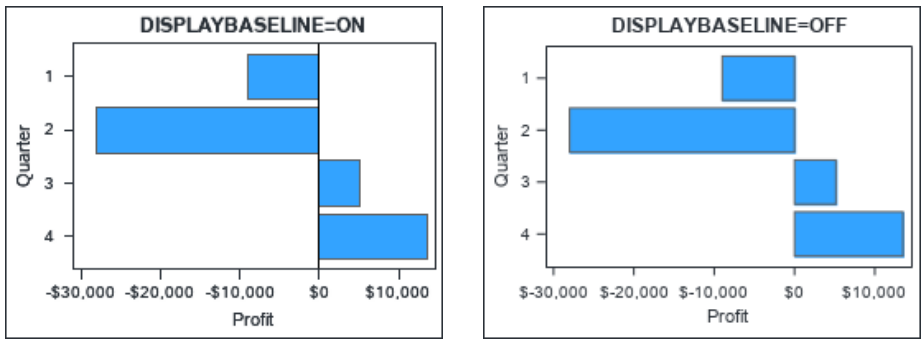
DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.



Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL
specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

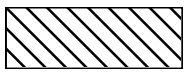
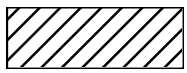
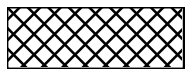
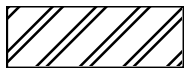
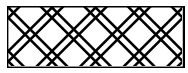
COLOR=*color*
specifies a color to use for the fill lines.

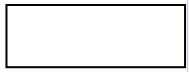
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.15 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interactions If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

When FREQ= and MISSING are both specified, the MISSING option has no effect.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, “Using Attribute Maps to Control Visual Attributes,” on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP= is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default ASCENDING

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

INTERVALBARWIDTH=*numeric-value*<*units*>
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is "Frequency".

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes attributes of the GraphError style element in the current style.

Interaction This option has no effect unless you also specify either the LIMITS= or LIMITSTAT= option.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITS= or LIMITSTAT= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITS=BOTH | LOWER | UPPER
specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending from each bar. Upper limits extend to the right of the bar and lower limits extend to the left of the bar. Specify one of the following values:

BOTH
adds lower and upper limit lines to the plot.

LOWER
adds lower limit lines to the plot.

UPPER
adds upper limit lines to the plot.

Default By default, no limits are displayed unless you specify either the LIMITS= or LIMITSTAT= option. If you specify the LIMITSTAT= option only, then LIMITS=BOTH is the default.

Interactions Limit lines are displayed only when you specify STAT=MEAN.

If you use the GROUP= option in the plot statement, the LIMITS= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines. Specify one of the following statistics:

CLM
confidence limits

STDDEV
standard deviation

STDERR
standard error

Default If you specify the LIMITS= option, then the default for LIMITSTAT= is CLM. Otherwise, there is no default.

Interactions If you specify the LIMITSTAT= option only, then the default value for the LIMITS= option is BOTH.

Limits lines are displayed only when you specify STAT=MEAN.

If you use the GROUP= option in the plot statement, the LIMITSTAT=option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

Tip To display limit labels, specify DATALABEL (without a variable).

MISSING

accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"

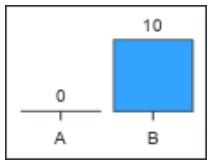
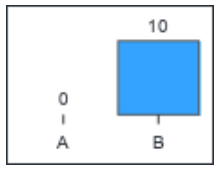
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS

suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

NUMSTD=*n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR. You can specify any positive number, including decimals.

Default 1

OUTLINE | NOOUTLINE

specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphOutlines style element
- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

This option is ignored if the DATASKIN= option is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

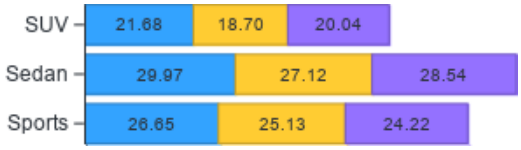
See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed for each value on the horizontal axis.

SEGLABEL

displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)

specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN

specifies a policy for fitting the bar segment labels within the bar segments.

NONE

no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP

does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN

drops any bar segment label that does not fit within its segment.

The label text height must not exceed the bar width, and the label length must not exceed the segment length.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the horizontal axis. Specify one of the following:

FREQ
the frequencies of the response or category variable.

The frequencies are calculated as follows:

- If you specify the RESPONSE= option, FREQ calculates the frequency of the response variable.
- If you do not specify the RESPONSE= option, FREQ calculates the frequency of the category variable.

MEAN
the mean of the response variable.

Interaction For STAT=MEAN to take effect, you must also specify the RESPONSE= option.

MEDIAN
the median of the response variable.

Interaction For STAT=MEDIAN to take effect, you must also specify the RESPONSE= option.

PERCENT
the percentage of the response or category variable.

The percentage is calculated as follows:

- If you specify the RESPONSE= option, PERCENT calculates the percentage of the sum of the response variable.
- If you do not specify the RESPONSE= option, PERCENT calculates the percentage of the frequency of the category variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the =PCTLEVEL= option in the PROC SGPLOT statement.

You can use the PCTNDEC= option in the SGPLOT procedure statement to control the number of decimals to be used when calculating the percent values.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level are zero.

SUM
the sum of the response variable.

This is the default value when you specify the RESPONSE= option.

Interaction For this value to take effect, you must also specify the RESPONSE= option.

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Restriction If you do not also specify the RESPONSE= option, then only the FREQ or PERCENT statistic is calculated (FREQ is the default). If you specify RESPONSE=, then you can use any of the statistics.

Interaction When the graph is generated, the statistic is appended to the variable name in the axis label and the legend (if it is created). However, if a label has been assigned to the variable, then the label appears in the axis label and legend instead of the statistic.

STATLABEL | NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels. STATLABEL forces the statistic to be displayed. NOSTATLABEL removes the statistic from the axis and legend labels.

Normally, the procedure displays the statistic along with the name of the response variable. However, when a custom label is assigned to the response variable, the procedure does not display the statistic. In each case, you can control whether the statistic is displayed.

Defaults The statistic is displayed for the response variable.

When a custom label is assigned to the response variable, the statistic is not displayed.

Interactions This option has no effect unless the RESPONSE= option is specified.

This option has no effect if you specify the axis label using the LABEL= option in an AXIS statement.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces the information that is displayed by default. However, if RESPONSE= is not specified and TIP= is other than NONE, the computed frequency is added to the specified tips.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

If your categorical chart is overlaid with other categorical charts, each chart can specify its own URL variable. The URL variable specified for each chart overlay is honored. This capability affects the following charts: DOT, HBAR, HLINE, VBAR, VLINE.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction If your plot is overlaid with other categorization plots that also specify WEIGHT=, then the first WEIGHT variable that you specified is used for all of the plots.

X2AXIS
assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the category variable to the secondary (right) vertical axis.

HBARBASIC Statement

Creates a horizontal bar chart that is compatible with other categorization charts as well as basic plots, such as scatter and series plots, and box plots.

Notes: When using the HBARBASIC statement with a procedure that uses the BY statement, axes are not guaranteed to be uniform across BY groups.

The bars are sorted in data order based on the category values. The sort order cannot be changed. This arrangement of the bars differs from the HBAR statement, which by default sorts bars in alphabetical order based on the category values.

Tip: The HBARBASIC statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

See: "Plot Compatibility" on page 1780

Example: "About Bar Charts" on page 53

Syntax

HBARBASIC *category-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
: specifies whether the baseline is displayed.

FILL

NOFILL
: specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLENDCOLOR=*color*
: specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
: specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
: specifies the thickness of the bars when the category data is numeric.

NOZEROBARS
: suppresses zero-length bars.

OUTLINE

NOOUTLINE
specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

X2AXIS
assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the category variable to the secondary (right) vertical axis.

Data tip options

TIP=(*role-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL
displays the bar statistic value for each bar.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL option.

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
specifies the fit policy for the bar labels.

DATALABELFORMAT=*format*
specifies the text format used to display the bar label.

LEGENDLABEL="*text-string*"

specifies the label that identifies the bar chart in the legend.

SEGLABEL
displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Plot options

MISSING
accepts a missing value as a valid category value.

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Statistics options

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors.

STAT=FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic for the horizontal axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the bars. he variable generates the midpoints to which each observation in the data set contributes.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

BARWIDTH=*numeric-value*

specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Defaults .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

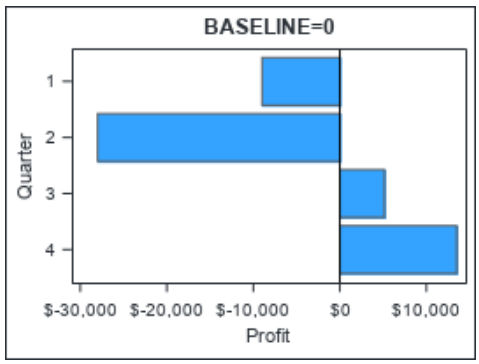
Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*

specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interaction When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified and when GROUPDISPLAY=CLUSTER.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR specifies the color for the smallest data value of the COLORRESPONSE= column.

NEUTRALCOLOR specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.

ENDCOLOR specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Requirement The COLORRESPONSE= option must also be specified, or COLORSTAT= must be FREQ or PCT. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Interaction This option is ignored if COLORSTAT=FREQ or COLORSTAT=PCT.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

When COLORRESPONSE= is specified, the following values are valid:

SUM
sum values for the color response

MEAN
mean values for the color response

MEDIAN
median values for the color response

Default SUM when option COLORRESPONSE= is also specified.

Requirement The COLORRESPONSE= option must also be specified. If COLORRESPONSE= is not specified, COLORSTAT= is ignored.

Note This option is independent of the STAT= and RESPONSE= options.

DATALABEL
displays the bar statistic value for each bar. For grouped clustered bars, each bar is labeled with the summarized value of the bar. For grouped stacked bars, the segmented bar is labeled with the accumulated, summarized value of all the bar segments.

Default No label is shown

Interaction By default, the data label fit policy is to show the labels unless they collide. As a result, the labels sometimes might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

Tip The font and color attributes for the label are specified by the DATALABELATTRS= option. The text format is specified by the DATALABELFORMAT= option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
```

```
Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:

```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
specifies the fit policy for the bar labels. Specify one of the following values:

NONE
specifies that no fit policy is implemented for the bar labels.

By default, the fit policy is to show the labels unless they collide. As a result, the labels might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

INSIDEPREFERRED
places bar labels inside the bars, when possible.

Labels are placed outside the bars in the following cases:

- If a label cannot fit within the length of its respective bar, then that label is moved outside the bar.
- If the font height for any label exceeds the bar width, then all labels are positioned outside the bars.
- If SEGLABEL is specified, the INSIDEPREFERRED policy is ignored and the labels are positioned outside the bars.

Default Show the labels unless they collide.

Interaction This option has no effect unless DATALABEL is also specified.

DATALABELFORMAT=*format*
specifies the text format used to display the bar label.

Default The column format assigned to the RESPONSE= column, or BEST6 if no format is assigned.

Interaction This option has no effect unless DATALABEL= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.16 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*

specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

DISPLAYBASELINE=AUTO | ON | OFF

specifies whether the baseline is displayed. Specify one of the following values:

AUTO

displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

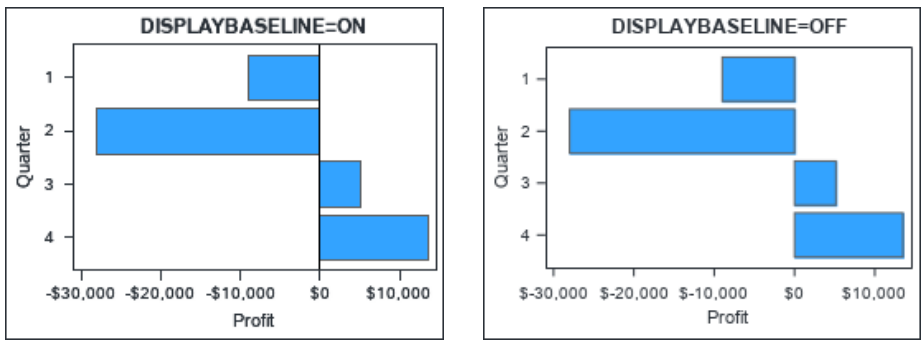
In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON

always displays the baseline.

OFF

hides the baseline.



Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL
specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interaction This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note
: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips
: Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

: Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

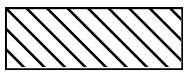
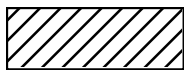
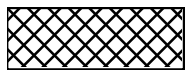
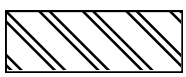
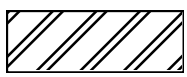
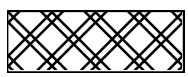
COLOR=*color*
specifies a color to use for the fill lines.

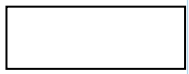
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.17 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, “Using Attribute Maps to Control Visual Attributes,” on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP= is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INTERVALBARWIDTH=*numeric-value<units>*
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is “Frequency”.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

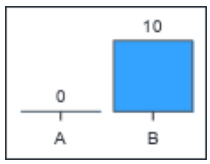
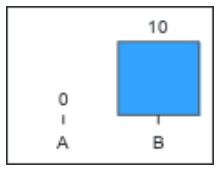
MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS
suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphOutlines style element
- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

This option is ignored if the DATASKIN= option is also specified.

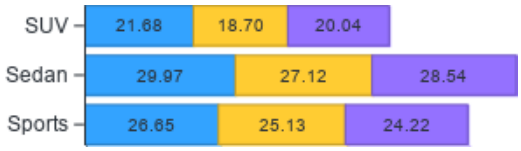
RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed for each value on the horizontal axis.

SEGLABEL
displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

NONE
no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP
does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN
drops any bar segment label that does not fit within its segment.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

STAT=FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic for the horizontal axis. For bar charts with no RESPONSE= variable, the following values are valid:

FREQ
frequency count

PERCENT
percentages between 0 and 100

Alias PCT

PROPORTION
proportions between 0 and 1

For bar charts with a RESPONSE= variable, the following values are valid:

SUM
sum values for the response

MEAN
mean values for the response

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Interaction When this option is used with the GROUP=group option, the specified statistic is computed for each segment that is created for the unique group values.

TIP=(*role-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*role-list*)
a space-separated list of unique chart roles enclosed in parentheses.

The roles for TIP are CATEGORY, GROUP, and RESPONSE. Specify one or more of these role names in *role-list*. Data tips are displayed using the data obtained from the specified roles.

Note: You must specify the GROUP and RESPONSE roles for the chart in order to use those roles for data tips.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(category response)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option. Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see “Using the ODS GRAPHICS Statement” on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the category variable to the secondary (right) vertical axis.

HBARPARM Statement

Creates a horizontal bar chart based on a pre-summarized response value for each unique value of the category variable. You can also assign variables to the upper and lower limits.

Requirements: Parameterized bar charts require a response variable in addition to the category variable. The response variable contains pre-summarized computed values such as a sum or a mean for each unique value of the category variable. This is the key distinction between HBARPARM and HBAR.

The data must contain only one response value per unique category variable. If more than one value is found, a warning is written to the SAS log, and the graph might produce unpredictable results.

Interaction: The HBARPARM statement can be combined only with other basic plot statements in the SGPLOT procedure. See "Plot Compatibility" on page 1780.

Note: The bars are sorted in data order based on the category values. The sort order cannot be changed. This arrangement of the bars differs from the HBAR statement, which by default sorts bars in alphabetical order based on the category values.

Tip: The HBARPARM statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

Example: "About Bar Charts" on page 53

Syntax

HBARPARM CATEGORY=*category-variable* RESPONSE=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed.

FILL

NOFILL
specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
specifies the thickness of the bars when the category data is numeric.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend.

NOZEROBARS
suppresses zero-length bars.

OUTLINE

NOOUTLINE
specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

X2AXIS
assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the category variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
specifies the fit policy for the bar labels.

DATALABELPOS=DATA | LEFT | RIGHT
specifies the location of the data label.

SEGLABEL
displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Limit options

LIMITATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*

specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITLOWER=*numeric-variable*
specifies values for the lower endpoints on the limit lines.

LIMITUPPER=*numeric-variable*
specifies values for the upper endpoints on the limit lines.

Plot options

MISSING
accepts a missing value as a valid category value.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME=*"text-string"*
assigns a name to a plot statement.

Required Arguments

CATEGORY=*category-variable*
specifies the variable that categorizes the data. All values are treated as discrete values. The input data for this variable should contain unique values. When the category values are not unique, a warning is logged, and multiple bars are superimposed at the duplicated category values. The CATEGORY axis is always discrete.

Interval bar charts are supported when the category axis is set to TYPE=LINEAR.

RESPONSE=*numeric-variable*
specifies a numeric response variable. The input data is expected to be pre-summarized computed values (sum, mean, and so on).

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Defaults .8

1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

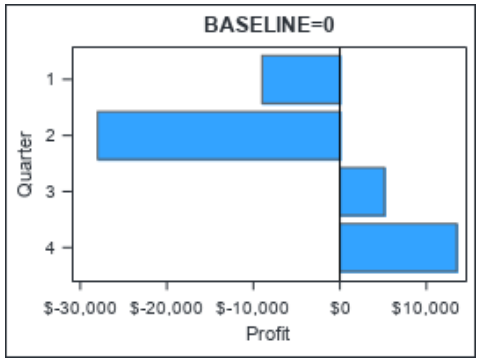
Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*

specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interaction When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style

element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CLUSTERWIDTH=*numeric-value*

specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified and when GROUPDISPLAY=CLUSTER.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)

specifies a color ramp that is to be used to determine response colors.

style-element

specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

DATALABEL
DATALABEL=*variable*
displays a label for each data point.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITLOWER or LIMITUPPER, then the limit labels are also shown. Limit labels are shown when you use DATALABEL (without a variable).

This option is supported with the GROUP= option only when GROUPDISPLAY=CLUSTER is specified. Otherwise, the label is not drawn. The labels are shown if you also specify the DATALABELPOS= option.

By default, the data label fit policy is to show the labels unless they collide. As a result, the labels sometimes might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=NONE | INSIDEPREFERRED
specifies the fit policy for the bar labels. Specify one of the following values:

NONE
specifies that no fit policy is implemented for the bar labels.

By default, the fit policy is to show the labels unless they collide. As a result, the labels might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

INSIDEPREFERRED
places bar labels inside the bars, when possible.

Labels are placed outside the bars in the following cases:

- If a label cannot fit within the length of its respective bar, then that label is moved outside the bar.
- If the font height for any label exceeds the bar width, then all labels are positioned outside the bars.

- If SEGLABEL is specified, the INSIDEPREFERRED policy is ignored and the labels are positioned outside the bars.

Default Show the labels unless they collide.

Interaction This option has no effect unless DATALABEL= is also specified.

DATALABELPOS=DATA | LEFT | RIGHT
specifies the location of the data label. Specify one of the following values:

DATA
places the label on the data primitives (at the right edge of the bars).

LEFT
places the label to the left of the bars on the left edge of the data area.

RIGHT
places the label to the right of the bars on the right edge of the data area.

Default DATA

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays limit information when limits are specified.

This option displays group values for each category when GROUP= is also specified.

This option displays response values for each overlaid chart.

This option does not support the splitting or rotation of data labels.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.18 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	**PRESSED**	**SHEEN**

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

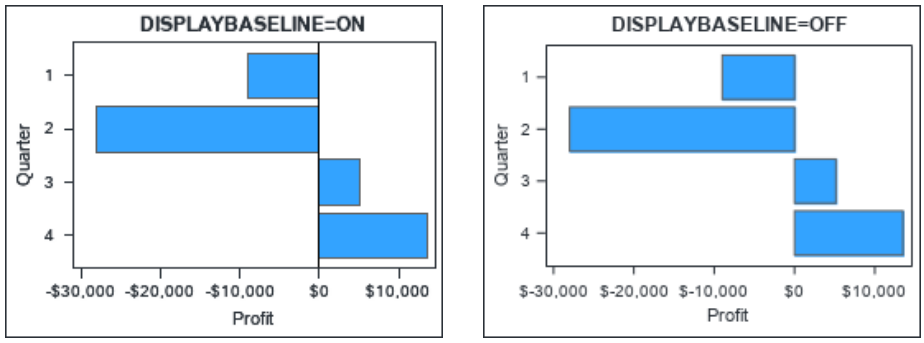
DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.



Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)

specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*

specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

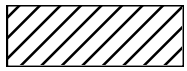
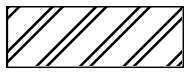
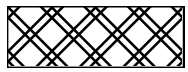
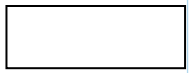
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.19 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions

For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note

This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, “Using Attribute Maps to Control Visual Attributes,” on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes

derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP= is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted

values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

INTERVALBARWIDTH=*numeric-value<units>*
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITLOWER= or LIMITUPPER= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITLOWER=*numeric-variable*
specifies values for the lower endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default The lower segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions If LIMITUPPER= is also specified, then the plot displays the lower and upper segments of the limit lines.

If you use the GROUP= option in the plot statement, the LIMITLOWER= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITUPPER=*numeric-variable*
specifies values for the upper endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default The upper segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions If LIMITLOWER= is also specified, then the plot displays the lower and upper segments of the limit lines.

If you use the GROUP= option in the plot statement, the LIMITLUPPER= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

Tip To display limit labels, specify DATALABEL (without a variable).

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

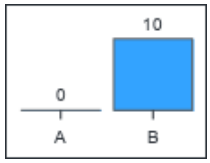
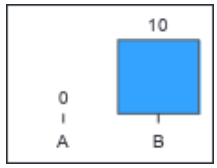
NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS
suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a

simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphOutlines style element

- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

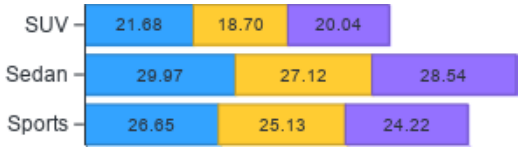
This option is ignored if the DATASKIN= option is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SEGLABEL
displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

NONE
no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP
does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN
drops any bar segment label that does not fit within its segment.

The label text height must not exceed the bar width, and the label length must not exceed the segment length.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Notes When TIP= is not specified, the CATEGORY, RESPONSE, COLORRESPONSE, and GROUP variables are displayed, if applicable, when you hover over a bar. The ERRORUPPER or ERRORLOWER variable is displayed when you hover over an error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a bar. No variables are displayed when you hover over an error bar or error bar cap.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see “Using the ODS GRAPHICS Statement” on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the category variable to the secondary (right) vertical axis.

HBOX Statement

Creates a horizontal box plot that shows the distribution of your data.

Restriction: This plot has compatibility restrictions when used with other plots. See Table 13.2 on page 1781.

See: "Visual Description of Box Plot Percentile Boundaries" on page 1094

Examples: "About Box Plots" on page 47
"Example 9: Creating a Horizontal Box Plot" on page 1670

Syntax

HBOX *numeric-analysis-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

BOXWIDTH=*numeric-value*
specifies the width of the box.

CAPSCALE=*numeric-value*
specifies a positive number to use as a multiplier for changing the default width of the whisker caps.

CAPSHAPE=BRACKET | LINE | SERIF | NONE
specifies the shape of the whisker cap lines.

CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
specifies that a connect line joins a statistic from box to box.

CONNECTATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the lines that connect multiple boxes.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all boxes from the discrete tick marks.

EXTREME
specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified.

FILL
NOFILL
specifies whether the boxes are filled with color.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency.

FILLPATTERN
enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern.

INTERVALBOXWIDTH=*numeric-value*
specifies the box width when an interval category (Y) variable is specified.

LINEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the box outlines.

MEANATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the marker that represents the mean in the box.

MEDIANATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the line that represents the median.

NOCAPS
hides the cap lines for the whiskers.

NOMEAN
hides the mean marker.

NOMEDIAN
hides the median line.

NOOUTLIERS
hides the outliers from the plot.

NOTCHES
specifies that the boxes be notched.

OUTLIERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the marker that represents the outliers.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

WHISKERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the whisker and cap lines.

WHISKERPCT=*number*
specifies the whisker length, in percentile units.

Axis options

X2AXIS
assigns the analysis variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the category variable to the secondary (right) vertical axis.

Data tip options

TIP=(*role-list*) | NONE
specifies the information to display when the cursor is positioned over a box or whisker in the box plot.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the midpoint spacing.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped boxes.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
adds data labels for the outlier markers.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

LABELFAR
specifies that only the far outliers have data labels.

LEGENDLABEL="*text-string*"
specifies a label that identifies the box plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

CATEGORY=*category-variable*
specifies the category variable for the plot.

DISPLAYSTATS=ALL | (*statistics-list*)
specifies statistics to be displayed for each box.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

MISSING

accepts a missing value as a valid category value.

PERCENTILE=1 | 2 | 3 | 4 | 5
specifies a method for computing the percentiles for the plot.

SPREAD
relocates outlier points that have identical values to prevent overlapping.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Argument

numeric-analysis-variable
specifies the analysis variable for the plot. If you do not specify the CATEGORY= option, then one box is created for the analysis variable.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BOXWIDTH=*numeric-value*
specifies the width of the box. Specify a value between 0.0 (0% of the available width) and 1.0 (100% of the available width).

Defaults 0.4

When GROUP is specified, the default box width is 0.6.

CAPSCALE=*numeric-value*
specifies a positive number to use as a multiplier for changing the default width of the whisker caps. For example, CAPSCALE=2 doubles the default cap width and CAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction This option has no effect if NOCAPS is specified or if CAPSHAPE=NONE.

Tips The thickness (height) of a SERIF, LINE, or BRACKET cap honors the line thickness specified in the WHISKERATTRS= option. The cap pattern is always solid.

Use the CAPSHAPE= option to specify the line shape of the caps.

CAPSHAPE=BRACKET | LINE | SERIF | NONE
specifies the shape of the whisker cap lines. Specify one of the following values:

BRACKET
displays a straight line with brackets.

LINE
displays a straight line.

SERIF
displays a short straight line.

NONE
does not display a cap.

Default SERIF

CATEGORY=*category-variable*
specifies the category variable for the plot. A box plot is created for each distinct value of the category variable.

If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete.

The default ordering of the tick values is ascending formatted order of the *category-variable*. To change this ordering, you can use the DISCRETEORDER=DATA option in the category axis statement.

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the midpoint spacing. Specify a value from 0.1 (narrowest) to 1.0 (widest).

Default 0.7

Interaction This option is applicable only when a GROUP is in effect and the category axis is discrete.

CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
specifies that a connect line joins a statistic from box to box.

Interaction This option applies only when the CATEGORY option is used to generate multiple boxes.

Tip You can use the CONNECTATTRS option to specify attributes for the connect line.

CONNECTATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the lines that connect multiple boxes. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphConnectLine style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interactions This option takes effect only if the CONNECT= option is also specified.

This option is ignored if the GROUP= option is also specified.

Examples
```
CONNECTATTRS=(Color="light green" Pattern=MediumDash
        Thickness=4)
```

This example specifies a style element:
```
CONNECTATTRS=GraphData3
```

DATALABEL
DATALABEL=*variable*
adds data labels for the outlier markers. If you specified a variable, then the values for that variable are used for the data labels. If you did not specify a variable, then the values of the analysis variable are used.

Note This option has no effect unless the plot contains outlier points.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled boxes. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work

best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.20 *DATASKIN Options for Box Plots*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interaction If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all boxes from the discrete tick marks.

Specify a value from -0.5 (left offset) to +0.5 (right offset). If you specify a value outside of this range, an error message appears in the SAS log and the graph is not produced.

Default 0.0 (no offset)

DISPLAYSTATS=ALL | (*statistics-list*)
specifies statistics to be displayed for each box.

ALL
displays all available statistics.

See the *statistics-list*.

Note: When you choose this option, there might not be enough space for the boxes to be drawn. For horizontal box plots, you might have better results if you specify DISPLAYSTATS by itself or specify a limited set of statistics using DISPLAYSTATS=(*statistics-list*).

statistics-list
specifies a space-separated list of statistics enclosed in parentheses.

Specify one or more of the following:

DATAMAX	maximum data value that includes not only the maximum whisker values but also the maximum outlier values. The DATAMAX value is greater than or equal to the MAX value and is always represented in the axis range.
DATAMIN	minimum data value that includes not only the minimum whisker values but also the minimum outlier values. The DATAMIN value is less than or equal to the MIN value and is always represented in the axis range.
IQR	interquartile range (Q3–Q1).
MAX	maximum data value below the box upper fence.
MEAN	mean data value for the box.
MEDIAN	median data value for the box.
MIN	minimum data value above the box lower fence.
N	number of observations for the box.
Q1	lower quartile (25th percentile) for the box.
Q3	upper quartile (75th percentile) for the box.
RANGE	range of the data (MAX–MIN).
STD	standard deviation of the data for the box.
SUMWGT	sum of the weights for the box.

Example `displaystats=(mean median)`

Restriction This option is ignored if the GROUP= option is also specified.

EXTREME
specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified. When you do not specify the EXTREME option, the whiskers cannot be longer than 1.5 times the length of the box.

FILL | NOFILL
specifies whether the boxes are filled with color. The FILL option shows the fill color. The NOFILL option hides the fill color.

Default FILL

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color

attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

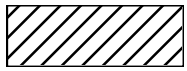
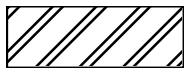
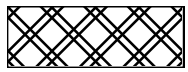
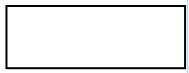
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.21 Fill Patterns

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FREQ=*numeric-variable*

specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interaction If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped boxes.

CLUSTER
the boxes are drawn adjacent to each other.

OVERLAY
all the boxes for a given group value are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Defaults CLUSTER for a discrete category axis

OVERLAY for a linear axis

Interaction This option is ignored unless GROUP= is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INTERVALBOXWIDTH=*numeric-value*
specifies the box width when an interval category (Y) variable is specified.

Alias INTBOXWIDTH=

Restriction The axis type for the category axis must be LINEAR, and the variable must be numeric.

Note INTERVALBOXWIDTH= is valid starting with 2024.05. In 2024.04 and prior releases, use INTBOXWIDTH= instead.

Example
```
proc sgplot data=sashelp.class;
  hbox weight / category=height intboxwidth=20 ;
  yaxis type=linear;
run;
```

LABELFAR
specifies that only the far outliers have data labels. Far outliers are points whose distance from the box is more than three times the length of the box.

Note This option has no effect if you do not specify the DATALABEL option, or if there are no far outliers.

LEGENDLABEL="*text-string*"
specifies a label that identifies the box plot in the legend. By default, the label of the analysis variable is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the box outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

Interaction This option is ignored if the DATASKIN= option is also specified.

MEANATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker that represents the mean in the box. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For grouped data, the GraphBoxMean style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

MEDIANATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the line that represents the median. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphBoxMedian style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interaction This option is ignored if the NOMEDIAN option is also specified.

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOCAPS
hides the cap lines for the whiskers.

Interaction Using several options that hide box features can cause the NOCAPS option to be ignored. For example, if you use NOCAPS, NOFILL, NOMEAN, NOMEDIAN, and NOOOUTLIERS in the same statement, the NOCAPS option might be ignored.

NOMEAN
hides the mean marker.

NOMEDIAN
hides the median line.

NOOUTLIERS
hides the outliers from the plot.

NOTCHES
specifies that the boxes be notched. See "Notches" on page 1095.

OUTLIERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker that represents the outliers. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphOutlier style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

PERCENTILE=1 | 2 | 3 | 4 | 5
specifies a method for computing the percentiles for the plot.

Default 5 (empirical distribution function with averaging)

See "Calculating Percentiles" on page 1096

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear

consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPREAD
relocates outlier points that have identical values to prevent overlapping.

Note This option has no effect if your data does not contain two or more outliers with identical values for the analysis variable.

TIP=(*role-list*) | NONE
specifies the information to display when the cursor is positioned over a box or whisker in the box plot.

(role-list)
a space-separated list of unique box plot roles enclosed in parentheses.

The box plot roles for TIP are CATEGORY, N, STD, MIN, MAX, MEAN, MEDIAN, Q1, and Q3. Specify one or more of these role names in *role-list*. Data tips are displayed using the data obtained from the specified roles.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(mean median)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option. Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

WHISKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the whisker and cap lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphBoxWhisker style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

WHISKERPCT=*number*
specifies the whisker length, in percentile units. When this option is specified, *number* is used as the low percentile, and 100–*number* is used as the high percentile.

Here are some examples of values and their effect:

0 specifies the high and low extremes

10 specifies the 10th percentile low and the 90th percentile high

25 specifies the 25th percentile low and the 75th percentile high

Default The whiskers are drawn from the box to the most extreme point that is less than or equal to 1.5 times the IQR

Range 0–25

Notes When this option is specified, fences and far outliers are not drawn.

When this option is set to 25, no whiskers are drawn because the box extends from the 25th to the 75th percentile.

X2AXIS
assigns the analysis variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the category variable to the secondary (right) vertical axis.

Details

Statement Summary

The plot displays a single box if only the analysis variable is provided. The plot displays multiple boxes if a category variable is also provided and that variable has more than one unique value.

The ANALYSIS variable is displayed on the horizontal axis. The axis for the analysis column is always LINEAR.

By default for numeric or character columns, the CATEGORY= axis is TYPE=DISCRETE. You can override the default and set TYPE=LINEAR in the axis statement, provided that the category column is numeric.

If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete. For the interval case, you can use the INTBOXWIDTH= option to specify the box width.

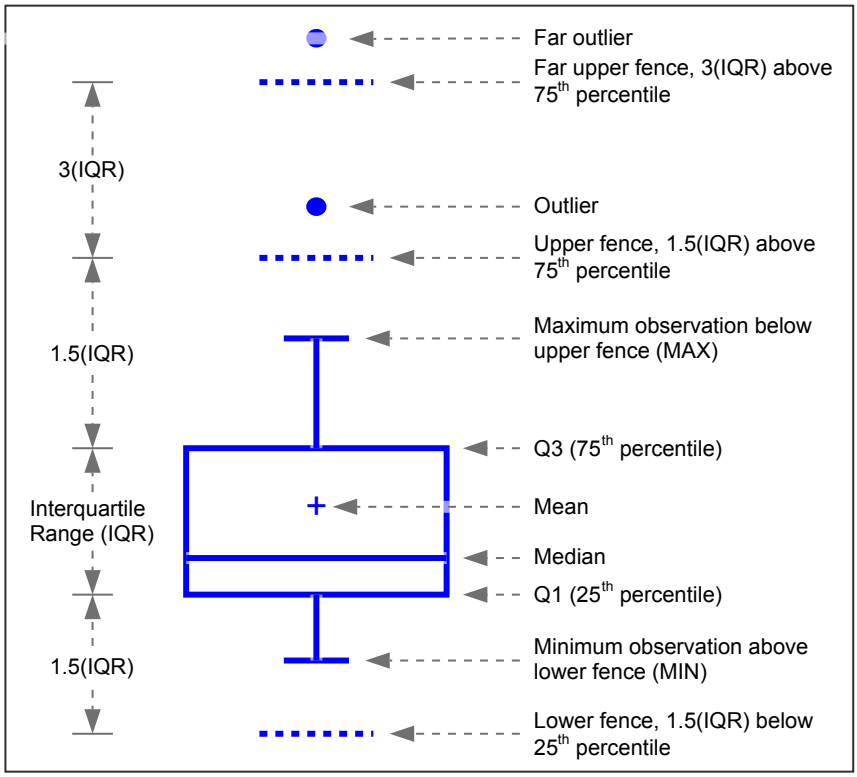
Two basic box plot representations can be drawn: a schematic (Tukey) box plot and a skeletal box plot. See the EXTREME option for details.

Visual Description of Box Plot Percentile Boundaries

Box plots display the distribution of data by using a rectangular box and whiskers. Whiskers are lines that indicate a data range outside of the box.

Note: Although the following figure shows a vertical box plot, the basic concepts apply to horizontal box plots.

Figure 10.3 *Parts of a Box Plot*



In the previous figure, the bottom and top edges of the box indicate the intra-quartile range (IQR). That is the range of values between the first and third quartiles (the 25th and 75th percentiles). The marker inside the box indicates the mean value. The line inside the box indicates the median value.

The elements that are outside the box are dependent on your options. By default, the whiskers that extend from each box indicate the range of values that are outside of the intra-quartile range. However, they are close enough not to be considered outliers (a distance less than or equal to 1.5*IQR). If you specify the EXTREME option, then the whiskers indicate the entire range of values, including outliers.

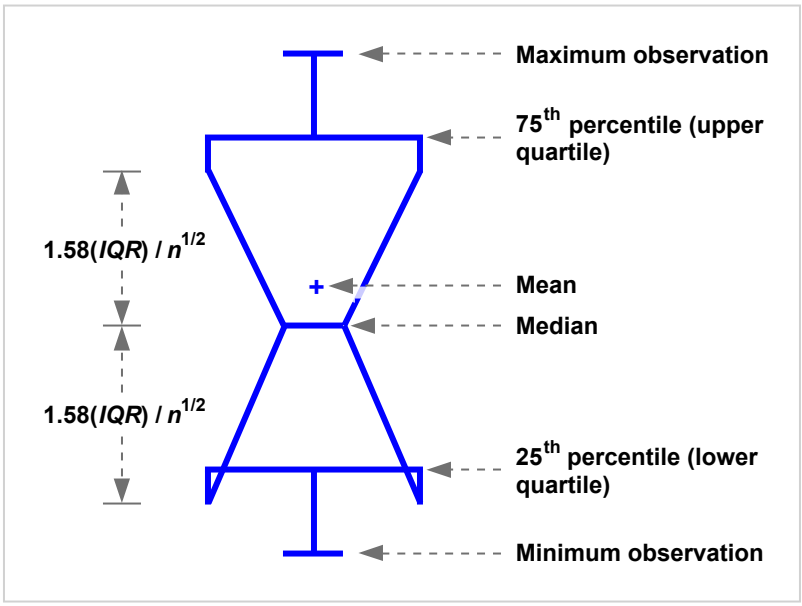
Outliers are observations that are more extreme than the upper and lower fences ($\pm\ 1.5\ IQR$). Outliers that are beyond upper and lower far fences ($\pm\ 3\ \ IQR$) are called FAR OUTLIERS. By default, outliers are indicated by markers. If you specify the DATALABEL= option, then the outlier points have data labels. If you also specify the LABELFAR option, then only outliers that are 3*IQR from the box have data labels.

Notches

In a notched box plot, the endpoints of the notches are at the following computed locations:

$$median\ \pm\ 1.58\ (IQR/\sqrt{N})$$

In the equation, IQR (IQR=Q3-Q1) is the interquartile range and *n* is the sample size. The medians (central lines) of the two boxes are significantly different at approximately the 0.05 level if the corresponding notches do not overlap.



Note: In some cases, the notches in the box plot can extend beyond DATAMIN and DATAMAX.

Calculating Percentiles

You can specify one of five definitions for computing the percentiles with the PERCENTILE= option. Let n be the number of nonmissing values for a variable, and let $X_1, X_2, ..., X_n$ represent the ordered values of the variable. X_1 is the smallest value, X_2 is the next smallest, and X_n is the largest value. Let the tth percentile be y, set:

$$p = \frac{t}{100}$$

and let:

$$np = j + g$$

when PERCENTILE=1, 2, 3, or 5, or let:

$$(n+1)p = j + g$$

when PERCENTILE=4, where j is the integer part of np, and g is the fractional part of np. Then the PERCENTILE= option defines the tth percentile, y, as described in the following table:

Percentile	Description	Equation	Notes
1	Weighted average at X_{np}	$y = (1-g)x_j + gx_{j+1}$	x_0 is taken to be x_1

Percentile	Description	Equation	Notes
2	Observation numbered closest to *np*	$y = x_j$	Used when $g < \frac{1}{2}$ or when $g = \frac{1}{2}$ and *j* is even
		$y = x_{j+1}$	Used when $g = \frac{1}{2}$ and *j* is odd or when $g > \frac{1}{2}$
3	Empirical distribution function	$y = x_j$	Used when $g = 0$
		$y = x_{j+1}$	Used when $g > 0$
4	Weighted average aimed at $x_{(n+1)p}$	$y = (1-g)x_j + gx_{j+1}$	x_{n+1} is taken to be x_n
5	Empirical distribution function with averaging	$y = \frac{1}{2}(x_j + x_{j+1})$	Used when $g = 0$
		$y = x_{j+1}$	Used when $g > 0$

HEATMAP Statement

Creates a plot of color-coded rectangles for the response variable of a pair of X and Y variables after it bins the data in two dimensions.

Interaction: The number of bins is limited to 100,000. If the number of bins exceeds the limit, the heat map is not drawn. A warning message is written to the SAS log in that case. The limit is specified by the ODS GRAPHICS statement NXYBINSMAX= option, which defaults to 100,000. See NXYBINSMAX= on page 115.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Heat Maps" on page 21

Syntax

HEATMAP X=*variable* Y=*variable* </*options*>;

Summary of Optional Arguments

Appearance options

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-variable*
: specifies a numeric variable that is used to color the regions of the heat map.

COLORSTAT=FREQ | PCT | SUM | MEAN
: specifies the statistic to use for the COLORRESPONSE= variable.

FILLATTRS=(TRANSPARENCY=*number*)
: specifies the transparency of the area fill in the rectangles.

OUTLINE
: displays an outline around each colored region.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the rectangle outlines.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

SHOWXBINS
: specifies that bins be used as the basis for the X-axis tick marks.

SHOWYBINS
: specifies that bins be used as the basis for the Y-axis tick marks.

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

XENDLABELS
: specifies that axis ticks and value labels are drawn at the endpoints of the bins for the X axis.

Y2AXIS
: assigns the Y variable to the secondary (right) vertical axis.

YENDLABELS
: specifies that axis ticks and value labels are drawn at the endpoints of the bins for the Y axis.

Binning options

NXBINS=*positive-integer*
: specifies the number of bins to use for the X role.

NYBINS=*positive-integer*
: specifies the number of bins to use for the Y role.

XBINSIZE=*positive-number*
: specifies the horizontal size of bins in X-axis coordinate system units.

XBINSTART=*number*
: specifies the data value for the first bin of the X role.

YBINSIZE=*positive-number*

specifies the vertical size of bins in Y-axis coordinate system units.

YBINSTART=*number*
specifies the data value for the first bin of the Y role.

Data options

DISCRETEX
forces the X axis to be discrete when the X= variable is numeric.

DISCRETEY
forces the Y axis to be discrete when the Y= variable is numeric.

Data tip options

TIP=(*role-list*) | NONE
specifies the information to display when the cursor is positioned over a rectangle.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Plot options

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Requirement If you specify a numeric variable and the X-axis type is discrete, then you must also specify DISCRETEX in the HEATMAP statement. Otherwise, the heat map might not be drawn correctly.

Y=*variable*
specifies the variable for the Y axis.

Requirement If you specify a numeric variable and the Y-axis type is discrete, then you must also specify DISCRETEY in the HEATMAP statement. Otherwise, the heat map might not be drawn correctly.

Optional Arguments

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element

COLORRESPONSE=*numeric-variable*
specifies a numeric variable that is used to color the regions of the heat map.

Tip The color ramp is specified by the COLORMODEL= option. The color ramp represents the range of unique response values.

COLORSTAT=FREQ | PCT | SUM | MEAN
specifies the statistic to use for the COLORRESPONSE= variable. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

When COLORRESPONSE= is specified, the following values are valid:

SUM
sum values for the color response

MEAN
mean values for the color response

Defaults FREQ if COLORRESPONSE= is not specified

SUM if COLORRESPONSE= is specified

DISCRETEX
forces the X axis to be discrete when the X= variable is numeric.

Requirement If X= specifies a numeric variable and the X-axis type is discrete, then you must specify DISCRETEX. Otherwise, the heat map might not be drawn correctly.

Interactions If X= specifies a character variable, then this option is ignored, and the X axis is considered to be discrete.

When this option is used, the XENDLABELS= option is ignored.

DISCRETEY
forces the Y axis to be discrete when the Y= variable is numeric.

Requirement If Y= specifies a numeric variable and the Y-axis type is discrete, then you must specify DISCRETEY. Otherwise, the heat map might not be drawn correctly.

Interactions If Y= specifies a character variable, then this option is ignored, and the Y axis is considered to be discrete.

When this option is used, the YENDLABELS= option is ignored.

FILLATTRS=(TRANSPARENCY=*number*)
specifies the transparency of the area fill in the rectangles.

Default The TRANSPARENCY= option value.

Range 0 (opaque) to 1 (entirely transparent)

Interaction This option overrides the TRANSPARENCY= option for the colored region only. It does not affect the region outlines.

Tip You can combine this option with TRANSPARENCY= to set one transparency for the outline but a different transparency for the fill. Example:

```
outline transparency=0.2 fillattrs=(transparency=0.6)
```

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NXBINS=*positive-integer*
specifies the number of bins to use for the X role. The system determines the XBINSIZE= and XBINSTART= values, if not specified. The bins always span the range of the data.

Default Determined by the system.

Range 2 or more

Interaction NXBINS is ignored if SHOWXBINS is also specified.

See "Binning Options" on page 1106

NYBINS=*positive-integer*
specifies the number of bins to use for the Y role. The system determines the YBINSIZE= and YBINSTART= values, if not specified. The bins always span the range of the data.

Default Determined by the system.

Range 2 or more

Interaction NYBINS is ignored if SHOWYBINS is also specified.

See "Binning Options" on page 1106

OUTLINE
displays an outline around each colored region.

Default No outline is displayed

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the rectangle outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphOutlines style element in the current style.

Interaction This option has no effect unless OUTLINE is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SHOWXBINS
specifies that bins be used as the basis for the X-axis tick marks.

Default Without this option, a standard axis is used, ignoring bin boundaries and midpoints.

Restriction This option is ignored when a BY variable is used and uniform axis scaling is specified. (You specify uniform axis scaling using the UNIFORM= option in the PROC SGPLOT statement.)

Interaction When this option is used, the XENDLABELS= option determines how the axis ticks and value labels are displayed. When this option is not specified, the XENDLABELS= option is ignored.

SHOWYBINS
specifies that bins be used as the basis for the Y-axis tick marks.

Default Without this option, a standard axis is used, ignoring bin boundaries and midpoints.

Restriction This option is ignored when a BY variable is used and uniform axis scaling is specified. (You specify uniform axis scaling using the UNIFORM= option in the PROC SGPLOT statement.)

Interaction When this option is used, the YENDLABELS= option determines how the axis ticks and value labels are displayed. When this option is not specified, the YENDLABELS= option is ignored.

TIP=(*role-list*) | NONE
specifies the information to display when the cursor is positioned over a rectangle.

(*role-list*)
a space-separated list of unique heat map roles enclosed in parentheses.

The roles for TIP are X, Y, and COLORRESPONSE. Specify one or more of these role names in *role-list*. Data tips are displayed using the data obtained from the specified roles.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(x y)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option. Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

XBINSIZE=*positive-number*
specifies the horizontal size of bins in X-axis coordinate system units. The system determines the NXBINS= and XBINSTART= values, if not specified. The bins always span the X data range.

Default Determined by the system.

See "Binning Options" on page 1106

XBINSTART=*number*
specifies the data value for the first bin of the X role. The system determines the NXBINS= and XBINSIZE= values, if not specified. The bins always span the X data range.

Default Determined by the system.

See "Binning Options" on page 1106

XENDLABELS
specifies that axis ticks and value labels are drawn at the endpoints of the bins for the X axis.

Default If this option is not used, then the axis ticks and value labels are drawn at the bin midpoints.

Interactions This option is ignored when the X axis is discrete.

This option has no effect unless the SHOWXBINS option is also used.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

YBINSIZE=*positive-number*
specifies the vertical size of bins in Y-axis coordinate system units. The system determines the NYBINS= and YBINSTART= values, if not specified. The bins always span the Y data range.

Default Determined by the system.

See "Binning Options" on page 1106

YBINSTART=*number*
specifies the data value for the first bin of the Y role. The system determines the NYBINS= and YBINSIZE= values, if not specified. The bins always span the Y data range.

Default Determined by the system.

See "Binning Options" on page 1106

YENDLABELS
specifies that axis ticks and value labels are drawn at the endpoints of the bins for the Y axis.

Default If this option is not used, then the axis ticks and value labels are drawn at the bin midpoints.

Interactions This option is ignored when the Y axis is discrete.

This option has no effect unless the SHOWYBINS option is also used.

Details

Binning Options

Overview of Binning Options

In a heat map, each variable range is subdivided into equal size bins to create a rectangular grid of bins. The number of observations that fall into each bin is computed, and the grid is displayed by coloring each bin with a shade of color computed from a color gradient. You can use the COLORRESPONSE= option to specify the variable that is used for the color gradient. You can also use the COLORSTAT= option to specify the statistic to apply to the variable.

Note: Although the examples in this section were created in the SGPLOT procedure, the general concepts apply also to heat maps in the SGPANEL procedure.

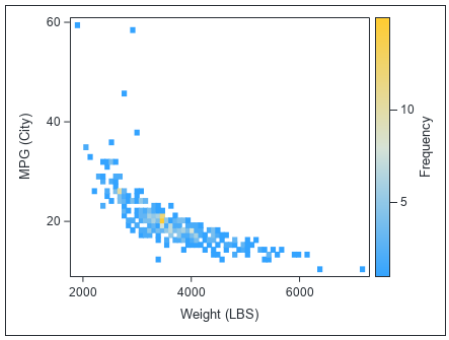
Heat maps have several options for controlling how bins are used to represent the data. You can do the following:

- specify the number of bins (SGPLOT procedure only)
- specify the bin size
- specify the bin start values

The following sections describe the binning options and show how they interact.

The examples are all based on the following simple program:

Table 10.22 *Example Program for Binning Options*




```
ods graphics / reset=all;
ods graphics / width=4.5in;

proc sgplot data=sashelp.cars;
  heatmap x=weight y=mpg_city;
run;
```

Specifying the Number of Bins

The NXBINS= and NYBINS= options specify the number of bins to use for the X role and the Y role respectively. The system determines the bin size and bin start values, if not specified.

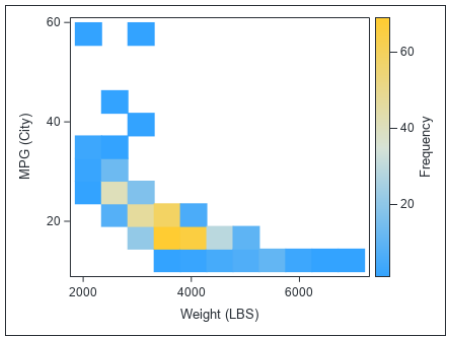
Note: The NXBINS= and NYBINS= options are available only in the SGPLOT procedure. In the SGPANEL procedure, you can manage bins using the options to control bin size and bin start values.

In the following example, the number of bins is reduced to 11 on both axes.

The following occur when you run this example:

- fewer bins are used to represent the data.
- because the bins always span the range of the data, the size of each bin increases. If you also specify the XBINSIZE, YBINSIZE, or both options, those options are ignored in order to accommodate the data.

Table 10.23 *Specified Number of Bins*




```
proc sgplot data=sashelp.cars;
 heatmap x=weight y=mpg_city /
   nxbins=11
   nybins=11;
run;
```

Conversely, if you increase the number of bins, more bins are available to represent the data, and the size of each bin decreases.

Specifying the Bin Size

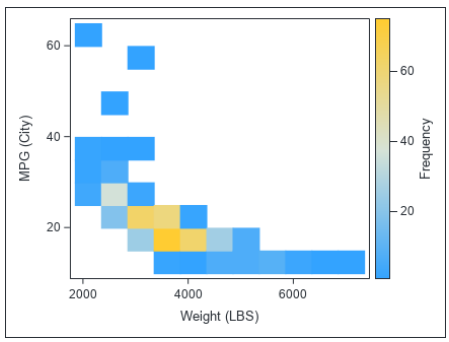
The XBINSIZE= and YBINSIZE= options specify the size of bins along the X role and the Y role in their respective data units. The system determines the number of bins and the bin start values, if not specified.

In the following example, the bin size is increased for both the X and Y roles.

The following occur when you run this example:

- the size of each bin increases.
- because the bins always span the range of the data, fewer bins are used to represent the data.

Table 10.24 *Specified Bin Size*




```
proc sgplot data=sashelp.cars;
 heatmap x=weight y=mpg_city /
   xbinsize=500
   ybinsize=5;
run;
```

Specifying the Bin Start Values

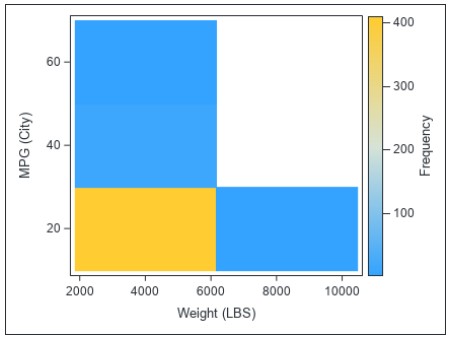
The XBINSTART= and YBINSTART= options specify the data value for the first bin of the X and Y roles, respectively. The system determines the number of bins and the bin size, if not specified.

In the following example, the bin start values are increased for both the X and Y roles.

The following occur when you run this example:

- bins are drawn only for the data that starts with the specified values. In this case, the bins represent the intersection of data values in which X >= 4000 and Y >= 20.
- because the bins always span the range of the data, the size of each bin increases and fewer bins are used to represent the data.

Table 10.25 *Specified Bin Start Values*




```
proc sgplot data=sashelp.cars;
 heatmap x=weight y=mpg_city /
   ybinstart=20
   xbinstart=4000;
run;
```

HEATMAPPARM Statement

Generates an X, Y grid of rectangles from the values of two independent variables and colors the rectangles to represent the values of a third variable, which can be a response variable or a group variable.

Requirements: The COLORGROUP= role or the COLORRESPONSE= role must be specified.

When X=, Y=, or both specify a numeric variable, the data must have at least two observations that have a distinct value for each numeric variable. Otherwise, the heat map is not drawn.

Interaction: The number of bins is limited to 100,000. If the number of bins exceeds the limit, the heat map is not drawn. A warning message is written to the SAS log in that case. The limit is specified by the ODS GRAPHICS statement NXYBINSMAX= option, which defaults to 100,000. See NXYBINSMAX= on page 115.

Note: The data for a parameterized heat map should contain only one observation for each X and Y value pair.

See: “Plot Compatibility” on page 1780 for information about overlaying this plot with other plot types

Example: “About Heat Maps” on page 21

Syntax

HEATMAPPARM X=*variable* Y=*variable*
COLORGROUP=*variable* </*options*>;

HEATMAPPARM X=*variable* Y=*variable*
COLORRESPONSE=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

FILLATTRS=(TRANSPARENCY=*number*)
: specifies the transparency of the area fill in the rectangles.

NOMISSINGCOLOR
: excludes missing values of the COLORGROUP= variable or of the COLORRESPONSE= variable from the heat map.

OUTLINE
: displays an outline around each colored region.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the rectangle outlines.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
: specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins.

YVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
: specifies whether the input Y values represent midpoints, lower endpoints, or upper endpoints of the bins.

Axis options

SHOWXBINS
: specifies that bins be used as the basis for the X-axis tick marks.

SHOWYBINS
: specifies that bins be used as the basis for the Y-axis tick marks.

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

XENDLABELS
: specifies that axis ticks and value labels are drawn at the endpoints of the bins for the X axis.

Y2AXIS
: assigns the Y variable to the secondary (right) vertical axis.

YENDLABELS
specifies that axis ticks and value labels are drawn at the endpoints of the bins for the Y axis.

Data options

DISCRETEX
forces the X axis to be discrete when the X= variable is numeric.

DISCRETEY
forces the Y axis to be discrete when the Y= variable is numeric.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Plot options

MISSING
accepts a missing value as a valid category value.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME=*"text-string"*
assigns a name to a plot statement.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Requirement If you specify a numeric variable and the X-axis type is discrete, then you must also specify DISCRETEX in the HEATMAPPARM statement. Otherwise, the heat map might not be drawn correctly.

Y=*variable*
specifies the variable for the Y axis.

Requirement If you specify a numeric variable and the Y-axis type is discrete, then you must also specify DISCRETEY in the HEATMAPPARM statement. Otherwise, the heat map might not be drawn correctly.

COLORGROUP=*variable*
specifies a variable that is used to color the regions of the heat map.

Requirement This argument is required when the response variable is of type discrete.

COLORRESPONSE=*numeric-variable*
specifies a numeric variable that is used to color the regions of the heat map.

Requirement This argument is required when the response variable is of type interval.

Tip The color ramp is specified by the COLORMODEL= option. The color ramp represents the range of unique response values.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

Note: This feature applies to the COLORGROUP variable in the heat map.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example
```
colormodel=TwoColorRamp
```

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorRamp style element

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

DISCRETEX
forces the X axis to be discrete when the X= variable is numeric.

Requirement If X= specifies a numeric variable and the X-axis type is discrete, then you must specify DISCRETEX. Otherwise, the heat map might not be drawn correctly.

Interactions If X= specifies a character variable, then this option is ignored, and the X axis is considered to be discrete.

When this option is used, the XENDLABELS= option is ignored.

DISCRETEY
forces the Y axis to be discrete when the Y= variable is numeric.

Requirement If Y= specifies a numeric variable and the Y-axis type is discrete, then you must specify DISCRETEY. Otherwise, the heat map might not be drawn correctly.

Interactions If Y= specifies a character variable, then this option is ignored, and the Y axis is considered to be discrete.

When this option is used, the YENDLABELS= option is ignored.

FILLATTRS=(TRANSPARENCY=*number*)
specifies the transparency of the area fill in the rectangles.

Default The TRANSPARENCY= option value.

Range 0 (opaque) to 1 (entirely transparent)

Interaction This option overrides the TRANSPARENCY= option for the colored region only. It does not affect the region outlines.

Note The fill colors are determined by the COLORRESPONSE= or COLORGROUP= variable.

Tip You can combine this option with TRANSPARENCY= to set one transparency for the outline but a different transparency for the fill. Example:
```
outline transparency=0.2 fillattrs=(transparency=0.6)
```

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Requirement The option enables missing values to show up along the X or Y axis provided that the axis is discrete. Either the values must be character, or you can specify the DISCRETEX or DISCRETEY option.

Interaction This option has no effect when FREQ= is specified.

NOMISSINGCOLOR
excludes missing values of the COLORGROUP= variable or of the COLORRESPONSE= variable from the heat map. If missing color values are present, observations with missing COLORGROUP= or COLORRESPONSE= values are not displayed in the heat map. However, their values are still represented on the axis.

Default Without this option, missing values are included in the heat map. The attributes of the missing value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphData*n* style element instead of by the GraphMissing style element.

OUTLINE
displays an outline around each colored region.

Default No outline is displayed

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the rectangle outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphOutlines style element in the current style.

Interaction This option has no effect unless OUTLINE is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SHOWXBINS
specifies that bins be used as the basis for the X-axis tick marks.

Default Without this option, a standard axis is used, ignoring bin boundaries and midpoints.

Restriction This option is ignored when a BY variable is used and uniform axis scaling is specified. (You specify uniform axis scaling using the UNIFORM= option in the PROC SGPLOT statement.)

Interaction When this option is used, the XENDLABELS= option determines how the axis ticks and value labels are displayed. When this option is not specified, the XENDLABELS= option is ignored.

SHOWYBINS
specifies that bins be used as the basis for the Y-axis tick marks.

Default Without this option, a standard axis is used, ignoring bin boundaries and midpoints.

Restriction This option is ignored when a BY variable is used and uniform axis scaling is specified. (You specify uniform axis scaling using the UNIFORM= option in the PROC SGPLOT statement.)

Interaction When this option is used, the YENDLABELS= option determines how the axis ticks and value labels are displayed. When this option is not specified, the YENDLABELS= option is ignored.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option. Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

XENDLABELS
specifies that axis ticks and value labels are drawn at the endpoints of the bins for the X axis.

Default If this option is not used, then the axis ticks and value labels are drawn at the bin midpoints. This is true regardless of whether the XVALUES= option identifies the X data as endpoint values or midpoint values.

Interactions This option is ignored when the X axis is discrete.

This option has no effect unless the SHOWXBINS option is also used.

XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins.

Default MIDPOINTS

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

YENDLABELS
specifies that axis ticks and value labels are drawn at the endpoints of the bins for the Y axis.

Default If this option is not used, then the axis ticks and value labels are drawn at the bin midpoints. This is true regardless of whether the YVALUES= option identifies the Y data as endpoint values or midpoint values.

Interactions This option is ignored when the Y axis is discrete.

This option has no effect unless the SHOWYBINS option is also used.

YVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the input Y values represent midpoints, lower endpoints, or upper endpoints of the bins.

Default MIDPOINTS

HIGHLOW Statement

Creates a display of floating vertical or horizontal lines or bars that represent high and low values.

Note: This plot does not summarize data. If multiple observations have the same X or Y value, the observations are all plotted separately based on their values.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Examples: "About High-Low Plots" on page 23
"Example 11: Creating a High-Low Chart" on page 1672

Syntax

HIGHLOW X=*variable* | Y=*variable*
HIGH=*numeric-variable* LOW=*numeric-variable* </*option(s)*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

CLIPCAP
: displays a clip indicator cap at the end of a bar or line when the bar or line extends beyond the axis range.

CLIPCAPSHAPE=DEFAULT | SERIF | BARBEDARROW | CLOSEDARROW | OPENARROW
: specifies the type of cap used for the clip cap.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all lines or bars from the category midpoints.

FILL

NOFILL
: specifies whether the area fill is visible for bars.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

HIGHCAP=*character-variable* | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW
: specifies the type of cap used at the high end of the bar or line.

INTERVALBARWIDTH=*numeric-value*<*units*>
: specifies the thickness of the bar when the X (or Y) data is numeric.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the outlines for the band.

LOWCAP=*character-variable* | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW
: specifies the type of cap used at the low end of the bar or line.

OUTLINE

NOOUTLINE
: specifies whether the outlines of the bars are visible.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=OVERLAY | CLUSTER
specifies how to display grouped data.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

HIGHLABEL=*variable*
specifies the label to be shown at the high end of the line or bar.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the HIGHLABEL= option, the LOWLABEL= option, or both options.

LEGENDLABEL="*text-string*"
specifies a label that identifies the elements from the band plot in the legend.

LOWLABEL=*variable*
specifies the label to be shown at the low end of the line or bar.

Plot options

CLOSE=*numeric-variable*
specifies the data for the CLOSE tick on the bar or line.

OPEN=*numeric-variable*

specifies the data for the OPEN tick on the bar or line.

TYPE=BAR | LINE
specifies how the data is to be represented.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME=*"text-string"*
assigns a name to a plot statement.

Required Arguments

X=*variable*
Y=*variable*
specifies a variable that is used to plot the values along the N or Y axis.

Note If you specify X=*variable*, then the statement creates vertical lines or bars on the X axis and the HIGH and LOW values are plotted along the Y axis. Conversely, if you specify Y=*variable*, then the statement creates horizontal lines or bars on the Y axis and the HIGH and LOW values are plotted along the X axis.

HIGH=*numeric-variable*
specifies the upper value for the floating lines or bars.

LOW=*numeric-variable*
specifies the lower value for the floating lines or bars.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of `1`, then there is no distance between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

Default 0.85

Restriction This option applies only when the X or Y axis is discrete.

Interaction This option has no effect unless TYPE=BAR.

Tip If the category axis is numeric, use the INTERVALBARWIDTH= option to specify the bar width.

CLIPCAP

displays a clip indicator cap at the end of a bar or line when the bar or line extends beyond the axis range. The cap indicates where clipping has occurred. When the MIN= and MAX= axis options are specified for an axis and a data value exceeds the specified axis range, the bar or line for that value is clipped. If the bar or line already has a high or low cap, it is replaced by the clip cap.

For vertical bars, the clip cap is added to the end of the bar that is clipped by the Y-axis range. The cap is a vertical arrowhead that points toward the clip edge (or).

For horizontal bars, the clip cap is added to the end of the bar that is clipped by the X-axis range. The cap is a horizontal arrowhead that points toward the clip edge (or).

If an entire bar or line is clipped, a clip cap is displayed at the high or low side where the bar or line was clipped.

Interactions Clip indicators appear only when CLIPCAP is specified and the data values exceed the axis range that is specified by the MIN= and MAX= options. When the MIN= and MAX= options are not specified, the axis range is adjusted to accommodate the data values, and clipping does not occur.

When TYPE=BAR, the caps are drawn to fit within the bar width. The width of the bar itself might be reduced.

Note When the high-low TYPE=LINE, you can change the appearance of the clip cap using the CLIPCAPSHAPE= option.

Tip If you specify the HIGHLABEL or LOWLABEL option, and the bar or line is clipped, the label value is still drawn outside the tip of the clip cap. If the entire bar or line is clipped, no labels are shown.

CLIPCAPSHAPE=DEFAULT | SERIF | BARBEDARROW | CLOSEDARROW | OPENARROW

specifies the type of cap used for the clip cap.

The following table shows each clip-cap shape.

Table 10.26 *Clip-Cap Shapes*

Clip-Cap Shape	TYPE=LINE	TYPE=BAR
DEFAULT		
SERIF		Not applicable.[1]

Clip-Cap Shape	TYPE=LINE	TYPE=BAR
BARBEDARROW		Not applicable.[1]
CLOSEDARROW		
OPENARROW		Not applicable.[1]

1 When specified for bars, DEFAULT is used instead.

Default DEFAULT

CLOSE=*numeric-variable*
specifies the data for the CLOSE tick on the bar or line. For a vertical plot, the tick value is represented by an indicator on the side of the bar or line that has higher X values. For a horizontal plot, the value is represented by an indicator on the side with higher Y values.

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Requirement This option is applicable only when the X or Y axis is discrete.

Interactions This option is applicable only when TYPE=BAR.

This option is applicable only when the GROUP option is specified and GROUPDISPLAY=CLUSTER.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for lines or for outline-only bars.

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.27 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

High-low plots can specify lines rather than bars.

Table 10.28 DATASKIN Options for Lines

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines or bars from the category midpoints. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset from the category midpoints)

Requirement This option is applicable only when the X or Y axis is discrete.

FILL | NOFILL
specifies whether the area fill is visible for bars. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default The default status of the area fill is specified by the DisplayOpts attribute of the GraphBand style element in the current style.

Interactions This option has no effect unless TYPE=BAR.

Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataConfidence style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

0.0 transparency

Interaction This option has no effect if you specify the NOFILL option.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interactions If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

TYPE=BAR must be specified for the fill pattern to have any effect.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

COLOR=*color*
specifies a color to use for the fill lines.

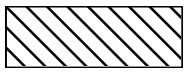
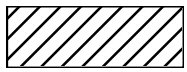
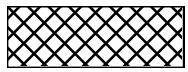
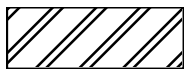
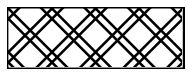
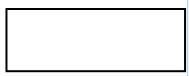
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With

grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.29 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions
For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note
This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=OVERLAY | CLUSTER
specifies how to display grouped data.

OVERLAY
groups are overlaid without any clustering.

Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
observations with different group values are displayed in adjacent clusters around the category value.

Each set of group values is centered at the midpoint tick mark for the category.

Default OVERLAY

Interactions GROUPDISPLAY=CLUSTER is applicable only when TYPE=BAR.

This option is ignored unless GROUP= is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

HIGHCAP=*character-variable* | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW

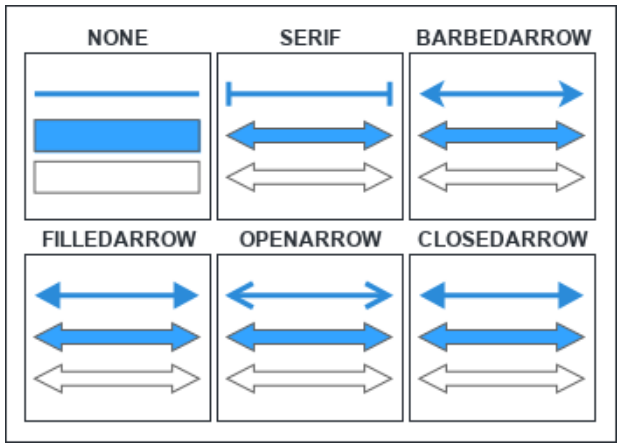
specifies the type of cap used at the high end of the bar or line. You can specify one of the keywords, or you can specify a character variable that contains one of the keywords.

All of the keywords can be specified for any high-low chart. However, the effect of each keyword depends on the setting for the TYPE= option and also the fill state of the bars, when displayed:

- When TYPE=BAR and the bars are filled, FILLEDARROW is used for all settings other than NONE.
- When TYPE=BAR and the bars are not filled, CLOSEDARROW is used for all settings other than NONE.
- When TYPE=LINE and CLOSEDARROW is specified, FILLEDARROW is used instead.

The following figure shows the effect of each cap value on horizontal lines, filled bars, and unfilled bars.

Figure 10.4 Horizontal High and Low Cap Shapes for Lines, Filled Bars, and Unfilled Bars



Default NONE

Restriction Caps are not displayed for very short bars. Bar height must be at least twice the size of the cap in order for the cap to appear.

Interaction When TYPE=BAR, the caps are drawn to fit within the bar width. The width of the bar itself might be reduced.

HIGHLABEL=*variable*
specifies the label to be shown at the high end of the line or bar.

INTERVALBARWIDTH=*numeric-value<units>*
specifies the thickness of the bar when the X (or Y) data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along X or Y.

Requirement This option is applicable only when the X or Y axis is a linear axis or a TIME axis.

Interaction This option has no effect unless TYPE=BAR.

Tip If the category axis is discrete, use the BARWIDTH= option to specify the bar width.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the HIGHLABEL= option, the LOWLABEL= option, or both options. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LEGENDLABEL="*text-string*"
specifies a label that identifies the elements from the band plot in the legend. By default, the label "band" is used for ungrouped data, and the group values are used for grouped data.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the outlines for the band. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the GraphOutlines style element in the current style.

For grouped data:

- if only outlines are displayed, LineThickness from GraphOutlines, and ContrastColor and LineStyle from a GraphData1–GraphData*n* style element in the current style
- if outlines and fill are displayed, LineThickness and LineStyle from GraphOutlines, and ContrastColor from a GraphData1–GraphData*n* style element in the current style

Interactions This option has no effect unless you also specify the OUTLINES option.

This option is ignored if the DATASKIN= option is also specified.

LOWCAP=*character-variable* | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW

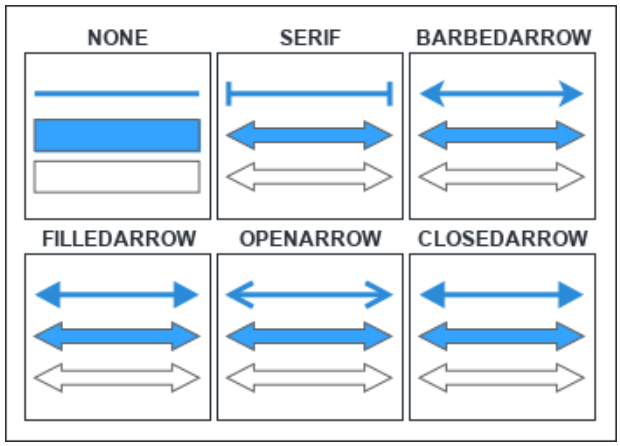
specifies the type of cap used at the low end of the bar or line. You can specify one of the keywords, or you can specify a character variable that contains one of the keywords.

All of the keywords can be specified for any high-low chart. However, the effect of each keyword depends on the setting for the TYPE= option and also the fill state of the bars, when displayed:

- When TYPE=BAR and the bars are filled, FILLEDARROW is used for all settings other than NONE.
- When TYPE=BAR and the bars are not filled, CLOSEDARROW is used for all settings other than NONE.
- When TYPE=LINE and CLOSEDARROW is specified, FILLEDARROW is used instead.

The following figure shows the effect of each cap value on horizontal lines, filled bars, and unfilled bars.

Figure 10.5 *Horizontal High and Low Cap Shapes for Lines, Filled Bars, and Unfilled Bars*



Default NONE

Restriction Caps are not displayed for very short bars. Bar height must be at least twice the size of the cap in order for the cap to appear.

Interaction When TYPE=BAR, the caps are drawn to fit within the bar width. The width of the bar itself might be reduced.

LOWLABEL=*variable*

specifies the label to be shown at the low end of the line or bar.

NAME="*text-string*"

assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OPEN=*numeric-variable*
specifies the data for the OPEN tick on the bar or line.

For a vertical plot, the tick value is represented by an indicator on the side of the bar or line that has lower X values. For a horizontal plot, the value is represented by an indicator on the side with lower Y values.

OUTLINE | NOOUTLINE
specifies whether the outlines of the bars are visible. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions This option has no effect unless TYPE=BAR.

Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, “Using Range Attribute Maps,” on page 1921

“Overview of Attribute Maps” on page 1897

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

TYPE=BAR | LINE
specifies how the data is to be represented. BAR uses fill and outline attributes. LINE uses line attributes.

Default LINE

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

HISTOGRAM Statement

Creates a histogram that displays the frequency distribution of a numeric variable.

Interactions: The HISTOGRAM statement can be combined only with DENSITY statements in the SGPLOT procedure. See "Plot Compatibility" on page 1780.

The number of bins is limited to 10,000. If the number of bins exceeds the limit, the histogram is not drawn. A warning message is written to the SAS log in that case. The limit is specified by the ODS GRAPHICS statement NBINSMAX= option, which defaults to 10,000. See NBINSMAX= on page 114.

Note: The range of the response variable is automatically divided into an appropriate number of bins.

Examples: "About Histograms" on page 51

"Example 8: Combining Histograms with Density Plots" on page 1669

Syntax

HISTOGRAM *response-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

FILL

NOFILL
: specifies whether the area fill is visible.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLENDCOLOR=*color*
: specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
: specifies the fill type that is applied to the chart.

OUTLINE

NOOUTLINE
specifies whether outlines are displayed for the bars.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

X2AXIS
assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the calculated values to the secondary (right) vertical axis.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

Label options

DATALABEL

DATALABEL=NONE | AUTO | COUNT | DENSITY | PERCENT | PROPORTION
specifies the statistic to display at the end of each bin.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

LEGENDLABEL="*text-string*"
specifies a label that identifies the histogram in the legend.

Plot options

BINSTART=*numeric-value*
specifies the X coordinate of the first bin.

BINWIDTH=*numeric-value*
specifies the bin width.

BOUNDARY=LOWER | UPPER
specifies how boundary values are assigned to bins.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

NBINS=*numeric-value*
specifies the number of bins.

SCALE=COUNT | PERCENT | PROPORTION
specifies the scaling that is applied to the vertical axis.

SHOWBINS
specifies that the midpoints of the value bins are used to create the tick marks for the horizontal axis.

WEIGHT=*numeric-variable*
specifies a variable in the input data set that contains values to be used as weights for the bin height.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Argument

response-variable
specifies the response variable for the histogram.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BINSTART=*numeric-value*
specifies the X coordinate of the first bin. Use this option in conjunction with the BINWIDTH= or NBINS= options to specify bins. If neither BINWIDTH= nor the NBINS= option is specified, the system determines the number of bins. If the BINSTART value results in excluding the entire range of data, it is ignored and the default BINSTART value is used.

Default The default value is determined by the system.

BINWIDTH=*numeric-value*
specifies the bin width. The system determines the number of bins. The bins always span the range of the data.

Default The default value is determined by the system.

Restriction The maximum number of bins is limited to approximately 10,000. If the number of bins computed from the data and the BINWIDTH= value exceeds 10,000, SAS computes a new bin-width value that yields approximately 10,000 bins. A warning of the change is written to the SAS log.

Interactions

This option is ignored if the NBINS= option is also specified.

BOUNDARY=LOWER | UPPER
specifies how boundary values are assigned to bins.

LOWER
specifies that boundary values are assigned to the lower bin.

UPPER
specifies that boundary values are assigned to the upper bin.

Default UPPER

DATALABEL
DATALABEL=NONE | AUTO | COUNT | DENSITY | PERCENT | PROPORTION
specifies the statistic to display at the end of each bin. You can specify one of the following:

NONE
suppresses the data labels.

AUTO
uses the SCALE= option value. By default, SCALE=PERCENT.

COUNT
DENSITY
PERCENT
PROPORTION
specifies that the count, density, percentage, or proportion statistic is to be displayed at the end of each bar.

Default If DATALABEL is specified without a value, then the default is AUTO.

Interaction When DATALABEL=AUTO, the SCALE= option determines the statistic that is displayed at the end of each bar.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bins. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work

best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.30 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

FILL | NOFILL

specifies whether the area fill is visible. The FILL option shows the area fill. The NOFILL option hides the area fill.

Default The default status of the area fill is specified by the DisplayOpts attribute of the GraphHistogram style element in the current style.

Interactions Specifying FILL also hides any visible outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)

specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults: For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions: This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default: The GraphWall style element Color attribute.

Interactions: This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction: If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips: Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

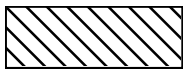
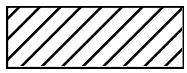
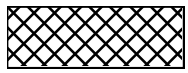
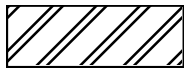
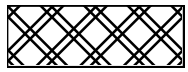
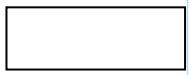
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.31 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bin is filled with the color that is assigned to the bin fill area.

GRADIENT
displays the color from the applied style at the end of the bin gradually changing to the color of the wall at the baseline of the bin.

This option does not affect transparency. Only the color is affected.

Interactions: To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bin is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bin to fully transparent at the baseline.

Interaction: Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips: If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bins.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default: SOLID

Interaction: This option has no effect if NOFILL is also specified.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction: If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

LEGENDLABEL="*text-string*"
specifies a label that identifies the histogram in the legend. By default, the label of the response variable is used.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NBINS=*numeric-value*
specifies the number of bins. The system determines the BINWIDTH= value. The bins always span the range of the data.

The procedure attempts to produce tick values that are easily interpreted (for example, 5, 10, 15, 20). The procedure sometimes adjusts the location of the first bin and the bin width accordingly. As a result, the number of bins shown in the plot might not exactly match the number specified with NBINS=.

Default The default number of bins is determined by the system.

Range 2 to 10,000

Note The limit on the number of bins is specified by the ODS GRAPHICS statement NBINSMAX= option, which defaults to 10,000. See "NBINSMAX=*n*" on page 114.

OUTLINE | NOOUTLINE
specifies whether outlines are displayed for the bars. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default The default status of the outlines is specified by the DisplayOpts attribute of the GraphHistogram style element in the current style.

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

SCALE=COUNT | PERCENT | PROPORTION
specifies the scaling that is applied to the vertical axis. Specify one of the following values:

COUNT
the axis displays the frequency count.

PERCENT
the axis displays values as a percentage of the total.

PROPORTION
the axis displays values as proportions (0.0 to 1.0) of the total.

Default PERCENT

SHOWBINS
specifies that the midpoints of the value bins are used to create the tick marks for the horizontal axis. By default, the tick marks are created at regular intervals based on the minimum and maximum values.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies a variable in the input data set that contains values to be used as weights for the bin height.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

X2AXIS
assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the calculated values to the secondary (right) vertical axis.

HLINE Statement

Creates a horizontal line chart. You can use the HLINE statement with the HBAR statement to create a horizontal bar-line chart.

Restrictions: The HLINE statement can be combined only with other categorization plot statements in the SGPLOT procedure. See "Plot Compatibility" on page 1780. and "Plot Axes" on page 1784.

When category plots are overlaid, all of the plots must use the same category variable.

See: "Marker Fills and Outlines" on page 1805

Example: "About Line Charts" on page 61

Syntax

HLINE *category-variable* </option(s)>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BREAK
: breaks the line whenever the computed statistic for a category value is missing.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all lines from the discrete category values.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the lines in the line plot.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

X2AXIS
: assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS
: assigns the category variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=DATA | LEFT | RIGHT
specifies the location of the data label.

LEGENDLABEL="*text-string*"
specifies the label that identifies the line plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

STATLABEL

NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels.

Limit options

LIMITATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITS=BOTH | LOWER | UPPER
: adds limit lines to the plot.

LIMITSTAT=CLM | STDDEV | STDERR
: specifies the statistic for the limit lines.

NOLIMITCAPS
: suppresses the serif cap on limit lines, if limit lines are displayed.

NUMSTD=*n*
: specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR.

Marker options

FILLEDOUTLINEDMARKERS
: specifies that markers have a fill and an outline.

MARKERATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
: specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the marker outlines.

MARKERS
: adds markers to the plot.

Plot options

ALPHA=*numeric-value*
: specifies the confidence level for the confidence limits.

CATEGORYORDER=RESPASC | RESPDESC
: specifies the order in which the categories are arranged.

FREQ=*numeric-variable*
: specifies a variable for the frequency count for each observation in the input data.

MISSING
: accepts a missing value as a valid category value.

RESPONSE=*response-variable*
: specifies a numeric response variable for the plot.

URL=*character-variable*
: specifies an HTML page to be displayed when parts of the plot are selected.

WEIGHT=*numeric-variable*
: specifies a variable that contains values to be used as weights for the calculations.

Plot reference options

NAME="*text-string*"
: assigns a name to a plot statement.

Statistics options

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
: specifies the statistic for the horizontal axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the lines.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default .05

Interactions This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA= value that you specify is used for all of the plots.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BREAK
breaks the line whenever the computed statistic for a category value is missing. For example, if a response variable is used and it has all missing values for a certain category value, the SUM or MEAN for this category value is missing. By default in such cases, the response value for the previous category is joined to the response value for the next category value by a line segment. If BREAK is specified, this segment is not drawn.

Interaction This option is ignored when the Y axis is discrete and the MISSING option is specified. To break the line at missing values in that case, remove the MISSING option.

Note The observation is excluded from the graph when there is a missing value for the FREQ variable.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged. By default, the plot is sorted in ascending order based on the category values. You can change the order so that it is based on the response values.

Specify one of the following values:

RESPASC
sorts by the response values in ascending order.

RESPDESC
sorts by the response values in descending order.

Default The plot is sorted in ascending order based on the category values.

Restrictions This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Uniform scaling and response sorting cannot occur on the same axis. If the UNIFORM= option is used in the SGPLOT statement, the UNIFORM option is ignored for the sorted response axis and a note is generated in the log. The UNIFORM= option is applied to the other axes and groups if requested. Note that the UNIFORM= option can selectively apply scaling to only the X or Y axis

Interactions When a group variable is used with the CATEGORYORDER= option, the category order is not affected by the value of the groups. The categories are always sorted by the response statistic at a category level.

When this option and the GROUPORDER= option are both specified, the GROUPORDER= option is ignored.

Notes If CATEGORYORDER= is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

When two or more observations have the same response values, there is no guarantee that the tied values will be sub-sorted.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interaction This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME=

option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default INSIDE

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label. Specify one of the following values:

AUTO
places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction This value takes effect only when CURVELABELLOC=OUTSIDE.

END
places the curve label at the last point on the curve.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

MIN
places the label at the part of the curve closest to the minimum X-axis value.

START
places the curve label at the first point on the curve.

Defaults END if CURVELABELLOC=INSIDE

AUTO if CURVELABELLOC=OUTSIDE

Interactions This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL
DATALABEL=*variable*
displays a label for each data point.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data

labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITS or LIMITSTAT, then the limit labels are also shown. Limit labels are shown when you use DATALABEL (without a variable).

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, for the label font, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style. For the label color, when DATALABELPOS=DATA, the ContrastColor attribute of a GraphData1–GraphData*n* style element. Otherwise, the Color attribute of the GraphDataText style element.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=DATA | LEFT | RIGHT
specifies the location of the data label. Specify one of the following values:

DATA
places the label on the data primitives (at the right edge of the lines).

LEFT
places the label to the left of the lines on the left edge of the data area.

RIGHT
places the label to the right of the lines on the right edge of the data area.

Default DATA

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays limit information when limits are specified.

This option displays group values for each category when GROUP= is also specified.

This option displays response values for each overlaid chart.

When you specify a value of LEFT or RIGHT for this option, the splitting of data labels (using the SPLITCHAR= option) is not supported.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. The data skin affects all line marker symbols, if they are displayed. If the marker symbol is not filled, then the data skin is applied to the outlines. Specify one of the following:

Table 10.32 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

If you specify markers with the plot, then the data skin affects the markers as well.

Table 10.33 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from the discrete category values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the category axis is discrete.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interaction This option has no effect unless MARKERS is also specified.

Tip Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interactions If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

When FREQ= and MISSING are both specified, the MISSING option has no effect.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map.

For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interaction This option is ignored unless GROUP= is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default ASCENDING

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the line plot in the legend. By default, the label of the response variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) are used. If you do not specify a response variable, then the legend label is "Frequency".

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

For grouped data, the LineStyle and LineThickness attributes of the GraphError style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITS= or LIMITSTAT= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITS=BOTH | LOWER | UPPER
adds limit lines to the plot. Specify one of the following values:

BOTH
adds lower and upper limit lines to the plot.

LOWER
adds lower limit lines to the plot.

UPPER
adds upper limit lines to the plot.

By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interaction Limit lines are displayed only when you specify STAT=MEAN.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines. Specify one of the following statistics:

CLM
confidence limits

STDDEV
standard deviation

STDERR
standard error

Default If you specify the LIMITS= option, then the default for LIMITSTAT= is CLM. Otherwise, there is no default.

Interaction If you specify the LIMITSTAT= option, then the default value for the LIMITS= option is BOTH.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the lines in the line plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style

element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interactions This option has no effect unless you also specify the MARKERS option.

Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

Suboption COLOR= in option LINEATTRS= overrides the default marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKERS
adds markers to the plot.

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOLIMITCAPS
suppresses the serif cap on limit lines, if limit lines are displayed.

Interaction The appearance of the limit lines is controlled by the LIMITATTRS= option.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR. You can specify any positive number, including decimals.

Default 1

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed for each value on the horizontal axis.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the horizontal axis. Specify one of the following:

FREQ
the frequencies of the response or category variable.

The frequencies are calculated as follows:

- If you specify the RESPONSE= option, FREQ calculates the frequency of the response variable.
- If you do not specify the RESPONSE= option, FREQ calculates the frequency of the category variable.

MEAN
the mean of the response variable.

Interaction For STAT=MEAN to take effect, you must also specify the RESPONSE= option.

MEDIAN
the median of the response variable.

Interaction For STAT=MEDIAN to take effect, you must also specify the RESPONSE= option.

PERCENT
the percentage of the response or category variable.

The percentage is calculated as follows:

- If you specify the RESPONSE= option, PERCENT calculates the percentage of the sum of the response variable.
- If you do not specify the RESPONSE= option, PERCENT calculates the percentage of the frequency of the category variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the =PCTLEVEL= option in the PROC SGPLOT statement.

You can use the PCTNDEC= option in the SGPLOT procedure statement to control the number of decimals to be used when calculating the percent values.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level are zero.

SUM
the sum of the response variable.

This is the default value when you specify the RESPONSE= option.

Interaction For this value to take effect, you must also specify the RESPONSE= option.

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Restriction If you do not also specify the RESPONSE= option, then only the FREQ or PERCENT statistic is calculated (FREQ is the default). If you specify RESPONSE=, then you can use any of the statistics.

Interaction When the graph is generated, the statistic is appended to the variable name in the axis label and the legend (if it is created). However, if a label has been assigned to the variable, then the label appears in the axis label and legend instead of the statistic.

STATLABEL | NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels. STATLABEL forces the statistic to be displayed. NOSTATLABEL removes the statistic from the axis and legend labels.

Normally, the procedure displays the statistic along with the name of the response variable. However, when a custom label is assigned to the response variable, the procedure does not display the statistic. In each case, you can control whether the statistic is displayed.

Defaults The statistic is displayed for the response variable.

When a custom label is assigned to the response variable, the statistic is not displayed.

Interactions This option has no effect unless the RESPONSE= option is specified.

This option has no effect if you specify the axis label using the LABEL= option in an AXIS statement.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Notes When TIP= is not specified, the category, RESPONSE, and GROUP variables are displayed, if applicable, when you hover over a line or

marker. The category and upper limit or lower limit variables are displayed when you hover over an error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a line or marker. No variables are displayed when you hover over an error bar or error bar cap.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

If your categorical chart is overlaid with other categorical charts, each chart can specify its own URL variable. The URL variable specified for each chart overlay is honored. This capability affects the following charts: DOT, HBAR, HLINE, VBAR, VLINE.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction If your plot is overlaid with other categorization plots that also specify WEIGHT=, then the first WEIGHT variable that you specified is used for all of the plots.

X2AXIS
assigns the response variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the category variable to the secondary (right) vertical axis.

INSET Statement

Adds a text box inside the axes of the plot.

Example: "About Text Insets" on page 38

Syntax

INSET *"text-string"* <*... "text-string-n"*> <*/options*>;

INSET (*label-list*) <*/options*>;

Summary of Optional Arguments

Appearance options

BACKCOLOR=*color*
specifies the background color of the inset.

BORDER

NOBORDER
specifies whether to display a border around the text box.

LABELALIGN=LEFT | CENTER | RIGHT
specifies how the labels are aligned when you specify label-value pairs.

OPAQUE
forces the inset background to be opaque rather than transparent.

POSITION=*position-value*
specifies the position of the text box within the plot.

TEXTATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the text in the text box.

TITLE="*text-string*"
specifies a title for the text box.

TITLEATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the title.

VALUEALIGN=LEFT | CENTER | RIGHT
specifies how text values are aligned.

Required Argument

***text-string-list* | (*label-list*)**
specifies the text. You must specify one of the following arguments:

text-string-list
specifies one or more quoted text strings.

Each string is placed on a separate line in the text box (for example, **"My line 1" "My line 2"**).

> **TIP** You can use the VALUEALIGN= option to align the text values. They can be centered in the graph, or right- or left-justified.

(*label-list*)
specifies a series of quoted labels and values for the text box.

Specify your label-value pairs as follows:

("*label-1*" ="*value-1*" ... "*label-n*" = "*value-n*")

Here is an example: **("My label 1" = "My value 1" "My label 2" = "My value 2")**.

> **TIP** You can use the LABELALIGN= and VALUEALIGN= options to align the labels and the values, respectively.

You can format the text in insets and add special characters using the following syntax:

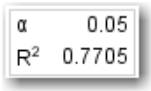
(*ESC*) {sup '*string*'}	Specifies that the string appears as a superscript.
(*ESC*) {sub '*string*'}	Specifies that the string appears as a subscript.
(*ESC*) {unicode '*hexadecimal-value*'x} (*ESC*) {keyword}	Specifies a glyph (graphical character) to be displayed using its Unicode hexadecimal value or a SAS keyword equivalent. For more information about SAS keywords, see "Reserved Keywords and Unicode Values" in *SAS Graph Template Language: Reference*.

Note: (*ESC*) is the default ODS escape character. You can set your own escape character with the `ODS ESCAPECHAR="character";` statement.

In the following inset statement, the Greek letter alpha denotes the alpha value for the confidence limits in a graph. The R-Square value is displayed using the superscript format:

```
inset ("(*ESC*){unicode alpha}" = "0.05"
         "R(*ESC*){sup '2'}" = "0.7705") / border;
```

Here is the text inset for this example:



Optional Arguments

BACKCOLOR=*color*
specifies the background color of the inset. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default By default, the background is transparent.

Interaction If BACKCOLOR= is set, then the OPAQUE option is also set automatically.

BORDER | NOBORDER
specifies whether to display a border around the text box. The BORDER option displays the border. The NOBORDER option hides the border.

LABELALIGN=LEFT | CENTER | RIGHT
specifies how the labels are aligned when you specify label-value pairs. Specify one of the following:

LEFT
aligns the text to the left.

CENTER
aligns the text to the center.

RIGHT
aligns the text to the right.

Default LEFT

OPAQUE
forces the inset background to be opaque rather than transparent. This option is useful when a transparent background makes the text in the inset difficult to see. For example, if the inset is positioned on top of a histogram, specifying OPAQUE sets the inset off from the histogram.

Default The background is transparent unless BACKCOLOR= is also specified.

Interaction If BACKCOLOR= is specified, the background is always opaque.

POSITION=*position-value*
specifies the position of the text box within the plot. The position values are as follows:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
TOP	TOPLEFT	TOPRIGHT
LEFT	RIGHT	

If you do not specify a position, then a position is determined automatically.

TEXTATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the text in the text box. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Tip The GraphDataText style element makes the text slightly smaller. The GraphLabelText style element makes the text slightly larger.

Examples
```
TEXTATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TEXTATTRS=GraphDataText
```

TITLE="*text-string*"
specifies a title for the text box. The title text is always center-aligned.

TITLEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the title. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
TITLEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TITLEATTRS=GraphTitleText
```

VALUEALIGN=LEFT | CENTER | RIGHT
specifies how text values are aligned.

Use this option to specify either of the following:

- how the values are aligned when you specify quoted label-value pairs in a *(label-list)*
- how the quoted strings are aligned when you specify a *text-string-list*

Specify one of the following values.

LEFT
aligns the text to the left.

CENTER
aligns the text to the center.

RIGHT
aligns the text to the right.

Default RIGHT

KEYLEGEND Statement

Adds a legend to the plot.

Tip: You can define a legend item that can be included in the legend. For more information, see "LEGENDITEM Statement" on page 1181.

See: "Using Discrete Legends" on page 1791

Examples: "Example 5: Adding a Prediction Ellipse to a Scatter Plot" on page 1664
"Example 6: Creating Lines and Bands from Pre-Computed Data" on page 1666
"Example 8: Combining Histograms with Density Plots" on page 1669

Syntax

KEYLEGEND <"*name-1*" ..."*name-n*"> </*options*>;

Summary of Optional Arguments

Appearance options

ACROSS=*n*
: specifies the number of columns in the legend.

BORDER

NOBORDER
: specifies whether the border around the legend is visible.

DOWN=*n*
: specifies the number of rows in the legend.

LINELENGTH=*numeric-value<units>*
: specifies the length of the line glyph for line entries in the legend.

LOCATION=OUTSIDE | INSIDE
: specifies whether the legend is placed outside or inside the axis area.

OPAQUE

NOOPAQUE
: OPAQUE removes the legend background's transparency.

OUTERPAD=*numeric-value<units>* | (*pad-options*)
: specifies the amount of extra space that is added outside the legend border.

POSITION=*position-value*
: specifies the position of the legend within the graph.

TITLE="*text-string*"
: specifies a title for the legend.

TITLEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the title.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the legend value labels.

Legend options

"name-1" ... "name-n"
: specifies the names of one or more plots that you want to include in the legend.

EXCLUDE=("*item-name*" <"*item-nameN*" ...>)
: specifies a list of legend entries to exclude from the display.

SORTORDER=ASCENDING | DESCENDING | REVERSEAUTO
: specifies the sort order to use for the legend entry labels.

TYPE=FILL | FILLCOLOR | LINE | LINECOLOR | LINEPATTERN | MARKER | MARKERCOLOR | MARKERSYMBOL
: specifies which visual attributes to display for legend entries in the legend.

Marker options

AUTOITEMSIZE
: specifies that all markers, lines, and filled symbols in the legend are sized in proportion to the font size that is used for the legend entry labels.

AUTOOUTLINE
: specifies that the outline settings in the plot statements determine whether the fill swatches in the legend have outlines.

FILLASPECT=GOLDEN | *positive-number*
: specifies an aspect ratio for the fill swatches based on their height.

FILLHEIGHT=*numeric-value<units>*
specifies the height of the fill swatches.

SCALE=*positive-number*
specifies a scale factor that is to be applied to the fill swatch height.

Optional Arguments

"name-1" ... "name-n"
specifies the names of one or more plots that you want to include in the legend. Each name that you specify must correspond to a value that you entered for the NAME= option in a plot statement.

You can specify names to control which plots are included in the legend. In addition, if you specify one or more names, the KEYLEGEND statement creates a legend when no legend is created automatically.

Note: A KEYLEGEND statement without a *"name"* reference affects only the automatic legend.

Default If you do not specify a name, then the legend contains references to all of the plots in the graph.

Note The names specified here determine which plots are included, but not the labels that appear in the legend for those plots. To specify labels, use the LEGENDLABEL= option on the plot statements.

ACROSS=*n*
specifies the number of columns in the legend. By default, the number of columns is determined automatically.

Note Depending on the number of legend entries and the number of columns and rows that you specify, the legend might not fit in your graph. If your legend does not appear, then you might need to specify a different value for the ACROSS= option.

AUTOITEMSIZE
specifies that all markers, lines, and filled symbols in the legend are sized in proportion to the font size that is used for the legend entry labels.

These proportional sizes take effect regardless of the attributes that are used in the plot. The line segments are drawn as deemed appropriate by the legend, regardless of the line thickness reported by the plot. As a result, the thickness of the line segments in the legend might differ from what is shown in the plot.

The following figures show a legend with fairly large labels. In the first figure, the color swatches are small compared to the labels. The second figure uses AUTOITEMSIZE to size the color swatches in proportion to the labels.

Default Swatch Size	AUTOITEMSIZE Used in the Statement

Tip Use the VALUEATTRS= option to control the font size for the legend value labels.

AUTOOUTLINE
specifies that the outline settings in the plot statements determine whether the fill swatches in the legend have outlines.

Default When this option is not specified, the legend fill swatches are always outlined.

Note The outlines are always 1 pixel wide with a solid pattern.

BORDER | NOBORDER
specifies whether the border around the legend is visible. The BORDER option shows the border. The NOBORDER option hides the border.

Default BORDER

DOWN=*n*
specifies the number of rows in the legend. By default, the number of rows is determined automatically.

Note: Depending on the number of legend entries and the number of columns and rows that you specify, the legend might not fit in your graph. If your legend does not appear, then you might need to specify a different value for the DOWN= option.

EXCLUDE=(*"item-name" <"item-nameN" ...>*)
specifies a list of legend entries to exclude from the display.

Default No items are excluded.

Requirement Each item name must be enclosed in quotation marks and separated from adjacent names by a space.

Note When the specified names are compared with the legend entry names, leading blanks are honored and trailing blanks are ignored.

Tip For plots with groups, you can exclude specific group values.

Example The following example excludes items Truck and Wagon from the legend.

```
exclude=("Truck" "Wagon")
```

FILLASPECT=GOLDEN | *positive-number*
specifies an aspect ratio for the fill swatches based on their height.

GOLDEN
specifies the golden ratio of 1.618 (width = 1.618 * height).

positive-number
specifies a custom aspect ratio.

Default The system determines the aspect.

Interaction This option is ignored when AUTOITEMSIZE is specified.

Tip Use FILLHEIGHT= to specify the height.

FILLHEIGHT=*numeric-value<units>*
specifies the height of the fill swatches. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The system determines the height.

Interaction This option is ignored when AUTOITEMSIZE is specified.

Tip Use FILLASPECT= to specify the aspect ratio.

LINELENGTH=*numeric-value<units>*
specifies the length of the line glyph for line entries in the legend. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The length is determined automatically by the system.

LOCATION=OUTSIDE | INSIDE
specifies whether the legend is placed outside or inside the axis area. The OUTSIDE option places the legend outside of the axis area. The INSIDE option places the legend inside the axis area.

Default OUTSIDE

Interaction If LOCATION=INSIDE and POSITION= is also specified with any value, you might need to offset the axis in order to make space for the legend within the axis area. For more information, see "Example 2: Legend with a Position Specified inside the Axis Area" on page 1180.

OPAQUE | NOOPAQUE
OPAQUE removes the legend background's transparency. NOOPAQUE forces the legend background to be transparent.

This feature is useful when the legend is positioned within the graph area, and you want to hide the graphics elements that are behind the legend.

Defaults OPAQUE when LOCATION=OUTSIDE (the default value)

NOOPAQUE when LOCATION=INSIDE

OUTERPAD=*numeric-value<units>* | (*pad-options*)
specifies the amount of extra space that is added outside the legend border.

numeric-value<units>
specifies a dimension to use for the extra space at the left, right, top, and bottom of the legend border.

The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

(*pad-options*)
a space-separated list of one or more of the following name-value pair options, enclosed in parentheses:

LEFT=*numeric-value<units>*	specifies the amount of extra space added to the left side.
RIGHT=*numeric-value<units>*	specifies the amount of extra space added to the right side.
TOP=*numeric-value<units>*	specifies the amount of extra space added to the top.
BOTTOM=*numeric-value<units>*	specifies the amount of extra space added to the bottom.

Note Sides that are not assigned padding are padded with the default amount.

Tip Use *pad-options* to create non-uniform padding.

Default No padding

Note The default units for *dimension* are pixels. If you want to specify values in other units, then you must specify the units with the value. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

POSITION=*position-value*
specifies the position of the legend within the graph. The positions are as follows:

BOTTOM
places the legend at the bottom of the graph.

BOTTOMLEFT
places the legend at the bottom left corner of the graph.

BOTTOMRIGHT
places the legend at the bottom right corner of the graph.

LEFT
places the legend at the left side of the graph.

RIGHT
places the legend at the right side of the graph.

TOP
places the legend at the top of the graph.

TOPLEFT
places the legend at the top left corner of the graph.

TOPRIGHT
places the legend at the top right corner of the graph.

Defaults BOTTOM

If LOCATION=INSIDE is specified, the legend is automatically positioned near an edge of the plot area that has the least amount of collision with the data.

Interaction If LOCATION=INSIDE and POSITION= is also specified with any value, you might need to offset the axis in order to make space for the legend within the axis area. For more information, see "Example 2: Legend with a Position Specified inside the Axis Area" on page 1180.

Notes By default, if you use more than one KEYLEGEND statement, then each legend is placed in a different position.

If you specify more than one legend with the same position, then all of your legends are placed at that position.

SCALE=*positive-number*
specifies a scale factor that is to be applied to the fill swatch height. Values greater than 1 increase the height while values less than 1 reduce the height.

Default 1

Restriction This option does not apply to markers with pattern fills.

Interaction This option is ignored when AUTOITEMSIZE is specified.

Tips Use FILLHEIGHT= to change the base height.

Use FILLASPECT= to specify the width.

SORTORDER=ASCENDING | DESCENDING | REVERSEAUTO
specifies the sort order to use for the legend entry labels. The REVERSEAUTO value reverses the order of the legend entry labels.

Defaults When the procedure contains more than one plot statement, the default is based on the order of the plot statements.

For grouped data, the default is derived from the data order. If using data that is stored in a table, the default is ASCENDING.

Interaction This option overrides the order that is set by any constituent plot statement's GROUPORDER= option.

Note Legend entries are sorted by the formatted values of the data.

TITLE="*text-string*"
specifies a title for the legend. The title is placed to the left of the legend body, except in the following cases:

- the legend contains two or more rows of items
- the legend title length exceeds the space that is available on the left side of the legend

In those cases, the title is placed above the legend body.

Defaults In an auto-generated legend, no title is displayed by default unless the legend shows group values. If the legend shows group values, the group variable label is displayed when a label is assigned, or the group variable name is displayed when a label is not assigned.

In a custom legend created using plot NAME= options and a KEYLEGEND statement, no legend title is displayed by default, even when the legend displays group values.

Requirement *text-string* must be enclosed in quotation marks.

Tips Use this option to add a title to a custom legend.

To suppress the default title in an auto-generated legend, specify TITLE="" in a KEYLEGEND statement as shown in the following example:

```
proc sgplot data=sashelp.class;
   scatter x=age y=height / group=sex;
   keylegend / title="";
run;
```

TITLEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the title. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
TITLEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TITLEATTRS=GraphTitleText
```

TYPE=FILL | FILLCOLOR | LINE | LINECOLOR | LINEPATTERN | MARKER | MARKERCOLOR | MARKERSYMBOL
specifies which visual attributes to display for legend entries in the legend.

FILL
displays the fill colors and filled-area outline colors that are used in the plots as outlined color swatches.

Filled areas include bars, bubbles, and so on.

FILLCOLOR
displays the fill colors that are used in the plots as outlined color swatches.

The swatch outline color is controlled by the style reference GraphOutlines.contrastColor in the current style. The swatch outline thickness is always 1 pixel.

LINE
displays the line colors and line patterns that are used in the plots.

LINECOLOR
displays the line colors and outline colors that are used in the plots as outlined color swatches.

The swatch outline color is controlled by the style reference GraphOutlines.contrastColor in the current style. The swatch outline thickness is always 1 pixel.

LINEPATTERN
displays the line patterns or outline patterns that are used in the plots.

The line color in the legend items is the same as the legend text color, which is controlled by the style reference GraphValueText:contrastColor in the current style.

MARKER
displays the marker symbols and marker colors that are used in the plots.

MARKERCOLOR
displays the marker colors that are used in the plots as outlined color swatches.

The swatch outline color is controlled by the style reference GraphOutlines.contrastColor in the current style. The swatch outline thickness is always 1 pixel.

MARKERSYMBOL
displays the marker symbols that are used in the plots.

The marker color in the legend items is the same as the legend text color, which is controlled by the style reference GraphValueText:contrastColor in the current style.

Default All attributes are displayed (FILL, LINE, and MARKER).

Note If a statement contributing to the legend does not have any visual attributes that match the TYPE specified, then the legend does not display any entries from that statement.

Tip This option can be used as a filter.

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the legend value labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

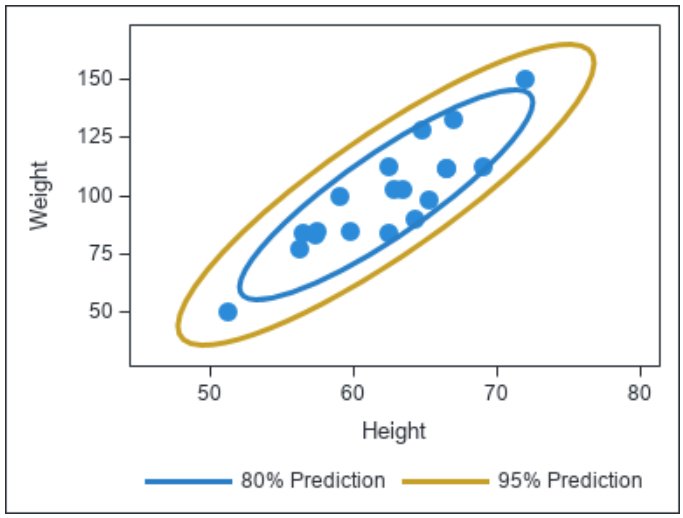
For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
valueattrs=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
valueattrs=GraphTitleText
```

Examples

Example 1: Legend for Two Ellipses

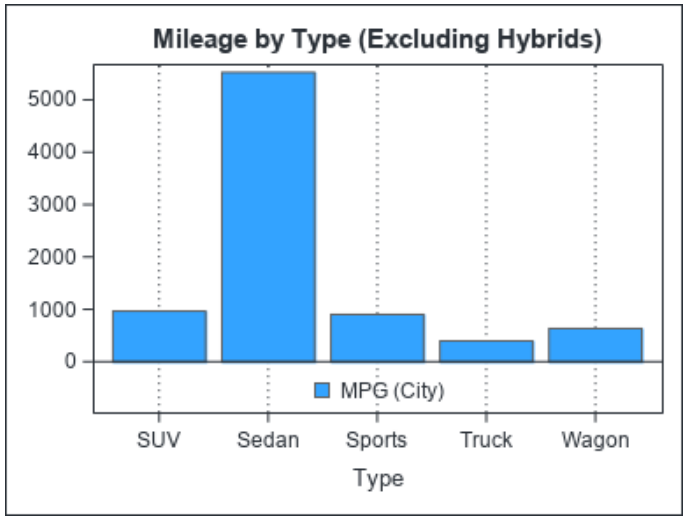



```
ods graphics / width=4in;
proc sgplot data=sashelp.class
      noautolegend;
  scatter x=height y=weight;
  ellipse x=height y=weight /
     alpha=.2
     name="eighty"
     legendlabel="80% Prediction";
  ellipse x=height y=weight /
     alpha=.05
     name="ninetyfive"
     legendlabel="95% Prediction";
  keylegend "eighty" "ninetyfive";
run;
```

Example 2: Legend with a Position Specified inside the Axis Area

If POSITION= on page 1175INSIDE and LOCATION= on page 1174INSIDE are both specified in the KEYLEGEND statement, you might need to offset the axis in order to make space for the legend within the axis area.

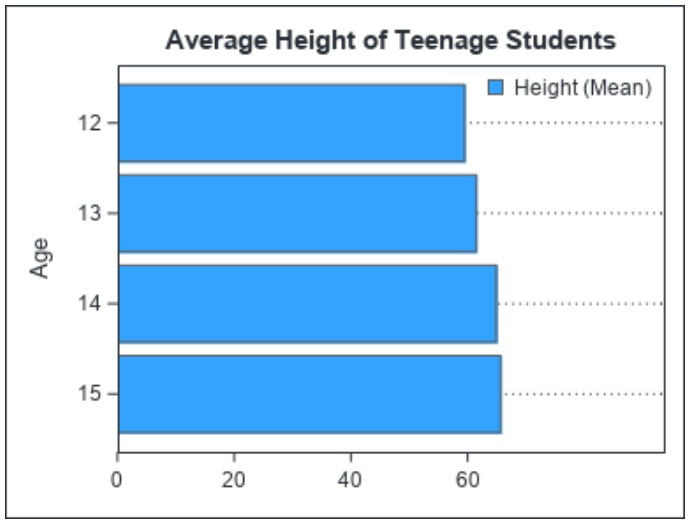
The following example specifies LOCATION=INSIDE and POSITION=BOTTOM. In the graph, the legend collides with the data. To avoid this problem, the graph specifies an OFFSETMIN= on page 1624 value for the Y axis. (If the legend's position was at the top of the plot, an OFFSETMAX= on page 1624 value would be specified instead.)




```
ods graphics / width=4in;
title "Mileage by Type (Excluding Hybrids)";
proc sgplot data=sashelp.cars;
  where type ne "Hybrid";
  vbar type / response=mpg_city;
  keylegend / location=inside position=bottom;
  xaxis grid;
  yaxis offsetmin=0.15 display=(nolabel);
run;
title;
```

Note: For HBOX on page 1077, HBAR on page 1008, HLINE on page 1145, and DOT on page 954 plot statements, the Y axis is reversed by default, so the axis origin is at the top. As a result, you use OFFSETMAX= to create an offset at the bottom of the axis. OFFSETMIN= is used to create an offset at the top of the axis for these plot types.

The following example specifies LOCATION= on page 1174INSIDE and POSITION= on page 1175TOPRIGHT. To avoid having the legend collide with the graph, the code specifies an OFFSETMAX= on page 1624 value for the X axis. The OFFSETMAX= value is applied to the end of the axis farther from the origin.




```
ods graphics / width=4in;
title "Average Height of Teenage Students";
proc sgplot data=sashelp.class;
  where age >= 12 and age < 16;
  hbar age / response=height stat=mean;
  keylegend / location=inside
    position=topright;
  xaxis offsetmax=0.3 display=(nolabel);
  yaxis grid;
run;
title;
```

LEGENDITEM Statement

Creates a legend item that can be included in the legend.

Requirement: The LEGENDITEM statement must be used with one or more plot statements.

Notes: You can use multiple LEGENDITEM statements.

When used with multiple plot statements, the order of the LEGENDITEM statement among the plot statements determines the legend item's position in the automatic legend.

Any of the supported attribute options can be specified for a legend item regardless of its type. However, attribute options that do not apply to the specified TYPE= value are ignored. For example, the MARKERATTRS= option sets the attributes for a marker and is useful if you set TYPE=MARKER or TYPE=MARKERLINE. However, if TYPE= is set to a value that does not display a marker symbol, then the MARKERATTRS= option is ignored. Similarly, if you specify a marker legend item that uses a filled marker type, then specifying NOFILL has no effect.

See: "KEYLEGEND Statement" on page 1170

Syntax

LEGENDITEM TYPE=*type* NAME="*text-string*" </*options*>;

Summary of Optional Arguments

Appearance options

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency when TYPE= is set to FILL.

FILLEDOUTLINEDMARKERS
specifies that a filled marker also has an outline.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the line when TYPE= is set to LINE or MARKERLINE.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker when TYPE= is set to MARKER or MARKERLINE.

NOFILL

NOOUTLINE
hides the fill or outline of the fill swatch for the legend item

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the outline when TYPE= is set to FILL.

TEXT="*text-string*"
specifies the text to display when TYPE= is set to TEXT.

TEXTATTRS=*style-element*< (*text-options*)> | (*text-options*)
specifies the color and font properties of the legend item when TYPE is set to TEXT.

Label options

LABEL="*text-string*"
specifies a label to be used for the item in the legend.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the color and font attributes of the legend item's label.

Required Arguments

TYPE=FILL | MARKER | MARKERLINE | LINE | TEXT
specifies the type of legend item.

FILL
specifies a fill (displayed as a filled square).

The appearance can be managed with the FILLATTRS= and OUTLINEATTRS= options.

MARKER
specifies a marker.

The appearance can be managed with the MARKERATTRS= option.

MARKERLINE
specifies a marker and a line.

The appearance can be managed with the MARKERATTRS= and LINEATTRS= options.

LINE
specifies a line.

The appearance can be managed with the LINEATTRS= option.

TEXT
specifies text that can be displayed in the legend area.

The text string is defined with the TEXT= option, and the text appearance can be managed with the TEXTATTRS= option.

NAME="*text-string*"
assigns a name to the legend item for reference in a KEYLEGEND statement.

Note The *text-string* is case-sensitive and must define a unique name within the procedure. The *text-string* must be enclosed in quotation marks.

Optional Arguments

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency when TYPE= is set to FILL. You can specify the color by using a style element or by specifying a specific option. If you specify a style element, you can also specify *options* to override the color.

Note: TRANSPARENCY cannot be derived from the style element, but it can be set in the *options*.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect if you specify the NOFILL option.

FILLEDOUTLINEDMARKERS
specifies that a filled marker also has an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored.

Interactions This option has no effect unless TYPE= is set to MARKER or MARKERLINE.

Use the FILLATTRS= and OUTLINEATTRS= options to specify attributes for the fill and outline.

Note When this option is specified, the default fill color is the color that is in effect for the FILLATTRS= option. The default line color is the color that is in effect for the OUTLINEATTRS= option. The default transparency is the transparency in effect for the MARKERATTRS= option (the transparency in the FILLATTRS= option is not honored).

LABEL="*text-string*"
specifies a label to be used for the item in the legend.

Default No label is displayed

Tip You can specify the font and color attributes for the label with the LABELATTRS= option.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the color and font attributes of the legend item's label. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Interaction For this option to take effect, the LABEL= option must also be specified.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the line when TYPE= is set to LINE or MARKERLINE. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphDataDefault style element in the current style.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker when TYPE= is set to MARKER or MARKERLINE. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default The ContrastColor, MarkerSize, and MarkerSymbol attributes of the GraphDataDefault style element in the current style.

NOFILL | NOOUTLINE
hides the fill or outline of the fill swatch for the legend item

Default The fill and outline are shown.

Interactions If NOFILL and NOOUTLINE are both specified, then both options are ignored.

If you specify a marker legend item that uses a filled marker type, then specifying NOFILL has no effect.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the outline when TYPE= is set to FILL. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the outline.

Default For ungrouped data, thte GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness.

Interaction This option has no effect if NOOUTLINE is also specified.

TEXT="*text-string*"
specifies the text to display when TYPE= is set to TEXT.

Default If this option is not specified in a legend item with TYPE=TEXT, then a default blank space is used.

Tip You can specify the appearance of the text with the TEXTATTRS= option.

TEXTATTRS=*style-element*< (*text-options*)> | (*text-options*)
specifies the color and font properties of the legend item when TYPE is set to TEXT. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

Details

The LEGENDITEM statement creates a definition for a legend item that can be included in a legend. The item that you define is independent of the data and enables you to customize the legend to enhance or replace a standard legend. For example, to display annotation text within the legend area, you can define a LEGENDITEM statement with TYPE=TEXT and specify the text in the TEXT= option.

The LEGENDITEM statement must be used with one or more plot statements within the procedure. The legend item is included in the automatically generated legend. The legend contains the legend item along with entries for the other plot statements. Note the following behaviors:

- When used with multiple plot statements, the order of the LEGENDITEM statement among the plot statements determines the legend item's position in the automatic legend.
- If a plot statement includes a group, then only the group plot shows up in the automatic legend by default.

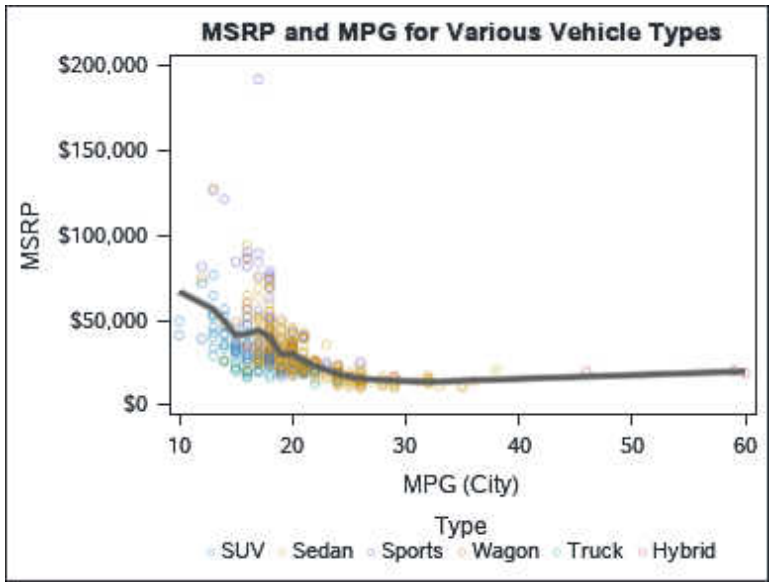
  You can override the automatic legend functionality with the KEYLEGEND statement. A single KEYLEGEND statement can reference one or more legend item names the same way it references plot names. Each legend item name that you specify must correspond to a value that you entered for the NAME= option in the LEGENDITEM statement.

  See also "KEYLEGEND Statement" on page 1170.

Example

This example scatter plot shows the city miles per gallon (MPG) and the manufacturer's suggested retail price (MSRP) for a group of vehicles. A Loess fit plot is also included. Without the LEGENDITEM statement, the auto-generated legend shows each of the six group values for vehicle type. The legend does not include the Loess plot.

Figure 10.6 Graph without the LEGENDITEM or KEYLEGEND Statements



In this example, the goal is to present the overall trend of vehicle prices and MPG rather than focus on individual vehicle types. There is no need for the detailed legend shown in the previous figure.

The example shows how to customize the legend to do the following:

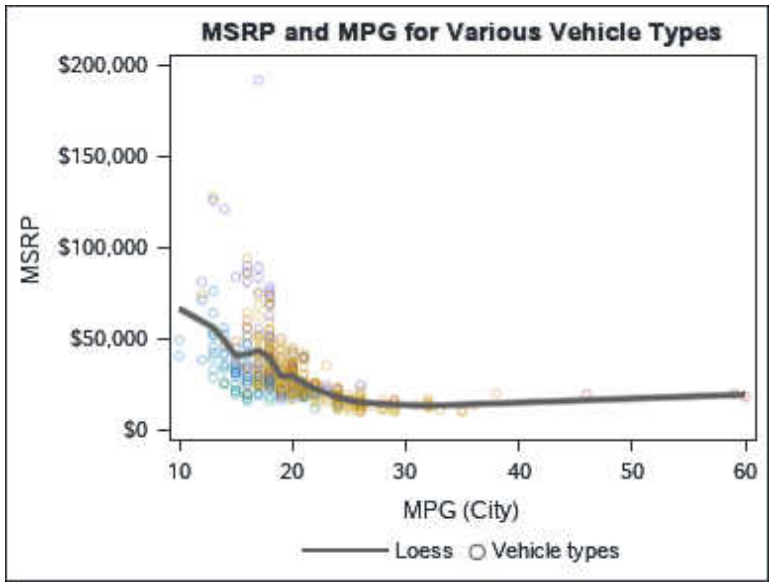
- combine all the vehicle types into a single legend entry
- include the Loess plot

In the following program, the LEGENDITEM statement is used to represent all the vehicle types within a single legend entry. The legend item is a marker that matches the scatter plot that it represents. The marker is given a neutral color so as not to be associated with any one of the vehicle types. The KEYLEGEND on page 1170 statement includes both the legend item and the Loess plot.

```
title "MSRP and MPG for Various Vehicle Types";
proc sgplot data=sashelp.cars;
   scatter x=mpg_city y=msrp / group=type
      markerattrs=(symbol=circle size=4pt);
   loess x=mpg_city y=msrp / nomarkers name="fit";
   legenditem type=marker name="car_types" /
      label="Vehicle types"
      markerattrs=(color=black symbol=circle);
   keylegend "fit" "car_types";
run;
title;
```

The result is a legend with a single entry for vehicles and an entry for the Loess plot.

Output 10.1 *Graph with the LEGENDITEM and KEYLEGEND Statements*



LINEPARM Statement

Creates a straight line specified by a point and a slope.

Requirement: The LINEPARM statement must be used with another plot statement that is derived from data values that provide boundaries for the axis area. For example, it can be used with a scatter plot or a histogram. See "Plot Compatibility" on page 1780.

Tip: You can generate a single line by specifying a constant for each required argument. You can generate multiple lines by specifying a numeric variable for any or all required arguments.

Example: "About Parameterized Lines" on page 27

Syntax

LINEPARM X=*numeric-value* | *numeric-variable*
Y=*numeric-value* | *numeric-variable*
SLOPE=*numeric-value* | *numeric-variable*
</option(s)>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

LINEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the line.

NOEXTEND
prevents the line from being extended beyond the axis offset.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS=AUTO | MIN | MAX
specifies the location of the curve label.

LEGENDLABEL="*text-string*"
specifies a label that identifies the plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text used for curve labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

CLIP
specifies that the data for the line is ignored when determining the data ranges for the axes.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

X=*numeric-value* | *numeric-variable*
specifies the X coordinate of a point.

Notes Values are in the units of the data.

If the value specified for the X= option is outside of the data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

If you specify a variable, and the variable contains any missing values, no line is drawn for the respective observation.

Y=*numeric-value* | *numeric-variable*
specifies the Y coordinate of a point.

Notes Values are in the units of the data.

If the value specified for the Y= option is outside of the data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

If you specify a variable, and the variable contains any missing values, no line is drawn for the respective observation.

SLOPE=*numeric-value* | *numeric-variable*
specifies the slope of the line. The slope can be positive or negative.

Tips SLOPE=0 creates a horizontal line (parallel to the X axis).

SLOPE= . (missing value) creates a vertical line (parallel to the Y axis).

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

CLIP
specifies that the data for the line is ignored when determining the data ranges for the axes. Each axis scale is determined by the other plots in the overlay. This might result in the line not being displayed if its data range is not within the data ranges of the other plots. This option ensures that the line is displayed.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the label from the Y variable is used as the curve label. For grouped data, the group values are used as curve labels.

Restriction This option is not valid when a constant value is specified for X=, Y=, and SLOPE= and the GROUP= option is not in effect. To label the curves in that case, use CURVELABEL="*text-string*" instead.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is valid only when a constant value is specified for X=, Y=, and SLOPE= and the GROUP= option is not in effect. If a column is specified for one or more of X=, Y=, or SLOPE, or if the GROUP= option is in effect, CURVELABEL="*text-string*" is ignored. To label the curves in that case, use CURVELABEL instead.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element

or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default INSIDE

Interactions This option is used in conjunction with the CURVELABELPOS= option to determine where the labels appear.

This option has no effect unless CURVELABEL is also specified.

CURVELABELPOS=AUTO | MIN | MAX
specifies the location of the curve label. Specify one of the following values:

AUTO
determines the best label position automatically.

The line label is positioned near the line boundary along unused axes whenever possible (typically Y2 and X2).

Interaction This value is available only when CURVELABELLOC=OUTSIDE. If CURVELABELLOC=INSIDE, you must specify either MIN or MAX.

MIN
places the curve label at the minimum value for the X axis.

MAX
places the curve label at the maximum value for the X axis.

Defaults AUTO when CURVELABELLOC=OUTSIDE

MAX when CURVELABELLOC=INSIDE

Interaction This option has no effect unless CURVELABEL is also specified.

GROUP=*variable*
specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

LEGENDLABEL="*text-string*"
specifies a label that identifies the plot in the legend. By default, the label "LineParm" is used (if you specify a numeric value for Y) or the Y variable name is used (if you specify a variable for Y).

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOEXTEND
prevents the line from being extended beyond the axis offset. When specified, there might be a gap between the line and the axis. The gap is controlled by the axis offset. If the offset is set to 0, then there is no gap.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text used for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless CURVELABEL is also specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

LOESS Statement

Creates a fitted loess curve.

Restriction: The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

See: "Marker Fills and Outlines" on page 1805
"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Loess Plots" on page 43

Syntax

LOESS X=*numeric-variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

LINEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the fit curve.

NOMARKERS
removes the scatter markers from the plot.

SMOOTH=*numeric-value*
specifies a smoothing parameter value.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
: specifies a variable that is used to group the data.

Label options

CURVELABEL
: adds a label to the curve or curves.

CURVELABEL="*text-string*"
: specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC=OUTSIDE | INSIDE
: specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS=AUTO | END | MAX | MIN | START
: specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
: displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
: specifies the location of each data label with respect to its data point.

LEGENDLABEL="*text-string*"
: specifies a label that identifies the fit line in the legend.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the text for curve labels is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

Legend options

NOLEGCLM
: hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

Limit options

CLM

CLM="*text-string*"
creates confidence limits.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY=*numeric-value*
specifies the degree of transparency for the confidence limits.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

JITTER
specifies that data markers are offset when multiple observations have the same response value.

MARKERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

Plot options

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

DEGREE=1 | 2
specifies the degree of the local polynomials to use for each local regression.

INTERPOLATION=CUBIC | LINEAR
specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

MAXPOINTS=*n*
specifies the maximum number of predicted points for the loess fit and the corresponding limits.

REWEIGHT=*n*
specifies the number of iterative reweighting steps to apply to the data.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

X=*numeric-variable*
specifies the variable for the X axis.

Y=*numeric-variable*
specifies the variable for the Y axis.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default .05

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect unless you also specify the CLM option.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CLM
CLM="*text-string*"
creates confidence limits. The optional text string overrides the default legend label for the confidence limit. The confidence level is set by the ALPHA= option.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

Note: The CLMATTRS= option has no effect unless you also specify the CLM option.

style-element
specifies the name of a style element.

CLM-attrs-suboptions
specifies one or more of the following suboptions:

CLMFILLATTRS=*style-element* <(*options*)> | (*options*)
Specifies the fill attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interactions CLMFILLATTRS= has no effect unless the fill is displayed. The fill is displayed by default. You can manually control this setting with the FILL and NOFILL suboptions.

The transparency specified with this option overrides the CLMTRANSPARENCY= option.

Tip If you want to display only the outlines, you can specify the OUTLINE and NOFILL suboptions.

Example
```
clmattrs=(clmfillattrs=(color=red transparency=.8))
```

CLMLINEATTRS=*style-element* <(*options*)> | (*options*)
Specify the line attributes of the confidence limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can specify the OUTLINE option.

Default For ungrouped data, the GraphConfidence style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interaction Specifying OUTLINE also hides the fill color, unless FILL is also specified.

Example
```
clmattrs=(outline clmlineattrs=(color=red thickness=8))
```

FILL
shows the fill color.

Default FILL

NOFILL
hides the fill color.

Default FILL

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

NOOUTLINE
hides the outlines.

Default NOOUTLINE

Interaction If NOOUTLINE and NOFILL are both specified, then both suboptions are ignored and a warning is written to the SAS log.

OUTLINE
shows the outlines.

Default NOOUTLINE

Interaction The CLMATTRS= option has no effect unless you also specify the CLM option.

CLMTRANSPARENCY=*numeric-value*
specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default 0.0

Interaction This option takes effect only if you also specify the CLM option.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default INSIDE

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label. Specify one of the following values:

AUTO
places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction This value takes effect only when CURVELABELLOC=OUTSIDE.

END
places the curve label at the last point on the curve.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

MIN
places the label at the part of the curve closest to the minimum X-axis value.

START
places the curve label at the first point on the curve.

Defaults END if CURVELABELLOC=INSIDE

AUTO if CURVELABELLOC=OUTSIDE

Interactions This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction This option has no effect if you also specify the NOMARKERS option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DEGREE=1 | 2
specifies the degree of the local polynomials to use for each local regression. 1 specifies a linear fit and 2 specifies a quadratic fit.

Default 1

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Tip Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

GROUP=*variable*
specifies a variable that is used to group the data. A separate plot is created for each unique value of the category variable. The plot elements for each group value are automatically distinguished by different visual attributes.

Interactions If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

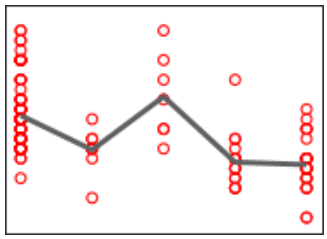
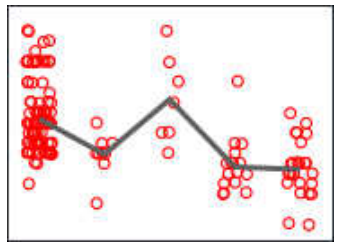
When the GROUP= option is used, the data is rendered in sorted order by the group variable.

INTERPOLATION=CUBIC | LINEAR
specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

Default LINEAR

JITTER
specifies that data markers are offset when multiple observations have the same response value. When the JITTER option is enabled, markers that represent the same response value are offset slightly in order to make all of the markers visible.

The following partial images show the effect of the JITTER option.

Default	JITTER Specified
	

Default Markers that represent the same response value are overlaid, which results in some markers being obscured.

Notes This option affects only how the scatter plot is drawn. It has no effect on the LOESS curve.

By default, the width of the jitter space is 40% of the minimal interval width of the axis. When the minimum data interval is very small, the jitter offset might not be noticeable.

LEGENDLABEL="*text-string*"
specifies a label that identifies the fit line in the legend. By default, the label "Loess" is used, along with the value of the SMOOTH= option if specified.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fit curve. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphFit style element in the current style.

For grouped data, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphFit style element in the current style.

Note If you overlay multiple Loess plots using ungrouped data, the procedure uses GraphFit and GraphFit2 for the first two plots. If three or more plots are requested, the GraphData1–GraphData*n* style elements are used instead for the ContrastColor and LineStyle

attributes. In this case, the LineThickness attribute comes from the GraphFit element.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interaction Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MAXPOINTS=*n*
specifies the maximum number of predicted points for the loess fit and the corresponding limits.

Default 201

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

NOMARKERS
removes the scatter markers from the plot.

REWEIGHT=*n*
specifies the number of iterative reweighting steps to apply to the data.

Default 0

SMOOTH=*numeric-value*
specifies a smoothing parameter value.

Default A smoothing value is determined automatically.

Range (0,1]

See What is loess regression? on blogs.sas.com.

The LOESS Procedure in *SAS/STAT User's Guide* for information about the fitting methodology that the LOESS statement uses.

SPLITCHAR="*character-list*"

specifies one or more characters on which the text for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP

specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Note The option affects only the scatter plot in this statement.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted "text strings" enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement except the confidence limits. The CLMTRANSPARENCY option must still be used to control transparency for confidence limits.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

NEEDLE Statement

Creates a plot with needles connecting each point to the baseline.

Restriction: The vertical axis that is used with the NEEDLE statement cannot be a discrete axis.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Needle Plots" on page 29

Syntax

NEEDLE X=*variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all needle lines from discrete X values.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the needle lines.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the Y-intercept for the baseline.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
: specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
: specifies how to display grouped needles.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the ordering of the groups within a category.

NOMISSINGGROUP
: specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL

DATALABEL=*variable*
: displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
: specifies the location of each data label with respect to its data point.

LEGENDLABEL="*text-string*"
: specifies a label that identifies the needle plot in the legend.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

Marker options

MARKERATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the markers in the plot.

MARKERS

adds markers to the tips of the needles.

Plot options

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*numeric-variable*
specifies the variable for the Y axis.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BASELINE=*numeric-value*
specifies the Y-intercept for the baseline. The baseline is always displayed in the plot, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the axis.

Default 0

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

For interval data, when markers are displayed and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
`DATALABELATTRS=GraphLabelText`

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. The data skin affects all line marker symbols, if they are displayed. If the marker symbol is not filled, then the data skin is applied to the outlines. Specify one of the following:

Table 10.34 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

If you specify markers with the plot, then the data skin affects the markers as well.

Table 10.35 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all needle lines from discrete X values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the X axis is discrete.

DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.

Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped needles.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interactions This option is ignored unless GROUP= is specified.

For interval data, when markers are displayed, and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced. The size could be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use

default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

LEGENDLABEL="*text-string*"
specifies a label that identifies the needle plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the needle lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interactions This option has no effect unless you also specify the MARKERS option.

Suboption COLOR= in option LINEATTRS= overrides the default marker color.

MARKERS
adds markers to the tips of the needles.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted "text strings" enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

PBSPLINE Statement

Creates a fitted penalized B-spline curve.

Restriction: The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

See: "Marker Fills and Outlines" on page 1805

"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Penalized B-Spline Plots" on page 44

Syntax

PBSPLINE X=*numeric-variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

LINEATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the fitted curve.

NKNOTS=*n*
specifies the number of evenly spaced internal knots.

SMOOTH=*numeric-value*
specifies a smoothing parameter value.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point.

SPLITCHAR="*character-list*"

specifies one or more characters on which the text for curve labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Legend options

LEGENDLABEL="*text-string*"
specifies a label that identifies the B-spline curve in the legend.

NOLEGCLI
hides the legend entry for the individual value prediction limits.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

Limit options

CLI

CLI="*text-string*"
creates confidence limits for the individual predicted values.

CLIATTRS=*style-element* | (*CLI-attrs-suboptions*)
specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying line and fill attributes.

CLM

CLM="*text-string*"
creates confidence limits.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY=*numeric-value*
specifies the degree of transparency for the confidence limits.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

JITTER
specifies that data markers are offset when multiple observations have the same response value.

MARKERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

NOMARKERS
removes the scatter markers from the plot.

PBSPLINE options

MAXPOINTS=*n*
: specifies the maximum number of predicted points for the spline curve and for any confidence limits.

Plot options

ALPHA=*numeric-value*
: specifies the confidence level for the confidence limits.

DEGREE=*non-negative-integer*
: specifies the degree of the spline transformation.

FREQ=*numeric-variable*
: specifies a variable for the frequency count for each observation in the input data.

GROUP=*variable*
: specifies a variable that is used to group the data.

WEIGHT=*numeric-variable*
: specifies a variable that contains values to be used as weights for the calculations.

Plot reference options

NAME="*text-string*"
: assigns a name to a plot statement.

Required Arguments

X=*numeric-variable*
specifies the variable for the X axis.

Y=*numeric-variable*
specifies the variable for the Y axis.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default .05

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect unless you specify the CLI or CLM option (or both).

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

CLI
CLI="*text-string*"
creates confidence limits for the individual predicted values. The optional text string overrides the default legend label for the prediction limits. The confidence level is set by the ALPHA= option.

CLIATTRS=*style-element* | (*CLI-attrs-suboptions*)
specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying line and fill attributes.

Note: The CLIATTRS= option has no effect unless you also specify the CLI option.

CLI-attrs-suboptions can be the following suboptions:

CLIFILLATTRS=*style-element*<(*options*)> | (*options*)
Specify the fill attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Default The Color and Transparency attributes of the GraphConfidence style element in the current style.

Interaction The CLIFILLATTRS= suboption has no effect unless you also specify the FILL suboption.

Example
```
cliattrs=(fill clifillattrs=(color=yellow transparency=.8))
```

CLILINEATTRS=*style-element* <(*options*)> | (*options*)
Specify the line attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphPredictionLimits style element in the current style.

Example
```
cliattrs=(clilineattrs=(color=orange pattern=longdash thickness=8px))
```

FILL
NOFILL
specifies whether the prediction band is filled.

The FILL suboption shows the fill color. The NOFILL suboption hides the fill color.

Default NOFILL

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

OUTLINE
NOOUTLINE
specifies whether the prediction band has outlines.

The OUTLINE suboption shows the outlines. The NOOUTLINE suboption hides the outlines.

Default OUTLINE

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

Default The default style element for CLIATTRS= is GraphPredictionLimits.

Interaction The CLIATTRS= option has no effect unless you also specify the CLI option.

CLM
CLM="*text-string*"
creates confidence limits. The optional text string overrides the default legend label for the confidence limit. The confidence level is set by the ALPHA= option.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

Note: The CLMATTRS= option has no effect unless you also specify the CLM option.

style-element
specifies the name of a style element.

CLM-attrs-suboptions
specifies one or more of the following suboptions:

CLMFILLATTRS=*style-element* <(*options*)> | (*options*)
Specifies the fill attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interactions CLMFILLATTRS= has no effect unless the fill is displayed. The fill is displayed by default. You can manually control this setting with the FILL and NOFILL suboptions.

The transparency specified with this option overrides the CLMTRANSPARENCY= option.

Tip If you want to display only the outlines, you can specify the OUTLINE and NOFILL suboptions.

Example
```
clmattrs=(clmfillattrs=(color=red transparency=.8))
```

CLMLINEATTRS=*style-element* <(*options*)> | (*options*)
Specify the line attributes of the confidence limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can specify the OUTLINE option.

Default For ungrouped data, the GraphConfidence style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interaction Specifying OUTLINE also hides the fill color, unless FILL is also specified.

Example
```
clmattrs=(outline clmlineattrs=(color=red thickness=8))
```

FILL
shows the fill color.

Default FILL

NOFILL
hides the fill color.

Default FILL

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

NOOUTLINE
hides the outlines.

Default NOOUTLINE

Interaction If NOOUTLINE and NOFILL are both specified, then both suboptions are ignored and a warning is written to the SAS log.

OUTLINE
shows the outlines.

Default NOOUTLINE

Interaction The CLMATTRS= option has no effect unless you also specify the CLM option.

CLMTRANSPARENCY=*numeric-value*
specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default 0.0

Interaction This option takes effect only if you also specify the CLM option.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default INSIDE

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label. Specify one of the following values:

AUTO
places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction This value takes effect only when CURVELABELLOC=OUTSIDE.

END
places the curve label at the last point on the curve.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

MIN
places the label at the part of the curve closest to the minimum X-axis value.

START
places the curve label at the first point on the curve.

Defaults END if CURVELABELLOC=INSIDE

AUTO if CURVELABELLOC=OUTSIDE

Interactions This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction This option has no effect if you also specify the NOMARKERS option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DEGREE=*non-negative-integer*
specifies the degree of the spline transformation.

Default 3

Range 0–10

Restriction DEGREE= and NKNOTS= cannot be set to 0 simultaneously. When both are set to 0, an error results.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Tip Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

GROUP=*variable*
specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

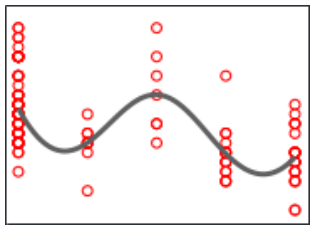
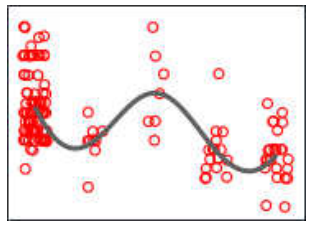
Interactions If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

When the GROUP= option is used, the data is rendered in sorted order by the group variable.

JITTER
specifies that data markers are offset when multiple observations have the same response value. When the JITTER option is enabled, markers that

represent the same response value are offset slightly in order to make all of the markers visible.

The following partial images show the effect of the JITTER option.

Default	JITTER Specified
	

Default Markers that represent the same response value are overlaid, which results in some markers being obscured.

Notes This option affects only how the scatter plot is drawn. It has no effect on the PBSPLINE curve.

By default, the width of the jitter space is 40% of the minimal interval width of the axis. When the minimum data interval is very small, the jitter offset might not be noticeable.

LEGENDLABEL="*text-string*"
specifies a label that identifies the B-spline curve in the legend. By default, the label "Penalized B-Spline" is used with the SMOOTH= value if specified, or else the group value for each B-spline is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fitted curve. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphFit style element in the current style.

For grouped data, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphFit style element in the current style.

Note If you overlay multiple PBSpline plots using ungrouped data, the procedure uses GraphFit and GraphFit2 for the first two plots. If three or more plots are requested, the GraphData1–GraphData*n* style elements are used instead for the ContrastColor and LineStyle attributes. In this case, the LineThickness attribute comes from the GraphFit element.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interaction Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style

element in the current style. The effective attributes are ContrastColor and LineThickness

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MAXPOINTS=*n*
specifies the maximum number of predicted points for the spline curve and for any confidence limits.

Default 201

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NKNOTS=*n*
specifies the number of evenly spaced internal knots. By default, a large number of knots (100) is specified, which allows for a lack of smoothness in the results. However, the final function is typically much smoother due to the penalty. When SMOOTH=0 is specified, you should typically request fewer knots than the default because there is no penalty for lack of smoothness. For example, ten or fewer knots are usually enough to follow the functional form found in most data.

Default 100

Range 0 to 1000

NOLEGCLI
hides the legend entry for the individual value prediction limits.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

NOMARKERS
removes the scatter markers from the plot.

SMOOTH=*numeric-value*
specifies a smoothing parameter value.

Default A smoothing value that minimizes a lack-of-smoothness penalty is determined automatically.

Range [0,1]

Tip Specify SMOOTH=0 for an ordinary B-spline fit.

See The TRANSREG Procedure in *SAS/STAT User's Guide* for information about the fitting methodology that the PBSPLINE statement uses.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Note The option affects only the scatter plot in this statement.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement except the confidence limits. The CLMTRANSPARENCY option must still be used to control transparency for confidence limits.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction When the CLI= option is used with this option, the confidence band for individual predicted values is displayed as a high-low plot instead of a band.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

POLYGON Statement

Draws one or more polygons from data stored in a data set.

Restrictions: You cannot overlay a polygon with a bar chart using the HBAR or VBAR statements. If you need to overlay a polygon plot on a bar chart, use an HBARPARM or VBARPARM statement instead. See "Plot Compatibility" on page 1780.

The POLYGON statement does not support data that is stored in a table. Only SAS data sets are supported for polygons.

Note: This statement can be used with other basic plot and fit plot statements.

Syntax

POLYGON X=*x-variable* Y=*y-variable* ID=*id-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

BACKLIGHT

BACKLIGHT=*number*
specifies that label text should have a back light of a contrasting color.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

FILL

NOFILL

specifies whether the polygon(s) are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)

specifies the fill color and transparency.

FILLPATTERN

enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)

specifies the line-fill color and pattern.

LINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the polygon outline.

OUTLINE

NOOUTLINE

specifies whether the polygon(s) have outlines.

RATTRID=*character-value*

specifies the value of the ID variable in a range attribute map data set.

ROTATE=*numeric-column* | *number*

specifies the angle of rotation for the polygon measured in degrees.

TRANSPARENCY=*value*

specifies the degree of transparency for the plot.

Axis options

X2AXIS

assigns the response variable to the secondary (top) horizontal axis.

XOFFSET=*numeric-value* | *numeric-variable*

specifies an amount to offset all polygon segment starting and ending points from discrete X values.

Y2AXIS

assigns the category variable to the secondary (right) vertical axis.

YOFFSET=*numeric-value* | *numeric-variable*

specifies an amount to offset all polygon segment starting and ending points from discrete Y values.

Data tip options

TIP=(*variable-list*) | NONE

specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)

applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*

creates a separate polygon color or outline pattern for each unique grouping that is specified.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

LABEL

LABEL=*variable*
specifies a label for the polygon curve.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot.

LABELLOC=INSIDEBBOX | OUTSIDEBBOX | OUTSIDE
specifies the location of the polygon label.

LABELPOS=CENTER | XMIN | XMAX | YMIN | YMAX
specifies the position of the polygon's label.

LEGENDLABEL="*text-string*"
specifies the label that identifies the polygon in the legend.

ROTATELABEL=AUTO | NONE | VERTICAL
specifies the rotation of the polygon label with respect to the rotation of the polygon.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text used for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

X=*x-variable*
specifies the variable for the X values.

Y=*y-variable*
specifies the variable for the Y values.

ID=*id-variable*
specifies the variable for the ID values that identify the polygon(s). All data rows for a single polygon must have the same ID value. Only unformatted values in the ID= column are used.

Requirement When multiple polygons are defined in the same data set, all rows with the same ID value must be in contiguous rows. The ID values cannot be interspersed.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BACKLIGHT
BACKLIGHT=*number*
specifies that label text should have a back light of a contrasting color. When you specify the BACKLIGHT option, you can also specify a number. *number* specifies the degree of the back-light effect.

The following figures show the effect of applying back light to the label. In these examples, FILL has also been specified.

BACKLIGHT=0	BACKLIGHT=1

The back light is based on text color. For dark colors, a white back-light effect is used. For lighter colors, a black back-light effect is used. The following figures show the back-light effects when full back light is applied (BACKLIGHT=1). In the first two examples, FILL has also been specified. The third example shows green text against a white background (no fill).

Black Text	Gray Text	Green Text

Defaults 0 (no back-light effect) when the BACKLIGHT option is not used.

When no number is specified for the BACKLIGHT option, the default is 0.5 when no GROUP= or COLORRESPONSE= option is used. The default is 0.75 when the GROUP= or COLORRESPONSE= option is used.

Range 0.0–1.0, where 0.0 specifies no effect and 1.0 specifies maximum effect

Note This option is most effective when text color has a low level of contrast with the background. It is also effective when the background is cluttered.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example
```
colormodel=TwoColorRamp
```

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see “Color-Naming Schemes” on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorRamp style element for filled polygons. The ThreeColorAltRamp style element for unfilled polygons.

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See “GRADLEGEND Statement” on page 1003

"Using Gradient Color Legends" on page 1793

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled polygons. Specify one of the following:

Table 10.36 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interaction If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

FILL | NOFILL
specifies whether the polygon(s) are filled. The FILL option shows the fill color for the polygons. The NOFILL option hides the fill color.

Default NOFILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color

attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

Tip The TRANSPARENCY= suboption sets the transparency for the polygon fill and the polygon outline. You can combine this option with TRANSPARENCY= to set one transparency for the outline but a different transparency for the fill. Example: `transparency=0.2 fillattrs=(transparency=0.6)`

FILLPATTERN

enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)

specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element

specifies the name of a style element.

options

specifies fill-pattern style options.

Specify one or both of the following options:

COLOR=*color*

specifies a color to use for the fill lines.

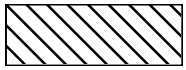
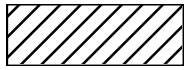
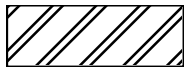
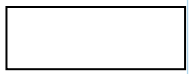
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*

specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.37 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions

For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note

This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

GROUP=*variable*
creates a separate polygon color or outline pattern for each unique grouping that is specified. The plot elements for each group value are automatically distinguished by different visual attributes.

The polygon attributes for each unique group value are derived from the GraphData1–GraphData*n* and GraphMissing style elements. If the polygon is filled, then the COLOR attribute is used for the polygon fill and the CONTRASTCOLOR attribute is used for the polygon outline. If the polygon is not filled, then the CONTRASTCOLOR and PATTERN attributes are used for the polygon outline.

Interactions When this option is used and the value is a variable associated with an ATTRID= option, the attribute mapping defined by the associated attribute map is used.

This option is ignored if the COLORRESPONSE= option is also used.

LABEL
LABEL=*variable*
specifies a label for the polygon curve.

Default When no *variable* is provided, the ID variable is used for labels.

Interactions If the plot uses a broken linear or time axis, the polygon label might not be displayed properly in some cases.

If a *variable* is supplied, rows that have the same ID value are expected to have the same value. If they do not, the ID value's first row determines the polygon label.

Tip The font and color attributes for the label are specified by the LABELATTRS= option.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the LABEL= option is also specified.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphValueText
```

LABELLOC=INSIDEBBOX | OUTSIDEBBOX | OUTSIDE
specifies the location of the polygon label.

INSIDEBBOX
inside the polygon.

OUTSIDEBBOX
outside the polygon but inside the plot area.

OUTSIDE
outside the plot area.

Default INSIDEBBOX

Interactions This option has no effect unless the LABEL= option is also specified.

The label's exact position is relative to the polygon's X and Y data ranges and is determined by the combination of this option and the LABELPOS= option.

LABELPOS=CENTER | XMIN | XMAX | YMIN | YMAX
specifies the position of the polygon's label. Specify one of the following values:

CENTER
centers the label.

Interaction LABELPOS=CENTER has no effect if you also specify LABELLOC=OUTSIDE or LABELLOC=OUTSIDEBBOX. The default label position is used in those cases.

XMIN
places the label at the part of the polygon closest to the minimum X-axis value, centered in the Y-axis range.

XMAX
places the label at the part of the polygon closest to the maximum X-axis value, centered in the Y-axis range.

YMIN
places the label at the part of the polygon closest to the minimum Y-axis value, centered in the X-axis range.

YMAX
places the label at the part of the polygon closest to the maximum Y-axis value, centered in the X-axis range.

Default CENTER

Interactions This option has no effect unless the LABEL= option is also specified.

The label's exact position is relative to the polygon's X and Y data ranges and is determined by the combination of this option and the LABELLOC= option.

When LABELLOC=OUTSIDE, increasing the length of the label might cause the available plot area to decrease. Also, when

LABELLOC=OUTSIDE, the label might collide with the axis ticks and tick values.

LEGENDLABEL="*text-string*"
specifies the label that identifies the polygon in the legend.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the polygon outline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the GraphOutlines style element in the current style.

For grouped data, the LineThickness attribute of the GraphOutlines style element, and the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element in the current style.

Interactions This option is ignored if the NOOUTLINE option is also specified.

This option is ignored if the DATASKIN= option is also specified.

Note If this option is used with grouped data, the COLOR= suboption has the effect of holding the polygon outline color constant across all group values.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OUTLINE | NOOUTLINE
specifies whether the polygon(s) have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interaction If NOOUTLINE and NOFILL are both specified, then both options are ignored.

Tip Use the LINEATTRS= option to control the appearance of the outline.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

ROTATE=*numeric-column* | *number*
specifies the angle of rotation for the polygon measured in degrees. Positive angles are measured in the counterclockwise direction, and negative angles are measured in the clockwise direction.

Default 0 (no rotation is performed)

Interaction When this option is specified, the LABELLOC= and LABELPOS= settings are ignored and they are automatically set to INSIDEBBOX and CENTER respectively.

Notes An angle that exceeds 360 degrees in absolute value can be specified.

Rotating a polygon might cause clipping in some cases.

ROTATELABEL=AUTO | NONE | VERTICAL
specifies the rotation of the polygon label with respect to the rotation of the polygon. Specify one of the following:

AUTO
rotates the label with the rotation of the polygon.

NONE
does not rotate the label with the rotation of the polygon.

The label position remains fixed regardless of the polygon rotation.

VERTICAL
rotates the label to a vertical position.

Default AUTO

Interaction If ROTATE= is also specified, then ROTATELABEL=VERTICAL is ignored.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text used for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

Default A single space

Interactions This option has no effect unless LABEL= is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the response variable to the secondary (top) horizontal axis.

XOFFSET=*numeric-value* | *numeric-variable*
specifies an amount to offset all polygon segment starting and ending points from discrete X values.

Default 0.0 (no offset)

Range –0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between discrete ticks.

Requirement The X axis must be discrete.

Y2AXIS
assigns the category variable to the secondary (right) vertical axis.

YOFFSET=*numeric-value* | *numeric-variable*
specifies an amount to offset all polygon segment starting and ending points from discrete Y values.

Default 0.0 (no offset)

Range –0.5 (downward) to +0.5 (upward), where 0.5 represents half the distance between discrete ticks.

Requirement The Y axis must be discrete.

Details

Overview

The POLYGON statement draws a polygon from a series of X-Y value pairs that are stored in a SAS data set. The first X-Y value pair defines the starting point of the polygon. When the next X-Y pair is encountered in the data set, a line is drawn from the starting point to the second X-Y point. For subsequent X-Y pairs, a line is drawn from the previous X-Y point the current X-Y point. This pattern repeats until all of the segments have been drawn

Using the POLYGON statement, you can draw any data-driven shape on your graph, which enables you to highlight data features, outline data boundaries, and so on. If you later want to modify your polygon, you need only modify the polygon data in the graph data set.

You can specify a “hole” in a polygon. A data row with missing X and Y values indicates that the rows that follow specify the X and Y points for the hole, which is inside the outer polygon.

Here is the sequence:

1. The initial data rows specify the outer polygon.
2. A row with missing X and Y values signals the beginning of a hole.
3. Subsequent data rows specify the X and Y points for the hole.

Multiple holes can be specified. Each hole is specified by a data row with missing X and Y values, followed by a series of rows that specify the X and Y points for the hole.

Requirements for the Polygon Data Set

In the simplest case of a single polygon, your data set must provide an X, Y, and ID column that stores the X-Y values and the ID for your polygon. The X-Y values in the first data set observation must specify the starting point of your polygon. The X-Y values in the subsequent observations must provide the points of each segment in the order in which the polygon is to be drawn. There should be no gaps in the data. In order to close the polygon, you can specify the starting X-Y values in your last observation, although this is not required. If your last observation does not close the polygon, the POLYGON statement automatically draws a segment from your last point back to the starting point in order to close the polygon.

If you want to draw multiple polygons, your ID column must specify a unique identifier string for each polygon. The identifier string associates the observations in the data set with a specific polygon. All of the observations for each individual polygon must be grouped together by ID and must be arranged in the order in which the polygon segments are to be drawn.

Options are available that enable you to customize the polygon and enhance its appearance. For example, you can do the following:

- show or hide the fill and outline, and specify line and fill attributes. You can also apply a data skin as well as transparency to the polygon.
- rotate the polygon.
- offset the polygon from the X or Y axis.
- specify a label, the locations of the label, and label attributes. You can also rotate the label and specify how it fits in the allotted space when there is not enough room to display the text normally.
- specify an HTML page to display when the selectable polygon is clicked.

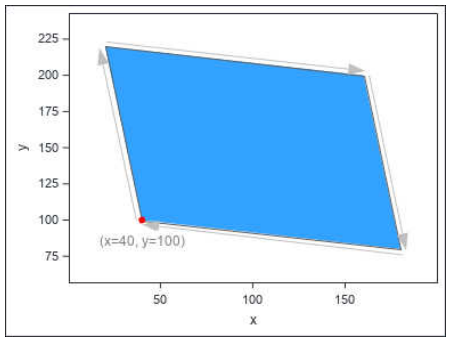
Drawing a Single Polygon

For a single polygon, the POLYGON data set contains an X and Y column that defines the polygon points, and an ID column. The polygon segments are drawn in the order in which they occur in the data. If the polygon overlaps any graphics elements that were drawn earlier, those elements are obscured. In that case, you can use transparency to enable the underlying graphics elements to show through.

Here is a listing of example data for a simple four-sided polygon identified as P1 that starts at point X=40, Y=100.

Obs	id	x	y
1	P1	40	100
2	P1	20	220
3	P1	160	200
4	P1	180	80
5	P1	40	100

The following figure shows how the polygon is drawn.



As shown in the figure, the polygon starting point is X=40, Y=100. The first segment (S1) is drawn from the starting point to X=20, Y=220. The second segment is drawn from X=20, Y=220 to X=160, Y=200. The remaining two segments (S3 and S4) follow the same pattern. Although provided in this example, the last observation (X=40, Y=100) is not required. If not provided in the data, the POLYGON statement draws the last segment (S4) automatically in order to close the polygon.

Drawing Multiple Polygons

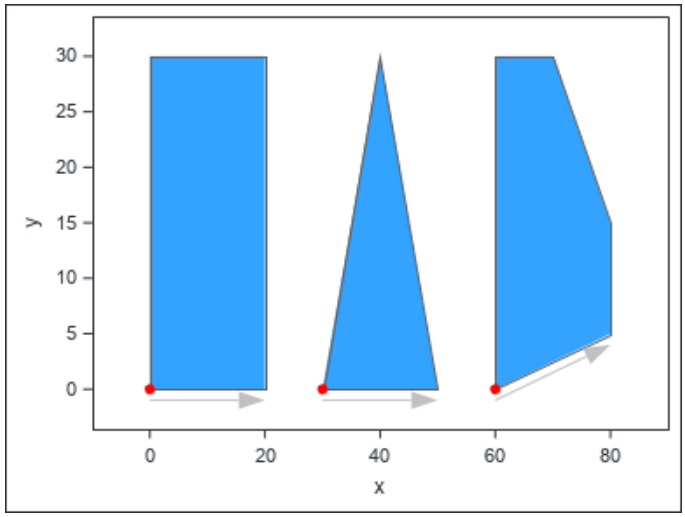
For multiple polygons, the POLYGON data ID column specifies a unique identifier for all of the observations that are associated with each polygon. The X and Y columns specify the polygon points. The polygons are overlaid on the graph in the order in which they occur in the data. For overlapping polygons, each polygon obscures part or all of the polygons and graphics elements that were drawn before it. In that case, you can use transparency to enable the underlying polygons and graphics elements to show through.

Here is a listing of example data for three separate polygons.

Obs	id	x	y
1	1	0	0
2	1	20	0
3	1	20	30
4	1	0	30
5	1	0	0
6	2	30	0
7	2	50	0
8	2	40	30
9	2	30	0
10	3	60	0
11	3	80	5
12	3	80	15
13	3	70	30
14	3	60	30
15	3	60	0

In addition to the ID, X, and Y columns, the LABEL column is added to label the polygons in the output. Notice that the observations for each ID value are grouped together in the data set. The observations for each ID must occur contiguously in the data. Otherwise, unexpected results might occur.

The following figure shows how the polygons are drawn from this data. Reference lines are provided to help you locate the polygon points in the output.



The polygons are drawn in the order in which they appear in the data: rectangle (ID=1), triangle (ID=2), and polygon (ID=3). The red dot on each shape indicates the starting point for that shape, and the gray arrows and dots indicate the subsequent points and drawing progression for each shape.

Drawing Holes in a Polygon

Using the POLYGONPLOT statement, you can draw one or more holes inside a polygon. To create data for a polygon with one or more holes:

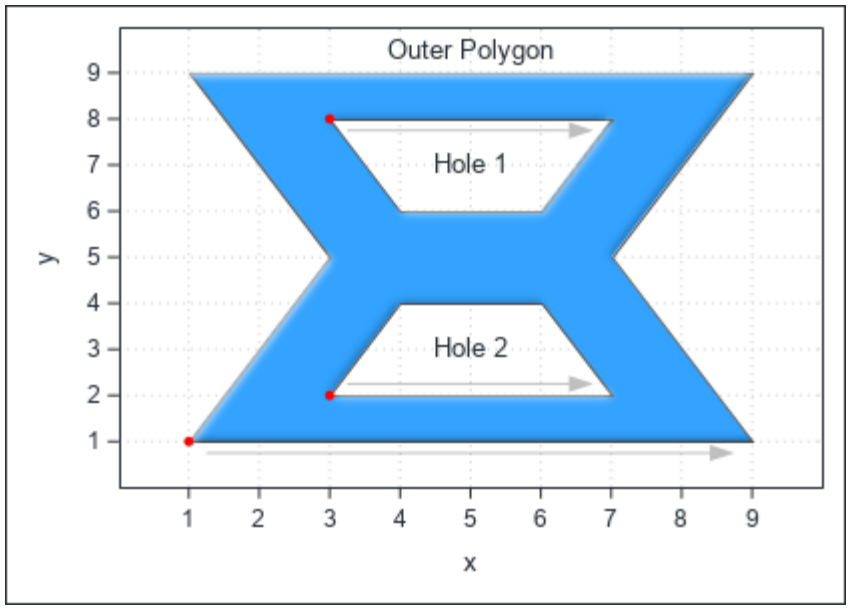
1. Specify the X and Y values for the outer polygon.
2. To start the data for a hole, add an observation that has missing X and Y values. The missing X and Y values signal the POLYGONPLOT statement that the observations that follow define the data for a hole.
3. Specify the X and Y values for the hole polygon.
4. Repeat Steps 2 and 3 for each additional hole.

Here is a listing of example data for a simple polygon that has two holes.

Obs	id	x	y
1	1	1	1
2	1	9	1
3	1	7	5
4	1	9	9
5	1	1	9
6	1	3	5
7	1	1	1
8	1	.	.
9	1	3	8
10	1	7	8
11	1	6	6
12	1	4	6
13	1	3	8
14	1	.	.
15	1	3	2
16	1	7	2
17	1	6	4
18	1	4	4
19	1	3	2

Observations 1–7 specify the data for the outer polygon. In observation 8, the X and Y values are missing, which indicates the start of the data for the first hole. Observations 9–13 define the data for the first hole polygon. Observation 14 indicates the start of the data for the second hole, which is defined by observations 15–19.

The following figure shows how the polygon is drawn from this data. Grid lines are provided to help you locate the polygon vertices in the output.

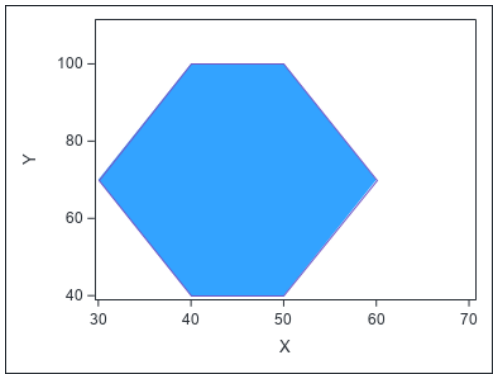


The outer polygon is drawn first, starting at point X=1, Y=1. The segments are drawn in data order, which is in a counterclockwise direction as indicated by the gray arrow. The first hole is drawn next, starting at point X=3, Y=8. Its segments are drawn in a clockwise direction. The second hole is drawn last, starting at point X=3, Y=2. Its segments are drawn in a counterclockwise direction.

Examples

Example 1: Create a Single Filled Hexagon

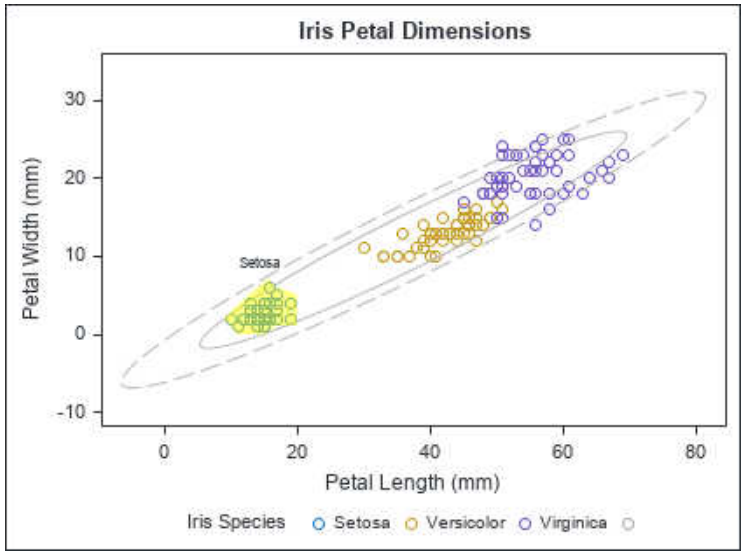
This example shows creates a single filled hexagon.




```
ods graphics / width=3in;

data polydata;
   input id $1-3 X Y;
datalines;
Tri 40  40
Tri 30  70
Tri 40  100
Tri 50  100
Tri 60  70
Tri 50  40
; run;


proc sgplot data=polydata;
polygon X=x Y=y id=id / fill outline
  lineattrs=GraphData3 ;
  xaxis min=30 max=70;
  yaxis min=40 max=110;
run;
```

Example 2: Use a Polygon to Highlight Data in a Scatter Plot

This example shows you how to use the POLYGON statement to draw a filled polygon that highlights data in an iris petal dimension scatter plot. The polygon surrounds the markers for the Setosa species in order to highlight the data with a yellow highlight for that species.




```
/* Generate the data for the polygon */
data polydata;
   input polyID polyX polyY label $8-40;
datalines;
1  9 2    Setosa
1 13 5
1 16 7
1 17 6
1 20 5
1 20 1
1 17 1
1 15 0
1 14 0
1 11 0
;
run;

/* Concatenate the SASHELP.IRIS and polygon data into
   data set IRIS */
data iris;
   set sashelp.iris polydata;
run;

/* Create the graph  */
title "Iris Petal Dimensions";
proc sgplot data=iris;
  scatter x=petallength y=petalwidth / name="s" group=species
     markerattrs=(symbol=circle size=6pt);
  ellipse x=petallength y=petalwidth / type=predicted  alpha=.2
          name="p80" legendlabel="80%"
          lineattrs=graphconfidence(thickness=1px);
  ellipse x=petallength y=petalwidth / type=predicted  alpha=.05
          name="p95" legendlabel="95%"
          lineattrs=graphconfidence2(thickness=1px);
  polygon x=polyX y=polyY id=polyID / fill
           fillattrs=(color=yellow transparency=0.5)
           label=label labelloc=outsidebbox labelpos=ymax;
run;
title;
```

To draw a single polygon, the data set must provide an X, Y, and ID column. The data specifies the polygon points around the Setosa data in the clockwise direction. A LABEL column is added to provide a label for the polygon in the plot output.

In the POLYGON statement, the FILLATTRS= on page 1380 option specifies the fill color as yellow and a fill transparency of 0.75. The LABEL= on page 1246 option specifies the column in the data set that contains the polygon label text. The LABELLOC= on page 1246 and LABELPOS= on page 1247 options place the polygon label outside of and above the polygon's bounding box.

To draw the polygon, the POLYGON statement starts at X=9, Y=2, and draws a segment between each point in data order. The last point, X=11, Y=0, does not close the polygon. To close the polygon, the POLYGON statement draws a segment between X=11, Y=0 and X=9, Y=2 automatically.

If you want to highlight the Versicolor data instead of the Setosa data, you need only modify the data in the POLYDATA data set to draw a polygon around the Versicolor data instead. You do not have to make any changes to the SGPLOT procedure.

See Also

Area Bar Charts using SGPLOT

REFLINE Statement

Creates a horizontal or vertical reference line.

Requirement: The REFLINE statement must be overlaid with another plot that is derived from data values that provide boundaries for the axis area.

Example: "About Reference Lines" on page 25

Syntax

REFLINE *variable* | *value-1*< *...value-n*> </*options*>;

Summary of Optional Arguments

Appearance options

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all lines from discrete X or Y values.

DISCRETETHICKNESS=*numeric-value*
: specifies that the line thickness should be interpreted as a percentage of the mid-point spacing.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the reference line.

NOCLIP
: extends the plot axes to contain the reference lines.

TRANSPARENCY=*value*
: specifies the degree of transparency for the reference line.

Axis options

AXIS=X | X2 | Y | Y2
: specifies the axis that contains the reference line values.

Label options

LABEL
: creates labels for each reference line from the reference line values.

LABEL=*variable-name* | *"text-string"* | (*"text-string-1"* ... *"text-string-n"*)
: creates labels for each reference line from values stored in a variable or from one or more specified label values.

LABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels.

LABELLOC=INSIDE | OUTSIDE
: specifies whether the labels are placed inside or outside of the plot area.

LABELPOS=AUTO | MIN | MAX
: specifies the position of the labels.

LEGENDLABEL=*"text-string"*
: specifies a label that identifies the plot in the legend.

SPLITCHAR=*"character-list"*
: specifies one or more characters on which the text for labels is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

Plot reference options

NAME=*"text-string"*
: assigns a name to a plot statement.

Required Arguments

variable

draws a reference line for each value of the specified variable.

Requirements
: Values must be the same type as the data type of the axis. For example, you should use numeric SAS date or time values (or SAS date/time constants) for a time axis.

Unformatted numeric values do not map to a formatted discrete axis. If reference lines are drawn on a formatted discrete axis, the specified variable must use the same format that is used for the axis.

***value-1* <*... value-n*>**
draws one or more reference lines at the values that you specify.

Requirements Values must be the same type as the data type of the axis. For example, you should use numeric SAS date or time values (or SAS date/time constants) for a time axis.

Unformatted numeric values do not map to a formatted discrete axis. If reference lines are drawn on a formatted discrete axis, the specified values must be the formatted values as they appear on the axis.

Optional Arguments

AXIS=X | X2 | Y | Y2
specifies the axis that contains the reference line values. For example, if you specify AXIS= X, vertical reference lines are drawn at points on the X axis.

Default Y

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 10.38 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

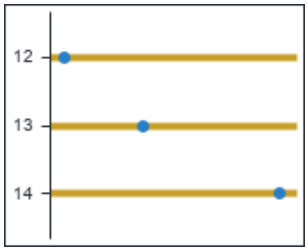
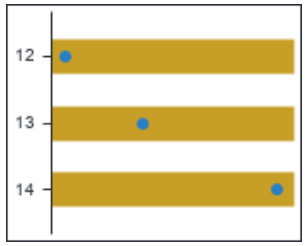
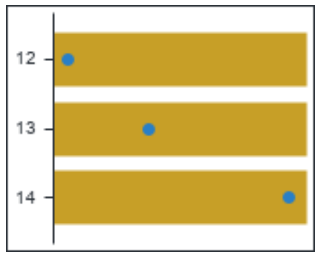
DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from discrete X or Y values.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset)

Requirement This option is applicable only when the X or Y axis is discrete.

DISCRETETHICKNESS=*numeric-value*
specifies that the line thickness should be interpreted as a percentage of the mid-point spacing. The following partial images show horizontal reference lines with varying values for DISCRETETHICKNESS=:

DISCRETETHICKNESS=0.1 (Occupies one tenth of the mid-point spacing)	DISCRETETHICKNESS=0.5 (Occupies one half of the mid-point spacing)	DISCRETETHICKNESS=0.75 (Occupies 75% of the mid-point spacing)
		

Default 0.01

Range 0–1

Requirement This option is applicable only when the REFLINE axis is discrete.

Interaction When DISCRETETHICKNESS= is used, specifying the LINEATTRS= thickness option has no effect.

Note Specifying a value of 0 effectively hides the line.

Tip This feature enables you to display category color bands.

LABEL
creates labels for each reference line from the reference line values.

Interaction If the plot uses a broken linear or time axis, the reference line label might not align properly with the reference line in some cases.

LABEL=*variable-name* | *"text-string"* | (*"text-string-1" ... "text-string-n"*)
creates labels for each reference line from values stored in a variable or from one or more specified label values.

variable-name
specifies the name of a variable that contains the reference line labels.

Restriction The specified label variable is used only when a variable is used for the REFLINE value. If this condition fails, the label variable is ignored and a message is written to the log.

"text-string"
specifies a text string for the label value.

Restriction The label string does not apply when a variable is used for the REFLINE value. In that situation, the label string is ignored and a message is written to the log.

Note If you specify multiple reference values and only one label, the specified label appears on each reference line.

"text-string-1" ... "text-string-n"
specifies a text string for each label value when multiple reference values are specified.

Restriction The label string does not apply when a variable is used for the REFLINE value. In that situation, the label strings are ignored and a message is written to the log.

Note The labels correspond to the reference values in the order specified. When too few labels are specified, some reference lines are not labeled. Excess labels are ignored.

Interaction If the plot uses a broken linear or time axis, the reference line label might not align properly with the reference line in some cases.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute from a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the LABEL option is also specified.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LABELLOC=INSIDE | OUTSIDE
specifies whether the labels are placed inside or outside of the plot area. The INSIDE option places the labels inside the plot area. The OUTSIDE option places the labels outside of the plot area.

Default OUTSIDE

LABELPOS=AUTO | MIN | MAX
specifies the position of the labels. MIN specifies the label is placed at the minimum value of the data axis. MAX specifies that the label is placed at the maximum value of the data axis. AUTO places the label automatically.

Default AUTO

LEGENDLABEL="*text-string*"
specifies a label that identifies the plot in the legend. By default, the label "Reference Line" is used.

Interaction This option has no effect unless you also specify the NAME= option.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the reference line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphReference style element in the current style.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOCLIP
extends the plot axes to contain the reference lines. By default, if a reference line is created outside of the data range, then the reference line is not visible. This option has no effect if you do not create reference lines that are outside of the data range.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless LABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TRANSPARENCY=*value*
specifies the degree of transparency for the reference line.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Note This option does not affect the reference-line label.

Details

Reference lines are always drawn perpendicular to the axes. They are drawn from one axis boundary to the companion boundary (X to X2 or Y to Y2). Axis offsets do not apply to reference lines. A plot statement that provides a sufficient data range

for determining axis extents must be included with the REFLINE statement in the SGPLOT procedure step. For example, a REFLINE statement can be used with a SCATTER or SERIES statement, but it cannot be used by itself. You can specify reference lines by using one or more constant values in the REFLINE statement, or you can specify one or more reference lines by using a data column name in the REFLINE statement. You can use the REFLINE options to add a label to each reference line and to control the appearance of the reference lines and labels.

A reference line can display a data tip in HTML output. Data tips are not enabled by default. To enable data tips, specify the IMAGEMAP option in an ODS GRAPHICS statement. For one or more reference lines that are specified by using constant values in the REFLINE statement, the data tip is formatted as X = *value* or Y = *value*, depending on the specified axis, where *value* is the axis value that was specified for that reference line. For reference lines that are specified by using a column name, the data tip is formatted as *column-name* = *value* or, if a label is assigned to the column, *column-label* = *value*, where *value* is the value in the specified column for that reference line.

REG Statement

Creates a fitted regression line or curve.

Restriction: The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

Interaction: A linear regression (DEGREE=1) cannot be used with logarithmic axes.

See: “Marker Fills and Outlines” on page 1805

“Plot Compatibility” on page 1780 for information about overlaying this plot with other plot types

Examples: “About Regression Plots” on page 45

“Example 4: Adding Prediction and Confidence Bands to a Regression Plot” on page 1663

Syntax

REG X=*numeric-variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*

specifies the value of the ID variable in a discrete attribute map data set.

LINEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the fit line.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point.

SPLITCHAR="*character-list*"

specifies one or more characters on which the text for curve labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Legend options

LEGENDLABEL="*text-string*"
specifies a label that identifies the regression curve in the legend.

NOLEGCLI
hides the legend entry for the individual value confidence limits.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

Limit options

CLI

CLI="*text-string*"
creates confidence limits for the individual predicted values.

CLIATTRS=*style-element* | (*CLI-attrs-suboptions*)
specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying line and fill attributes.

CLM

CLM="*text-string*"
creates confidence limits.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

CLMTRANSPARENCY=*numeric-value*
specifies the degree of transparency for the confidence limits.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

JITTER
specifies that data markers are offset when multiple observations have the same response value.

MARKERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

NOMARKERS
removes the scatter markers from the plot.

Plot options

ALPHA=*numeric-value*
: specifies the confidence level for the confidence limits.

DEGREE=*n*
: specifies the degree of the polynomial fit.

FREQ=*numeric-variable*
: specifies a variable for the frequency count for each observation in the input data.

MAXPOINTS=*n*
: specifies the maximum number of predicted points for the regression curve and for any confidence limits.

WEIGHT=*numeric-variable*
: specifies a variable that contains values to be used as weights for the calculations.

Plot reference options

NAME="*text-string*"
: assigns a name to a plot statement.

Required Arguments

X=*numeric-variable*
specifies the variable for the X axis.

Y=*numeric-variable*
specifies the variable for the Y axis.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default .05

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect unless you specify the CLI or CLM option (or both).

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

CLI
CLI="*text-string*"
creates confidence limits for the individual predicted values. The optional text string overrides the default legend label for the prediction limits. The confidence level is set by the ALPHA= option.

CLIATTRS=*style-element* | (*CLI-attrs-suboptions*)
specifies the appearance of the individual value prediction limits by using an ODS style element or by specifying line and fill attributes.

Note: The CLIATTRS= option has no effect unless you also specify the CLI option.

CLI-attrs-suboptions can be the following suboptions:

CLIFILLATTRS=*style-element*<(*options*)> | (*options*)
Specify the fill attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Default The Color and Transparency attributes of the GraphConfidence style element in the current style.

Interaction The CLIFILLATTRS= suboption has no effect unless you also specify the FILL suboption.

Example
```
cliattrs=(fill clifillattrs=(color=yellow transparency=.8))
```

CLILINEATTRS=*style-element* <(*options*)> | (*options*)
Specify the line attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphPredictionLimits style element in the current style.

Example
```
cliattrs=(clilineattrs=(color=orange pattern=longdash thickness=8px))
```

FILL
NOFILL
specifies whether the prediction band is filled.

The FILL suboption shows the fill color. The NOFILL suboption hides the fill color.

Default NOFILL

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

OUTLINE
NOOUTLINE
specifies whether the prediction band has outlines.

The OUTLINE suboption shows the outlines. The NOOUTLINE suboption hides the outlines.

Default OUTLINE

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

Default The default style element for CLIATTRS= is GraphPredictionLimits.

Interaction The CLIATTRS= option has no effect unless you also specify the CLI option.

CLM
CLM="*text-string*"
creates confidence limits. The optional text string overrides the default legend label for the confidence limit. The confidence level is set by the ALPHA= option.

CLMATTRS=*style-element* | (*CLM-attrs-suboptions*)
specifies the appearance of the mean value confidence limits by using an ODS style element or by specifying fill and line attributes.

Note: The CLMATTRS= option has no effect unless you also specify the CLM option.

style-element
specifies the name of a style element.

CLM-attrs-suboptions
specifies one or more of the following suboptions:

CLMFILLATTRS=*style-element* <(*options*)> | (*options*)
Specifies the fill attributes of the prediction limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interactions CLMFILLATTRS= has no effect unless the fill is displayed. The fill is displayed by default. You can manually control this setting with the FILL and NOFILL suboptions.

The transparency specified with this option overrides the CLMTRANSPARENCY= option.

Tip If you want to display only the outlines, you can specify the OUTLINE and NOFILL suboptions.

Example
```
clmattrs=(clmfillattrs=(color=red transparency=.8))
```

CLMLINEATTRS=*style-element* <(*options*)> | (*options*)
Specify the line attributes of the confidence limits.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

The confidence limits have no outlines by default (for most styles). To make the outlines appear, you can specify the OUTLINE option.

Default For ungrouped data, the GraphConfidence style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interaction Specifying OUTLINE also hides the fill color, unless FILL is also specified.

Example
```
clmattrs=(outline clmlineattrs=(color=red thickness=8))
```

FILL
shows the fill color.

Default FILL

NOFILL
hides the fill color.

Default FILL

Interaction If NOFILL and NOOUTLINE are both specified, then both suboptions are ignored and a warning is written to the SAS log.

NOOUTLINE
hides the outlines.

Default NOOUTLINE

Interaction If NOOUTLINE and NOFILL are both specified, then both suboptions are ignored and a warning is written to the SAS log.

OUTLINE
shows the outlines.

Default NOOUTLINE

Interaction The CLMATTRS= option has no effect unless you also specify the CLM option.

CLMTRANSPARENCY=*numeric-value*
specifies the degree of transparency for the confidence limits. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default 0.0

Interaction This option takes effect only if you also specify the CLM option.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default INSIDE

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label. Specify one of the following values:

AUTO
places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction This value takes effect only when CURVELABELLOC=OUTSIDE.

END
places the curve label at the last point on the curve.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

MIN
places the label at the part of the curve closest to the minimum X-axis value.

START
places the curve label at the first point on the curve.

Defaults END if CURVELABELLOC=INSIDE

AUTO if CURVELABELLOC=OUTSIDE

Interactions This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, then the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

Interaction This option has no effect if you also specify the NOMARKERS option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DEGREE=*n*
specifies the degree of the polynomial fit. For example, 1 specifies a linear fit, 2 specifies a quadratic fit, and 3 specifies a cubic fit.

Default 1

Range 1–10

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Tip Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

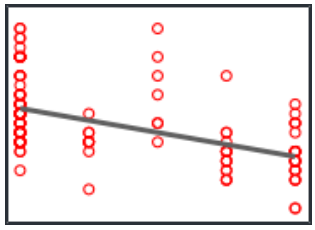
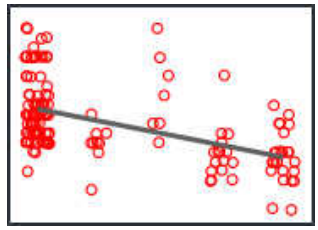
GROUP=*variable*
specifies a variable that is used to group the data. A separate plot is created for each unique value of the grouping variable. The plot elements for each group value are automatically distinguished by different visual attributes.

Interactions If you specify the GROUP= option in multiple fit plot statements, then the first GROUP= variable is used for all of the fit plots that specify GROUP=.

When the GROUP= option is used, the data is rendered in sorted order by the group variable.

JITTER
specifies that data markers are offset when multiple observations have the same response value. When the JITTER option is enabled, markers that represent the same response value are offset slightly in order to make all of the markers visible.

The following partial images show the effect of jittering.

Default	JITTER Specified
	

Default Markers that represent the same response value are overlaid, which results in some markers being obscured.

Notes This option affects only how the scatter plot is drawn. It has no effect on the REG curve.

By default, the width of the jitter space is 40% of the minimal interval width of the axis. When the minimum data interval is very small, the jitter offset might not be noticeable.

LEGENDLABEL="*text-string*"
specifies a label that identifies the regression curve in the legend. By default, the label "Regression" is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fit line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphFit style element in the current style.

For grouped data, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphFit style element in the current style.

Note If you overlay multiple REG plots using ungrouped data, the procedure uses GraphFit and GraphFit2 for the first two plots. If three or more plots are requested, the GraphData1–GraphData*n* style elements are used instead for the ContrastColor and LineStyle attributes. In this case, the LineThickness attribute comes from the GraphFit element.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you

specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interaction Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MAXPOINTS=*n*
specifies the maximum number of predicted points for the regression curve and for any confidence limits.

Default 10

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOLEGCLI
hides the legend entry for the individual value confidence limits.

NOLEGCLM
hides the legend entry for the mean value confidence limits.

NOLEGFIT
hides the legend entry for the fit line.

NOMARKERS
removes the scatter markers from the plot.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Note The option affects only the scatter plot in this statement.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement except the confidence limits. The CLMTRANSPARENCY option must still be used to control transparency for confidence limits.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction When the CLI= option is used with this option, the confidence band for individual predicted values is displayed as a high-low plot instead of a band.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

SCATTER Statement

Creates a scatter plot.

See: "Marker Fills and Outlines" on page 1805
"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Examples: "About Scatter Plots" on page 31
"Example 1: Grouping a Scatter Plot" on page 1660
"Example 5: Adding a Prediction Ellipse to a Scatter Plot" on page 1664

"Example 6: Creating Lines and Bands from Pre-Computed Data" on page 1666

Syntax

SCATTER X=*variable* Y=*variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all markers from discrete X or Y values.

ERRORBARATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the error bars in the plot.

ERRORCAPSCALE=*numeric-value*
: specifies a positive number to use as a multiplier for changing the default width of the error-bar caps.

LABELSTRIP
: strips leading and trailing blanks from marker characters or data labels with fixed positions before they are displayed in the plot.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
: assigns the Y variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped markers.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

NOERRORCAPS
suppresses the serif cap on error bars, if error bars are displayed.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point.

LEGENDLABEL="*text-string*"
specifies a label that identifies the markers from the plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

JITTER
specifies that markers that represent the same response value are offset slightly in order to make all of the markers visible.

JITTER=UNIFORM

specifies that data markers should first be jittered and then squeezed together uniformly across all category values.

JITTERWIDTH=*positive-number*
specifies the width of the jitter space as a fraction of either the midpoint spacing or of the minimal interval width.

MARKERATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERCHAR=*variable*
specifies a variable whose values replace the marker symbols in the plot.

MARKERCHARATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the markers in the plot when you use the MARKERCHAR= option.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

Plot options

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

XERRORLOWER=*numeric-variable*
specifies a variable that contains the lower endpoints for the X error bars.

XERRORUPPER=*numeric-variable*
specifies a variable that contains the upper endpoints for the X error bars.

YERRORLOWER=*numeric-variable*
specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER=*numeric-variable*
specifies a variable that contains the upper endpoints for the Y error bars.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*variable*
specifies the variable for the Y axis.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

For interval data, when GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorAltRamp style element

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the marker color, and the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element.

Interaction This option has no effect unless the DATALABEL option is also specified.

Note When the DATALABELPOS= option is specified, the data label font size might be reduced in some cases to avoid overlapping labels and markers.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all marker symbols. If the marker symbol is not filled, then the data skin is applied to the outlines. Specify one of the following:

Table 10.39 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interaction You can use the MARKERATTRS= option to specify a filled marker symbol.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all markers from discrete X or Y values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the X or Y axis is discrete.

ERRORBARATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the error bars in the plot. You can specify the appearance by using a style element or by using suboptions. If you specify a style element, you can also specify suboptions to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies one or more error-bar attribute options.

Specify one or more of the following options:

COLOR=*color*
specifies the color of the line.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Defaults For ungrouped data, the ContrastColor attribute of the GraphError style element in the current style.

For grouped data, the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

PATTERN=*line-pattern*
specifies the line pattern for the line.

You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default The LineStyle attribute of the GraphError style element in the current style.

THICKNESS=*n <units>*
specifies the thickness of the line.

You can also specify the unit of measure. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The LineThickness attribute of the GraphError style element in the current style.

ERRORCAPSCALE=*numeric-value*
specifies a positive number to use as a multiplier for changing the default width of the error-bar caps. For example, ERRORCAPSCALE=2 doubles the default cap width and ERRORCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction This option has no effect if NOERRORCAPS is specified.

Tip The thickness (height) of a cap honors the line thickness specified in the ERRORBARATTRS= option. The cap pattern is always solid.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Tip Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restriction If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

Note If the value is not an integer, only the integer portion is used.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped markers.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interactions This option is ignored unless GROUP= is specified.

For interval data, when GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

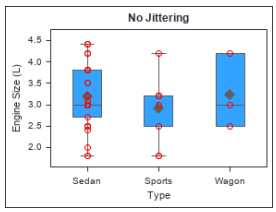
Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes

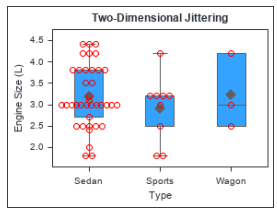
only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

JITTER

specifies that markers that represent the same response value are offset slightly in order to make all of the markers visible. The following examples show how the jittering differs based on the variable type that is being plotted on the X and Y axes.

Table 10.40 *X Axis Is Discrete and Y Axis Is Linear*





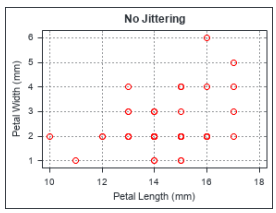
In the first case (No Jittering), markers that represent the same Y value are overlaid, which results in some markers being obscured.

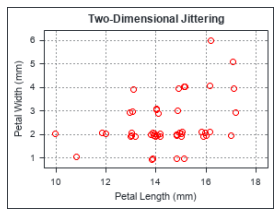
In the second figure, the JITTER option is specified in the SCATTER statement. In this case, one-dimensional systematic jittering occurs along the X axis. The markers that represent the same Y value are offset along the X axis from the midpoint of that value in order to make all of the markers visible.

Note: When the JITTER width is not wide enough to fit long strings of markers, those long strings are squeezed to fit. The other strings that do fit are not squeezed. To force the JITTER option to squeeze all the strings of markers proportionally, specify JITTER=UNIFORM.

The next figures show the case in which both axes are linear. In the first figure (No Jittering), markers that represent the same X and Y bin value are overlaid, which results in some markers being obscured. In the second figure, two-dimensional random jittering occurs along the X and Y axes. The markers are offset randomly along both the X and Y axes in order to make all of the markers visible.

Table 10.41 *Both Axes Are Linear*





Default When JITTER is not specified, markers that represent the same response value are overlaid. As a result, some markers might be obscured.

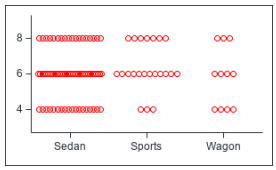
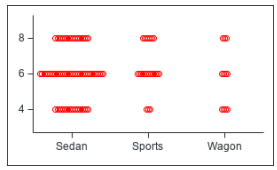
Interaction The JITTER option is not supported if MARKERCHAR= is also specified. The combination of these two options can produce unpredictable results.

Note By default, the width of the jitter space is 40% of the minimal interval width of the axis. When the minimum data interval is very small, the jitter offset might not be noticeable.

JITTER=UNIFORM
specifies that data markers should first be jittered and then squeezed together uniformly across all category values. If UNIFORM is not specified, data markers are squeezed together only for those categories whose data markers cannot fit within the jittering space.

The following partial graphs show the effect of specifying the UNIFORM argument.

Table 10.42 *JITTER versus JITTER=UNIFORM*

JITTER	JITTER=UNIFORM
	

Restriction The JITTER=UNIFORM option is available only for discrete, one-dimensional jittering.

JITTERWIDTH=*positive-number*
specifies the width of the jitter space as a fraction of either the midpoint spacing or of the minimal interval width.

Default 0.85

Notes The specified number can be greater than 1.

When used with a discrete axis, changes to JITTERWIDTH= take effect only as long as the markers overlap. Once the specified jitter width reaches the point where the markers are clustered side-by-side without overlapping, further increases to JITTERWIDTH= have no effect on the markers.

LABELSTRIP
strips leading and trailing blanks from marker characters or data labels with fixed positions before they are displayed in the plot. The MARKERCHAR=

option specifies the variable that provides the marker strings that are used in place of marker symbols.

Interactions This option affects marker strings only when the MARKERCHAR= option is specified.

This option affects data labels only when DATALABEL is specified.

Tip Stripping the blanks from the numeric value strings helps center each string relative to its data point. Stripping is useful when you want to overlay the data values near or inside the markers for a plot.

LEGENDLABEL="*text-string*"
specifies a label that identifies the markers from the plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interaction Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

MARKERCHAR=*variable*
specifies a variable whose values replace the marker symbols in the plot. If a format is associated with the variable, then the formatted values are used as the marker symbols. If there is not a format associated with the variable and the variable contains numeric data, then the BEST6. format is used.

Interaction The MARKERCHAR= option overrides the DATALABEL= option and the SYMBOL= suboption of the MARKERATTRS= option.

Tip If the MARKERCHAR= variable contains one or more long character strings, large axis offsets might result. In that case, try using the LABELSTRIP option in the SCATTER statement to reduce the offsets. If the result is unsatisfactory, use the OFFSETMIN= and OFFSETMAX= options in the axis statement to override the automatic axis offsets. See "Plot Axes" on page 1784.

MARKERCHARATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot when you use the MARKERCHAR= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute from a GraphData1–GraphData*n* style element in the current style.

Examples
```
MARKERCHARATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
MARKERCHARATTRS=GraphValueText
```

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOERRORCAPS
suppresses the serif cap on error bars, if error bars are displayed.

Interaction The appearance of the error bars is controlled by the ERRORBARATTRS= option.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Notes When TIP= is not specified, the X, Y, COLORRESPONSE, and GROUP variables are displayed, if applicable, when you hover over a marker. The Y and XERRORLOWER or XERRORUPPER variables are displayed when you hover over an X error bar or error bar cap. The X and YERRORLOWER or YERRORUPPER variables are displayed when you hover over a Y error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a marker. No variables are displayed when you hover over an error bar or error bar cap.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see “Using the ODS GRAPHICS Statement” on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

XERRORLOWER=*numeric-variable*
specifies a variable that contains the lower endpoints for the X error bars.

XERRORUPPER=*numeric-variable*
specifies a variable that contains the upper endpoints for the X error bars.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

YERRORLOWER=*numeric-variable*
specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER=*numeric-variable*
specifies a variable that contains the upper endpoints for the Y error bars.

SERIES Statement

Creates a line plot.

Restriction: The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

Note: The SERIES statement plots data in data order. For this reason, the input data set should be sorted by the X variable. If it is not sorted by the X variable, unexpected results might occur.

See: "Marker Fills and Outlines" on page 1805

"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Examples: "About Series Plots" on page 33

"Example 3: Plotting Three Series" on page 1662

Syntax

SERIES X=*variable* Y=*variable* </*options*>;

Summary of Optional Arguments

Appearance options

ARROWHEADPOS=START | END | BOTH
: specifies a position for arrowheads.

ARROWHEADSCALE=*positive-number*
: specifies an arrowhead scale factor based on the thickness of the arrow line.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
: specifies a shape for arrowheads.

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all lines from discrete X or Y values.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the series line.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

THICKMAX=*numeric-value*<*units*>
: specifies the maximum line thickness when a response variable is used to determine the line thickness.

THICKMAXRESP=*numeric-value*
: specifies the response value that corresponds to the maximum line thickness.

THICKRESP=*numeric-variable*
: specifies a response variable that is used to map a line thickness to each group value.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
: assigns the Y variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

GROUPLC=*variable*
specifies a variable that determines the line colors for a grouped plot independently of the GROUP= variable.

GROUPLP=*variable*
specifies a variable that determines the line patterns for a grouped plot independently of the GROUP= variable.

GROUPMC=*variable*
specifies a variable that determines the marker colors for a grouped plot independently of the GROUP= variable.

GROUPMS=*variable*
specifies a variable that determines the marker symbols for a grouped plot independently of the GROUP= variable.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

LCATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPLC= variable.

LPATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPLP= variable.

MCATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPMC= variable.

MSATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPMS= variable.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL
adds a label to the curve or curves.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point.

LEGENDLABEL="*text-string*"
specifies a label that identifies the series plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

MARKERATTRS=*style-element*<(**options**)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

MARKERS
adds data point markers to the series plot data points.

Plot options

BREAK
creates a break in the line for each missing value for the Y variable.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Series options

SMOOTHCONNECT
specifies that a smoothed line passes through all vertices.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*variable*
specifies the variable for the Y axis.

Optional Arguments

ARROWHEADPOS=START | END | BOTH
specifies a position for arrowheads. The arrowheads are placed at the starting and ending points using the X values in data order. In order to position the arrowheads properly, you might need to sort the data by the X column.

START displays an arrowhead at the starting point of each line.

END displays an arrowhead at the ending point of each line.

BOTH displays an arrowhead at each end of each line.

Default No arrowheads are displayed when this option is not specified

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSCALE=*positive-number*
specifies an arrowhead scale factor based on the thickness of the arrow line.

Default 1.0

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

Tip Use a factor greater than 1.0 to make a larger arrowhead.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
specifies a shape for arrowheads. The following figure shows each of the arrowhead shapes.

Default OPEN

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BREAK
creates a break in the line for each missing value for the Y variable.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

For interval data, when markers are displayed and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example
```
colormodel=TwoColorRamp
```

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorAltRamp style element

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default INSIDE

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label. Specify one of the following values:

AUTO
places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction This value takes effect only when CURVELABELLOC=OUTSIDE.

END
places the curve label at the last point on the curve.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

MIN
places the label at the part of the curve closest to the minimum X-axis value.

START
places the curve label at the first point on the curve.

Defaults END if CURVELABELLOC=INSIDE

AUTO if CURVELABELLOC=OUTSIDE

Interactions This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. The data skin affects all line marker symbols, if they are displayed. If the marker symbol is not filled, then the data skin is applied to the outlines. Specify one of the following:

Table 10.43 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

If you specify markers with the plot, then the data skin affects the markers as well.

Table 10.44 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from discrete X or Y values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the X or Y axis is discrete.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interaction This option has no effect unless MARKERS is also specified.

Tip Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interactions This option is ignored unless GROUP= is specified.

For interval data, when markers are displayed, and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced. The size could be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

GROUPLC=*variable*
specifies a variable that determines the line colors for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series line colors are selected from the GraphData1–GraphData*n* style elements based on the GROUPLC= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

This option overrides the COLOR= suboption of the LINEATTRS= option.

Tips Use the GROUPLP= option to set the line pattern.

Use the LCATTRID= option to associate the GROUPLC= variable with an attribute map ID.

You can also use LINEATTRS= option to set the line pattern and thickness.

See "Example 1: Using Group Options without an Attribute Map" on page 1323

GROUPLP=*variable*
specifies a variable that determines the line patterns for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series line patterns are selected from the GraphData1–GraphData*n* style elements based on the GROUPLP= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

This option overrides the PATTERN= suboption of the LINEATTRS= option.

Tips Use the GROUPLC= option to set the line color. You can also use the LINEATTRS= option to set the line thickness and color.

Use the LPATTRID= option to associate the GROUPLP= variable with an attribute map ID.

See "Example 1: Using Group Options without an Attribute Map" on page 1323

GROUPMC=*variable*

specifies a variable that determines the marker colors for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series marker colors are selected from the GraphData1–GraphData*n* style elements based on the GROUPMC= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

The MARKERS option must also be specified for this option to have any effect.

This option overrides the COLOR= suboption of the MARKERATTRS= option.

Tips Use the GROUPMS= option to set the marker symbol. You can also use the MARKERATTRS= option to set the marker size and symbol.

Use the MCATTRID= option to associate the GROUPMC= variable with an attribute map ID.

See "Example 1: Using Group Options without an Attribute Map" on page 1323

GROUPMS=*variable*

specifies a variable that determines the marker symbols for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series marker symbols are selected from the GraphData1–GraphData*n* style elements based on the GROUPMS= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

The MARKERS option must also be specified for this option to have any effect.

This option overrides the SYMBOL= suboption of the MARKERATTRS= option.

Tips Use the GROUPMC= option to set the marker color. You can also use the MARKERATTRS= option to set the marker size and color.

Use the MSATTRID= option to associate the GROUPMS= variable with an attribute map ID.

See "Example 1: Using Group Options without an Attribute Map" on page 1323

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

LCATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPLC= variable. This option enables the plot to associate an attribute map with the line color group. The option specifies the value of the ID variable that maps attributes to LINECOLOR in the attribute map.

Requirement The procedure statement must specify the attribute map data set using the DATTRMAP= option.

Interactions This option is ignored unless the GROUPLC= option is specified.

This option overrides the COLOR= suboption of the LINEATTRS= option.

Note Do not use this option and the ATTRID= option in the same plot statement. Doing so can cause unexpected results.

See "Example 1: Using Group Options without an Attribute Map" on page 1323

Chapter 18, "Using Discrete Attribute Maps," on page 1901

LEGENDLABEL="*text-string*"
specifies a label that identifies the series plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the series line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

LPATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPLP= variable. This option enables the plot to associate an attribute map with the line pattern group. The option specifies the value of the ID variable that maps attributes to LINEPATTERN in the attribute map.

Requirement The procedure statement must specify the attribute map data set using the DATTRMAP= option.

Interactions This option is ignored unless the GROUPLP= option is specified.

This option overrides the PATTERN= suboption of the LINEATTRS= option.

Note Do not use this option and the ATTRID= option in the same plot statement. Doing so can cause unexpected results.

See "Example 1: Using Group Options without an Attribute Map" on page 1323

Chapter 18, "Using Discrete Attribute Maps," on page 1901

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interactions This option has no effect unless you also specify the MARKERS option.

Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

Suboption COLOR= in option LINEATTRS= overrides the default marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default: For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness

Requirement: Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip: You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See: For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKERS
adds data point markers to the series plot data points.

MCATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPMC= variable. This option enables the plot to associate an attribute map with the marker color group. The option specifies the value of the ID variable that maps attributes to MARKERCOLOR in the attribute map.

Requirement: The procedure statement must specify the attribute map data set using the DATTRMAP= option.

Interactions: This option is ignored unless the GROUPMC= option is specified.

This option is ignored unless the MARKERS option is specified.

This option overrides the COLOR= suboption of the MARKERATTRS= option.

Note: Do not use this option and the ATTRID= option in the same plot statement. Doing so can cause unexpected results.

See: "Example 1: Using Group Options without an Attribute Map" on page 1323

Chapter 18, "Using Discrete Attribute Maps," on page 1901

MSATTRID=*discrete-attr-variable*
specifies an attribute map ID variable to be associated with the GROUPMS= variable. This option enables the plot to associate an attribute map with the marker symbol group. The option specifies the value of the ID variable that maps attributes to MARKERSYMBOL in the attribute map.

Requirement The procedure statement must specify the attribute map data set using the DATTRMAP= option.

Interactions This option is ignored unless the GROUPMS= option is specified.

This option is ignored unless the MARKERS option is specified.

This option overrides the SYMBOL= suboption of the MARKERATTRS= option.

Note Do not use this option and the ATTRID= option in the same plot statement. Doing so can cause unexpected results.

See "Example 1: Using Group Options without an Attribute Map" on page 1323

Chapter 18, "Using Discrete Attribute Maps," on page 1901

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

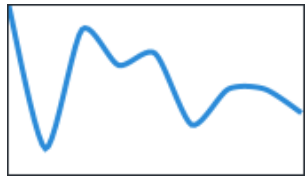
Interaction This option has no effect unless GROUP= is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SMOOTHCONNECT
specifies that a smoothed line passes through all vertices. The following graphics fragments show the effect of using SMOOTHCONNECT.

Default Series	SMOOTHCONNECT Specified
	

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THICKMAX=*numeric-value<units>*
specifies the maximum line thickness when a response variable is used to determine the line thickness. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

By default, this option determines the thickness of the line that represents the maximum response column value.

Default Ten times the thickness that is specified by the GraphDataDefault style element LineThickness attribute.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The THICKMAXRESP= option specifies the response value at which this maximum line thickness is reached. The line thickness for response values that exceed the THICKMAXRESP= value are set to the value that is specified by this option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

THICKMAXRESP=*numeric-value*
specifies the response value that corresponds to the maximum line thickness.

Default The maximum value in the response column that is specified in the THICKRESP= option.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The thickness for all lines that exceed the maximum response value is set to the value specified in the THICKMAX= option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

THICKRESP=*numeric-variable*
specifies a response variable that is used to map a line thickness to each group value.

Default The GraphDataDefault style element LineThickness attribute.

Restriction The THICKRESP= values are assumed to be constant for each group value. If the THICKRESP column has multiple values for a single GROUP value, only one of the THICKRESP= values is used for that group.

Requirement The GROUP= option must be specified with the THICKRESP= option. Otherwise, the THICKRESP= option is ignored.

Interactions When the column values are all zero, all negative, or all missing, this option is ignored. In that case, the default line thickness is used for all of the lines.

The THICKNESS= suboption of the LINEATTRS= option overrides this option for the line thickness attribute.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Examples

Example 1: Using Group Options without an Attribute Map

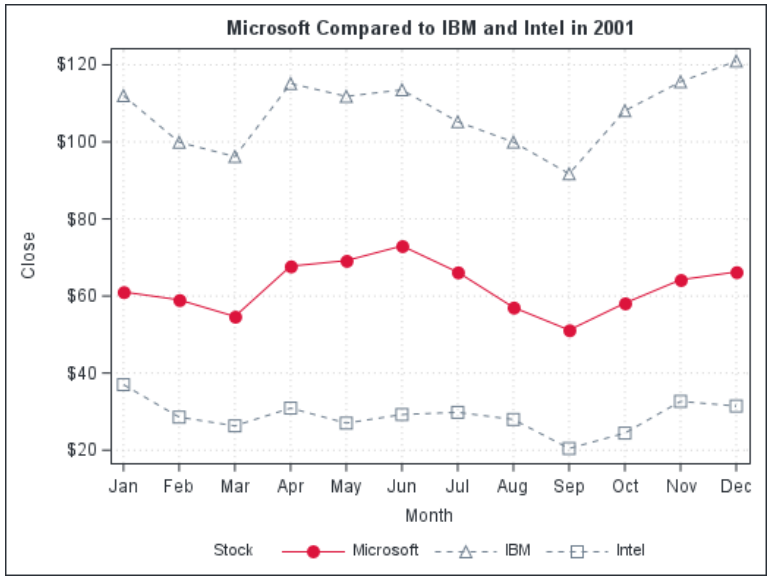
When you specify the GROUP= on page 1335 option, you specify a variable that is used to group the lines of your series plot. The plot lines and markers, if enabled, for each group value are automatically distinguished by different visual attributes.

The SERIES plot statement also includes four group options that give you additional control over grouped output:

- GROUPLC= on page 1312 specifies a variable that determines the line colors for the grouped series plot.
- GROUPLP= on page 1312 specifies a variable that determines the line patterns for the grouped series plot.
- GROUPMC= on page 1313 specifies a variable that determines the marker colors for the grouped series plot.
- GROUPMS= on page 1313 specifies a variable that determines the marker symbols for the grouped series plot.

When you specify one or more of these group options, you override the particular attribute (line color, line pattern, marker color, or marker symbol) from the GROUP= option.

The following example shows line and marker attributes that are specified using the GROUP=, GROUPLC=, GROUPLP=, GROUPMC=, and GROUPMS= options. The GROUPLC=, GROUPLP=, GROUPMC=, and GROUPMS= options are used to distinguish the Microsoft stock closing values from the IBM and Intel stock closing values. The Microsoft line is a solid crimson-red line with filled circle markers. The IBM line is a gray dashed line with triangle markers, and the Intel line is a gray dashed line with square markers.




```
data stocks;                                                    /* 1 */
   set sashelp.stocks(where=(year(date) eq 2001));
   select(stock);
      when("Microsoft") linegrp=1;
      when("IBM")       linegrp=2;
      otherwise         linegrp=3;
   end;
   month = put(date, monname3.);
   keep month close stock linegrp date;
run;

proc sort data=stocks;                                          /* 2 */
   by linegrp stock date;
run;
```

```
ods graphics / attrpriority=none;
title 'Microsoft Compared to IBM and Intel in 2001';
proc sgplot data=stocks;
   styleattrs                                              /* 3 */
      datacontrastcolors=(crimson slategray slategray)
      datalinepatterns=(solid shortdash shortdash)
      datasymbols=(circlefilled triangle square);
   xaxis label="Month" grid
      gridattrs=(color=lightgray pattern=dot);
   yaxis grid gridattrs=(color=lightgray pattern=dot);
   series x=month y=close /
      group=stock                                          /* 4 */
      grouplc=linegrp grouplp=linegrp
      groupmc=linegrp groupms=linegrp
      markers
      markerattrs=(size=8pt)
      lineattrs=(thickness=1px);
   format close dollar5.;
run;
title;
```

1 The DATA step creates data set Stocks by extracting the data for 2001 from data set Sashelp.Stocks. Column Linegrp is added as the line and marker attribute group variable. It is set to 1 for Microsoft, 2 for IBM, and 3 for Intel. Column Month is added as the category variable.

2 The SORT procedure sorts data set Stocks by columns Linegrp, Stock, and Date to ensure that the attributes are assigned in the correct order.

3 The STYLEATTRS statement specifies the group attributes.

4 The GROUP= option specifies the data group variable. The GROUPLC= and GROUPLP= options specify the line attribute group variable, and the GROUPMC= and GROUPMS= options specifiy the marker attribute group variable.

Example 2: Using Group Options with an Attribute Map

You can associate group variables with an attribute map.

Use the following attribute ID variables in the attribute-map data set:

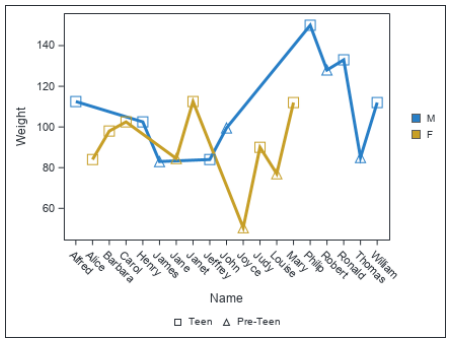
- LCATTRID= on page 1315 associates the GROUPLC= variable with an attribute map ID.
- LPATTRID= on page 1315 associates the GROUPLP= variable with an attribute map ID.
- MCATTRID= on page 1317 associates the GROUPMC= variable with an attribute map ID.
- MSATTRID= on page 1318 associates the GROUPMS= variable with an attribute map ID.

This example shows attribute mapping of grouped markers. The following code creates the plot data and the attribute-map data.

```
/* Create the plot data set.  */
data myclass;
```

```
  set sashelp.class;
  length type $10;
  if age > 12 then type='Teen';
  else type='Pre-Teen';
  label type="Age Group";
run;

/* Create the attribute map data set. */
data mymap;
retain id "mytest";
input value $ markersymbol $;
datalines;
Pre-Teen triangle
Teen square
;
run;
```

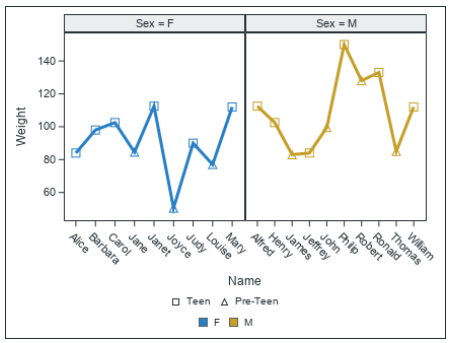
The following series plot shows markers that are determined by an attribute map.



Here is the SAS code that generated this plot.

```
proc sgplot data=myclass dattrmap=mymap;
  series x=name y=weight / markers
     markerattrs=(size=10pt)
     group=sex name="a"
     groupms=type msattrid=mytest;
  keylegend "a" / type=markersymbol;
  keylegend "a" / type=linecolor;
run;
```

The following panel of series plots shows markers that are determined by an attribute map.



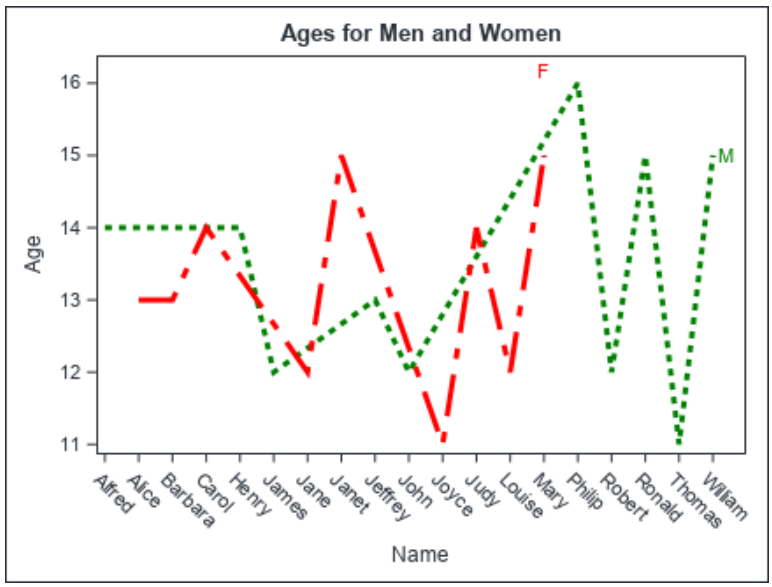
Here is the SAS code that generated this panel.

```
proc sgpanel data=myclass dattrmap=mymap;
```

```
  panelby sex / uniscale=row;
  series x=name y=weight / markers
    markerattrs=(size=10pt)
    group=sex name="a"
    groupms=type msattrid=mytest;
  keylegend "a" / type=markersymbol;
  keylegend "a" / type=linecolor;
run;
```

Example 3: Using Curve Label Options with a Series Plot

This example adds a curve label to identify gender. One line is green with a line pattern of 2, and the other line is red with a line pattern of 26. The CURVELABEL= on page 1333 option adds the curve label. The CURVELABELPOS= on page 1334MAX option places the label at the part of the curve closest to the maximum X-axis value. The ATTRPRIORITY= on page 107NONE option in the ODS GRAPHICS statement ensures that the list of line patterns is rotated.




```
ods graphics / width=4.5in attrpriority=none;
title "Ages for Men and Women";

proc sgplot data=sashelp.class;
  styleattrs datacontrastcolors=(green red) datalinepatterns=(2 26) ;
  series x=name y=age / group=sex  curvelabel curvelabelpos=max;
run;

title;
ods graphics / reset=attrpriority;
```

SPLINE Statement

Creates a series plot with a quadratic Bézier spline interpolation that produces smooth curves.

Notes: The SPLINE statement plots data in data order. For this reason, the input data set should be sorted by the X variable. If it is not sorted by the X variable, unexpected results might occur.

The spline plot is not guaranteed to intersect the original data points. For more information, see "Details" on page 1343.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Spline Plots" on page 34

Syntax

SPLINE X=*variable* Y=*variable* </*options*>;

Summary of Optional Arguments

Appearance options

ARROWHEADPOS=START | END | BOTH
: specifies a position for arrowheads.

ARROWHEADSCALE=*positive-number*
: specifies an arrowhead scale factor based on the thickness of the arrow line.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
: specifies a shape for arrowheads.

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all lines from discrete X or Y values.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the spline line.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

THICKMAX=*numeric-value*<*units*>
: specifies the maximum line thickness when a response variable is used to determine the line thickness.

THICKMAXRESP=*numeric-value*
: specifies the response value that corresponds to the maximum line thickness.

THICKRESP=*numeric-variable*
: specifies a response variable that is used to map a line thickness to each group value.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
: assigns the Y variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
: specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
: specifies how to display grouped lines.

GROUPLC=*variable*
: specifies a variable that determines the line colors for a grouped plot independently of the GROUP= variable.

GROUPLP=*variable*
: specifies a variable that determines the line patterns for a grouped plot independently of the GROUP= variable.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the ordering of the groups within a category.

NOMISSINGGROUP
: specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL
: adds a label to the curve or curves.

CURVELABEL="*text-string*"
: specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)

specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label.

LEGENDLABEL="*text-string*"
specifies a label that identifies the spline plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

BREAK
creates a break in the line for each missing value for the Y variable.

SEGPOINTS=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the INTERVAL= axis option.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*variable*
specifies the variable for the Y axis.

Optional Arguments

ARROWHEADPOS=START | END | BOTH
specifies a position for arrowheads. The arrowheads are placed at the starting and ending points using the X values in data order. In order to position the arrowheads properly, you might need to sort the data by the X column.

START displays an arrowhead at the starting point of each line.

END displays an arrowhead at the ending point of each line.

BOTH displays an arrowhead at each end of each line.

Default No arrowheads are displayed when this option is not specified

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSCALE=*positive-number*
specifies an arrowhead scale factor based on the thickness of the arrow line.

Default 1.0

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

Tip Use a factor greater than 1.0 to make a larger arrowhead.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
specifies a shape for arrowheads. The following figure shows each of the arrowhead shapes.

Default OPEN

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BREAK
creates a break in the line for each missing value for the Y variable.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interaction This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example
```
colormodel=TwoColorRamp
```

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorAltRamp style element

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default INSIDE

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label. Specify one of the following values:

AUTO
places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction This value takes effect only when CURVELABELLOC=OUTSIDE.

END
places the curve label at the last point on the curve.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

MIN
places the label at the part of the curve closest to the minimum X-axis value.

START
places the curve label at the first point on the curve.

Defaults END if CURVELABELLOC=INSIDE

AUTO if CURVELABELLOC=OUTSIDE

Interactions This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 10.45 DATASKIN Options for Lines

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all lines from discrete X or Y values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the X or Y axis is discrete.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped lines.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interaction This option is ignored unless GROUP= is specified.

GROUPLC=*variable*
specifies a variable that determines the line colors for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series line colors are selected from the GraphData1–GraphData*n* style elements based on the GROUPLC= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

This option overrides the COLOR= suboption of the LINEATTRS= option.

Tips Use the GROUPLP= option to set the line pattern.

You can also use LINEATTRS= option to set the line pattern and thickness.

See "Example 1: Using Group Options without an Attribute Map" on page 1323

GROUPLP=*variable*
specifies a variable that determines the line patterns for a grouped plot independently of the GROUP= variable. When this option is specified with the GROUP= option, the series line patterns are selected from the GraphData1–GraphData*n* style elements based on the GROUPLP= variable instead of the GROUP= variable.

Interactions The GROUP= option must be specified for this option to have any effect.

This option overrides the PATTERN= suboption of the LINEATTRS= option.

Tip Use the GROUPLC= option to set the line color. You can also use the LINEATTRS= option to set the line thickness and color.

See "Example 1: Using Group Options without an Attribute Map" on page 1323

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

LEGENDLABEL="*text-string*"
specifies a label that identifies the spline plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the spline line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SEGPOINTS=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the INTERVAL= axis option.

Default 20

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THICKMAX=*numeric-value<units>*
specifies the maximum line thickness when a response variable is used to determine the line thickness. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

By default, this option determines the thickness of the line that represents the maximum response column value.

Default Ten times the thickness that is specified by the GraphDataDefault style element LineThickness attribute.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The THICKMAXRESP= option specifies the response value at which this maximum line thickness is reached. The line thickness for response values that exceed the THICKMAXRESP= value are set to the value that is specified by this option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

THICKMAXRESP=*numeric-value*
specifies the response value that corresponds to the maximum line thickness.

Default The maximum value in the response column that is specified in the THICKRESP= option.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The thickness for all lines that exceed the maximum response value is set to the value specified in the THICKMAX= option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

THICKRESP=*numeric-variable*
specifies a response variable that is used to map a line thickness to each group value.

Default The GraphDataDefault style element LineThickness attribute.

Restriction The THICKRESP= values are assumed to be constant for each group value. If the THICKRESP column has multiple values for a single GROUP value, only one of the THICKRESP= values is used for that group.

Requirement The GROUP= option must be specified with the THICKRESP= option. Otherwise, the THICKRESP= option is ignored.

Interactions When the column values are all zero, all negative, or all missing, this option is ignored. In that case, the default line thickness is used for all of the lines.

The THICKNESS= suboption of the LINEATTRS= option overrides this option for the line thickness attribute.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

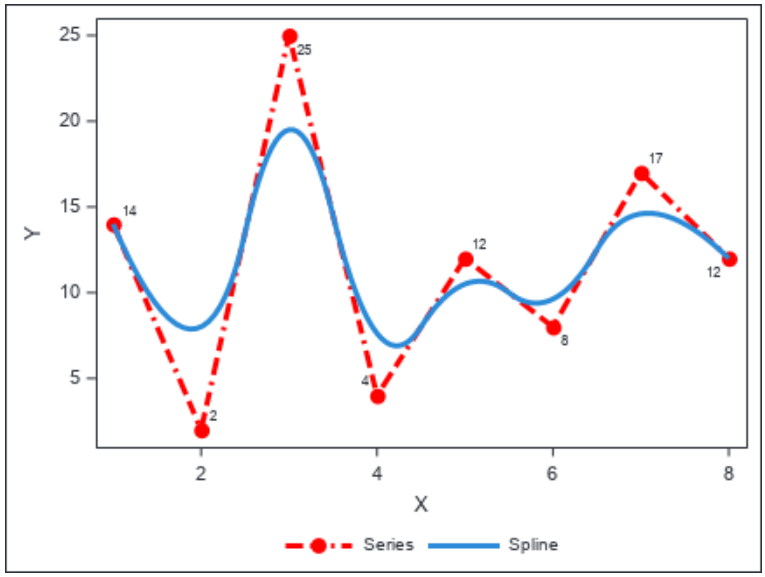
Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Details

Bézier curves are used to achieve smooth curves. They do this by approximating sequences of line segments.

The following figure shows an example of a spline plot overlaid on a series plot. As shown here, the spline plot does not always intersect the original data points.

Figure 10.7 *Spline and Series Plots*



STEP Statement

Creates a step plot.

Restrictions: The vertical axis that is used with the STEP statement cannot be a discrete axis.

The number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID using the LINEATTRS= option.

Note: The STEP statement plots data in data order. For this reason, the input data set should be sorted by the X variable. If it is not sorted by the X variable, unexpected results might occur.

See: "Marker Fills and Outlines" on page 1805

"Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Step Plots" on page 35

Syntax

STEP X=*variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ARROWHEADPOS=START | END | BOTH
: specifies a position for arrowheads.

ARROWHEADSCALE=*positive-number*
: specifies an arrowhead scale factor based on the thickness of the arrow line.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
: specifies a shape for arrowheads.

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all step lines from discrete X values.

ERRORBARATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the error bars in the plot.

ERRORCAPSCALE=*numeric-value*
: specifies a positive number to use as a multiplier for changing the default width of the error-bar caps.

JUSTIFY=LEFT | CENTER | RIGHT
: specifies the location of each step relative to its data point.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the step line.

THICKMAX=*numeric-value*<*units*>
: specifies the maximum line thickness when a response variable is used to determine the line thickness.

THICKMAXRESP=*numeric-value*
: specifies the response value that corresponds to the maximum line thickness.

THICKRESP=*numeric-variable*
: specifies a response variable that is used to map a line thickness to each group value.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
: assigns the Y variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
: specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
: specifies how to display grouped step lines.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the ordering of the groups within a category.

NOERRORCAPS
: suppresses the serif cap on error bars, if error bars are displayed.

NOMISSINGGROUP
: specifies that missing values of the group variable are not included in the plot.

Label options

CURVELABEL
: adds a label to the curve or curves.

CURVELABEL="*text-string*"
: specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC=OUTSIDE | INSIDE
: specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS=AUTO | END | MAX | MIN | START
: specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
: displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
: specifies the location of each data label with respect to its data point.

LEGENDLABEL="*text-string*"
: specifies a label that identifies the step plot in the legend.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the text for curve labels is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

Marker options

FILLEDOUTLINEDMARKERS
: specifies that markers have a fill and an outline.

MARKERATTRS=*style-element*<(**options**)> | (*options*)
: specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
: specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the marker outlines.

MARKERS
: adds markers to the step plot data points.

Plot options

URL=*character-variable*
: specifies an HTML page to be displayed when parts of the plot are selected.

YERRORLOWER=*numeric-variable*
: specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER=*numeric-variable*
: specifies a variable that contains the upper endpoints for the Y error bars.

Plot reference options

NAME="*text-string*"
: assigns a name to a plot statement.

Step options

BREAK
: creates a break in the line for each missing value for the Y variable.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*numeric-variable*
specifies the variable for the Y axis.

Optional Arguments

ARROWHEADPOS=START | END | BOTH
specifies a position for arrowheads. The arrowheads are placed at the starting and ending points using the X values in data order. In order to position the arrowheads properly, you might need to sort the data by the X column.

START displays an arrowhead at the starting point of each line.

END displays an arrowhead at the ending point of each line.

BOTH displays an arrowhead at each end of each line.

Default No arrowheads are displayed when this option is not specified

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSCALE=*positive-number*
specifies an arrowhead scale factor based on the thickness of the arrow line.

Default 1.0

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

Tip Use a factor greater than 1.0 to make a larger arrowhead.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ARROWHEADSHAPE=OPEN | FILLED | BARBED
specifies a shape for arrowheads. The following figure shows each of the arrowhead shapes.

Default OPEN

Interaction ARROWHEADPOS= must also be specified for this option to have any effect.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BREAK
creates a break in the line for each missing value for the Y variable.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interactions This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

For interval data, when markers are displayed and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default INSIDE

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label. Specify one of the following values:

AUTO
places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction This value takes effect only when CURVELABELLOC=OUTSIDE.

END
places the curve label at the last point on the curve.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

MIN
places the label at the part of the curve closest to the minimum X-axis value.

START
places the curve label at the first point on the curve.

Defaults END if CURVELABELLOC=INSIDE

AUTO if CURVELABELLOC=OUTSIDE

Interactions This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT

CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. The data skin affects all line marker symbols, if they are displayed. If the marker symbol is not filled, then the data skin is applied to the outlines. Specify one of the following:

Table 10.46 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

If you specify markers with the plot, then the data skin affects the markers as well.

Table 10.47 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all step lines from discrete X values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the X axis is discrete.

ERRORBARATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the error bars in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes attributes of the GraphError style element in the current style.

For grouped data, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphDataDefault style element in the current style.

ERRORCAPSCALE=*numeric-value*
specifies a positive number to use as a multiplier for changing the default width of the error-bar caps. For example, ERRORCAPSCALE=2 doubles the default cap width and ERRORCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction This option has no effect if NOERRORCAPS is specified.

Tip The thickness (height) of a cap honors the line thickness specified in the ERRORBARATTRS= option. The cap pattern is always solid.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interaction This option has no effect unless MARKERS is also specified.

Tip Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped step lines.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interactions This option is ignored unless GROUP= is specified.

For interval data, when markers are displayed, and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced. The size could be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

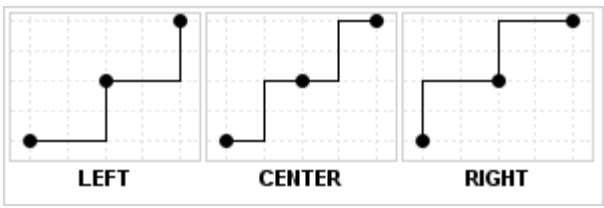
By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

JUSTIFY=LEFT | CENTER | RIGHT
specifies the location of each step relative to its data point. Figure 10.30 on page 1354 shows the effect of each option:

Figure 10.8 *Values for JUSTIFY=*



LEGENDLABEL="*text-string*"
specifies a label that identifies the step plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the step line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the

current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interactions This option has no effect unless you also specify the MARKERS option.

Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

Suboption COLOR= in option LINEATTRS= overrides the default marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness

Requirement Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKERS
adds markers to the step plot data points.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOERRORCAPS
suppresses the serif cap on error bars, if error bars are displayed.

Interaction The appearance of the error bars is controlled by the ERRORBARATTRS= option.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THICKMAX=*numeric-value<units>*
specifies the maximum line thickness when a response variable is used to determine the line thickness. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

By default, this option determines the thickness of the line that represents the maximum response column value.

Default Ten times the thickness that is specified by the GraphDataDefault style element LineThickness attribute.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The THICKMAXRESP= option specifies the response value at which this maximum line thickness is reached. The line thickness for response values that exceed the THICKMAXRESP= value are set to the value that is specified by this option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

THICKMAXRESP=*numeric-value*

specifies the response value that corresponds to the maximum line thickness.

Default The maximum value in the response column that is specified in the THICKRESP= option.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The thickness for all lines that exceed the maximum response value is set to the value specified in the THICKMAX= option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

THICKRESP=*numeric-variable*

specifies a response variable that is used to map a line thickness to each group value.

Default The GraphDataDefault style element LineThickness attribute.

Restriction The THICKRESP= values are assumed to be constant for each group value. If the THICKRESP column has multiple values for a single GROUP value, only one of the THICKRESP= values is used for that group.

Requirement The GROUP= option must be specified with the THICKRESP= option. Otherwise, the THICKRESP= option is ignored.

Interactions When the column values are all zero, all negative, or all missing, this option is ignored. In that case, the default line thickness is used for all of the lines.

The THICKNESS= suboption of the LINEATTRS= option overrides this option for the line thickness attribute.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

TIP=(*variable-list*) | NONE

specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)

a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE

suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Notes When TIP= is not specified, the X, Y, and GROUP variables are displayed, if applicable, when you hover over a line or marker. The X and ERRORUPPER or ERRORLOWER variables are displayed when you hover over an error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a line or marker. No variables are displayed when you hover over an error bar or error bar cap.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

YERRORLOWER=*numeric-variable*
specifies a variable that contains the lower endpoints for the Y error bars.

YERRORUPPER=*numeric-variable*
specifies a variable that contains the upper endpoints for the Y error bars.

SYMBOLCHAR Statement

Defines a marker symbol using a Unicode character that can be referenced in other statements.

Note: You can use more than one SYMBOLCHAR statement.

Syntax

SYMBOLCHAR NAME=*identifier*
CHAR="*hex-string*"x | *keyword* < */options*>;

Summary of Optional Arguments

HOFFSET=*offset*
specifies a horizontal offset for the marker symbol.

ROTATE=*number*
specifies the angle of rotation for the marker symbol in degrees.

SCALE=*double*
specifies a scale factor for the marker symbol as a percentage.

TEXTATTRS=*style-element*<(***options***)> | (*options*)
specifies the text attributes for the character symbol.

VOFFSET=*offset*
specifies a vertical offset for the marker symbol.

Required Arguments

NAME=*identifier*
specifies a unique identifier for the marker symbol. The identifier can be used in statements that support marker symbols. If multiple SYMBOLCHAR statements are used in a procedure, each statement must have a unique *identifier*. The identifier is not case-sensitive.

Interaction If the identifier matches one of the system-provided symbol names such as CIRCLE, the system symbol is replaced by the user-defined symbol. See marker symbols on page 1834.

Note Do not enclose the identifier in quotation marks.

CHAR="*hex-string*"x | *keyword*
specifies a glyph (character) to be used as the marker symbol. The character is specified using its Unicode specification or its keyword equivalent.

***"hex-string"*x**
specifies a four-byte hexadecimal constant, such as '03c3'x, that represents a Unicode character in the current font.

You can find a complete listing of the Unicode hexadecimal constants at the following URL: http://www.unicode.org/charts/charindex.html

keyword
specifies a SAS keyword for a Unicode character, such as alpha or alpha_u.

See Appendix 2, “Reserved Keywords and Unicode Values,” on page 2081.

Note: The “_u” in a keyword makes the character uppercase.

TIP This statement attempts to access the specified Unicode value in the current font. Some fonts do not support accessing characters using their Unicode value while other fonts support only a limited set of Unicode values. If the Unicode value is not accessible, this statement might be ignored or a nonprintable character might be substituted.

Optional Arguments

HOFFSET=*offset*
specifies a horizontal offset for the marker symbol.

Default 0 (the marker symbol is centered on its data point)

Range -0.5 to +0.5, where 0.5 represents one-half of the original marker size.

Interaction The specified offset is also applied to the marker symbol that is displayed in the legend.

Note A positive offset moves the marker symbol to the right within its bounding box, and a negative offset moves it to the left. The bounding box remains centered on the data point. After the offset, the size, and the rotation are applied to the marker symbol, any portion of the symbol that falls outside of the marker bounding box is clipped.

Tip If clipping occurs, you can use this option, the VOFFSET= option, and the SCALE= option to remove the clipping. In the accompanying plot statement that uses markers, you can also use the suboption SIZE= in the MARKERATTRS= option to help remove the clipping.

ROTATE=*number*
specifies the angle of rotation for the marker symbol in degrees. Positive angles are measured in the counterclockwise direction, and negative angles are measured in the clockwise direction.

Default 0 (no rotation is performed)

Note An angle that exceeds 360 degrees in absolute value can be specified.

SCALE=*double*
specifies a scale factor for the marker symbol as a percentage. The scale factor is applied to the character's height.

Default 1.0 (100%)

Range Greater than zero. Very small (for example, 0.1) or very large (for example, 4) scale factors can make the markers invisible or truncated, respectively.

TEXTATTRS=*style-element*<(*options*)> | (*options*)
specifies the text attributes for the character symbol. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Note: Only the text attributes FAMILY=, STYLE=, and WEIGHT= are used. The color and size text attributes are derived from the MARKERATTRS= option in the plot statement.

Default The FontFamily, FontStyle, and FontWeight attributes of the GraphUnicodeText style element in the current style.

Examples `TEXTATTRS=(Family='Albany AMT' Style=Italic Weight=Bold)`

Here is an example that specifies a style element:
`TEXTATTRS=GraphDataText`

VOFFSET=*offset*
specifies a vertical offset for the marker symbol.

Default 0 (the marker symbol is centered on its data point)

Range -0.5 to +0.5, where 0.5 represents one-half of the original marker size.

Interaction The specified offset is also applied to the marker symbol that is displayed in the legend.

Note A positive offset moves the marker symbol up within its bounding box, and a negative offset moves it down. The bounding box remains centered on the data point. After the offset, the size, and the rotation are applied to the marker symbol, any portion of the symbol that falls outside of the marker bounding box is clipped.

Tip If clipping occurs, you can use this option, the HOFFSET= option, and the SCALE= option to remove the clipping. In the accompanying plot statement that uses markers, you can also use the suboption SIZE= in the MARKERATTRS= option to help remove the clipping.

Details

The SYMBOLCHAR statement defines a custom marker symbol from a Unicode character. The custom marker symbol can be specified in any plot statement that uses markers.

Unicode symbol specifications are not validated at compile time. An invalid specification might not generate a warning when the output is rendered and might create unexpected results.

The custom marker symbol can also be specified for the value of the MARKERSYMBOL variable in an attribute map.

The procedure can contain more than one SYMBOLCHAR statement. In addition, you can have more than one SYMBOLCHAR statement that defines the same character. This enables you to use SYMBOLCHAR statements in IF/ELSE statements.

For an example that uses more than one SYMBOLCHAR statement, see the SAS blog article More symbols, you say?.

Note: You might need to specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement to ensure that the data does not use color as the primary distinction between symbols. The need to specify ATTRPRIORITY=NONE depends on which ODS style you are using. If your results do not cycle through the symbols, try specifying ATTRPRIORITY=NONE in the ODS GRAPHICS statement.

You can modify the appearance of markers that are created by the SYMBOCHAR statement in the following ways:

- use the COLOR=, SIZE=, and TRANSPARENCY= suboptions of the MARKERATTRS= option in the plot statement.

  Note: The WEIGHT= suboption has no effect on these markers.

- use the FAMILY=, WEIGHT= , and STYLE= suboptions of the TEXTATTRS= option in the SYMBOLCHAR statement to format the symbol character. The color and size are controlled using the MARKERATTRS= option.

When you use the SCALE= option, you might need to test the output with multiple trials before you find the best output. This is true especially if you also specify the marker size (using the MARKERATTRS= option in the plot statement).

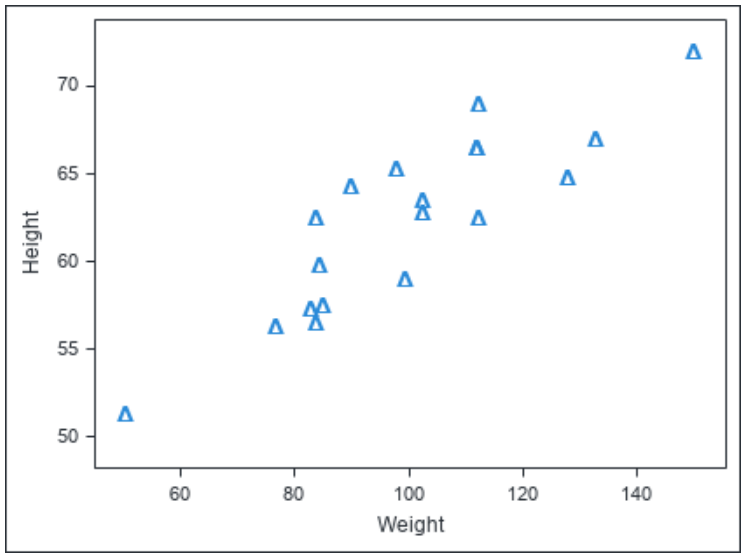
When rendering the graphics output, the procedure performs the following steps:

1. makes adjustments for the HOFFSET and VOFFSET values, if they are specified
2. scales the markers as specified by the SCALE= option
3. clips the markers to the size that is specified in the MARKERATTRS= option in the plot statement
4. rotates the markers, if ROTATE= is specified

Examples

Example 1: Simple SYMBOLCHAR Example

This example specifies the SAS keyword DELTA_U, which produces the delta (Δ) Unicode symbol. A bold font style is applied to the symbol. The scatter plot statement references the symbol name and specifies the marker size.




```
proc sgplot data=sashelp.class;
```

```
    symbolchar name=sym1 char=delta_u / textattrs=(Weight=Bold);
    scatter x=weight y=height / markerattrs=(symbol=sym1 size=15pt);
run;
```

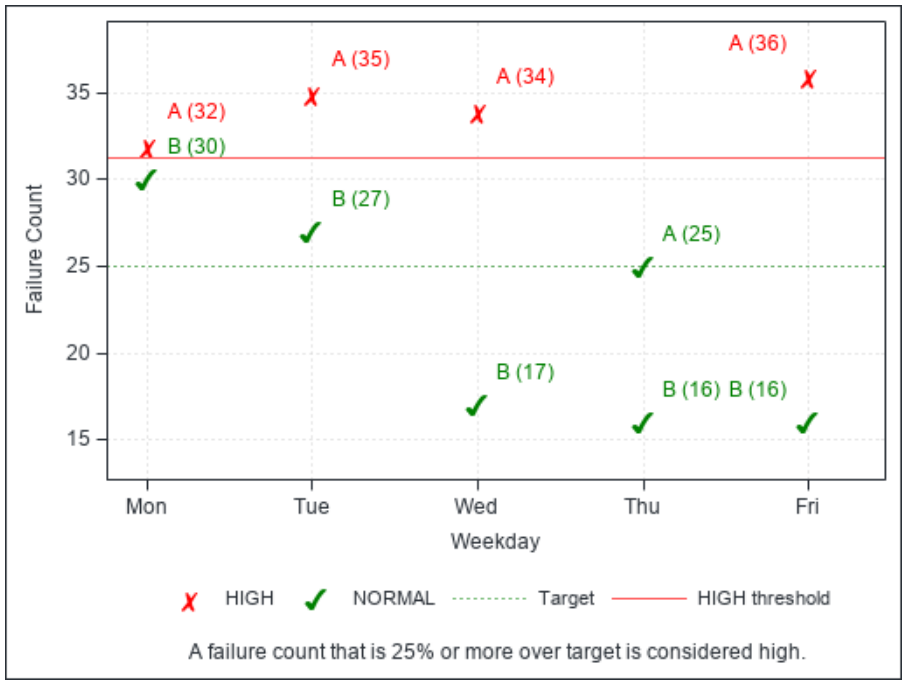
Example 2: SYMBOLCHAR Used with an SG Attribute Map

This example generates different marker symbols for the failure rate in two production processes. The example uses an SG attribute map to assign colors and symbols for each grouping of the variable STATUS.

In the example, the following marker symbols are used:

✓	indicates a failure count that is no more than 25% over the target failure count
✗	indicates a failure count that is over the target failure count by more than 25%

The procedure uses two SYMBOLCHAR statements to identify the marker symbols used in the plot. The example creates marker symbols from the Unicode check mark, heavy character ('2714'x) and the Ballot X character ('2717'x). The example then shows how to use the symbols in a scatter plot.




```
/* Summarize the data in SASHELP.FAILURE to compute the
   failure count by day and process */
proc sort data=sashelp.failure out=temp;
   by day process;
run;

proc summary data=temp;
   by day process;
   var count;
```

```
   output out=failure sum=count;
run;

/* Determine the daily status for each process based
   on the target failure count */
%let target=25;
data failure;
   length status details $10;
   set failure;
   if (count <= (&target*1.25)) then status="NORMAL";
   if (count > (&target*1.25)) then status="HIGH";
   sympos = count+2;
   details = substr(process, index(process, " ")+1, 1) || " (" ||
      strip(put(count, 3.)) || ")";
   label day="Weekday" count="Failure Count" status="Status";
run;

/* Create the attribute map that assigns
   symbols for each grouping of SIZE. */
data myattrmap;
  length markercolor $15;
  input ID $ value  $ markersymbol $ markercolor $;
  datalines;
myid  NORMAL normal green
myid  HIGH   high   red
;
run;

/*  Specify the marker symbols used in the scatter plot,
    and apply the SG attribute map. */
footnote "A failure count that is 25% or more over target is
considered high.";
proc sgplot data=failure dattrmap=myattrmap;
  symbolchar name=normal char='2714'x;
  symbolchar name=high char='2717'x;
  scatter x=day y=count / name="scatter" attrid=myid datalabel=details
     markerattrs=(size=18pt) datalabelattrs=(size=9pt)
     group=status;
  refline &target / name="target" lineattrs=(color=green pattern=dot)
     legendlabel="Target";
  refline %sysevalf(&target*1.25) / name="HIGH"
     lineattrs=(color=lightred pattern=solid)
     legendlabel="HIGH threshold";
  yaxis offsetmax=0.12 grid gridattrs=(color=lightgray pattern=dot);
  xaxis offsetmax=0.1 valuesformat=downame3. grid
     gridattrs=(color=lightgray pattern=dot);
  keylegend "scatter" "target" "HIGH";
  format day downame3.;
run;
footnote;
```

SYMBOLIMAGE Statement

Defines a marker symbol using an image.

Note: The marker symbol can be referenced in other statements.

Syntax

SYMBOLIMAGE NAME=*identifier* IMAGE="*image-file-specification*" < */options*>;

Summary of Optional Arguments

HOFFSET=*offset*
: specifies a horizontal offset for the marker symbol.

ROTATE=*number*
: specifies the angle of rotation for the marker symbol in degrees.

SCALE=*double*
: specifies a scale factor for the marker symbol as a percentage.

VOFFSET=*offset*
: specifies a vertical offset for the marker symbol.

Required Arguments

NAME=*identifier*
: specifies a unique identifier for the marker symbol. The identifier can be used in statements that support marker symbols. If multiple SYMBOLIMAGE statements are used in a procedure, each statement must have a unique *identifier*. The name identifier is not case-sensitive.

  Interaction: If the identifier matches one of the system-provided symbol names such as CIRCLE, the system symbol is replaced by the user-defined symbol. See marker symbols on page 1834.

  Note: Do not enclose the identifier in quotation marks.

IMAGE="*image-file-specification*"
: specifies the name and location of the image file. The supported image types are GIF, JPEG, and PNG.

  Requirements: The image file specification must be enclosed in quotation marks.

  The image file must be located on the local file system. URL access is not supported.

Example `image="c:\temp\mylogo.gif"`

Optional Arguments

HOFFSET=*offset*
specifies a horizontal offset for the marker symbol.

Default 0 (the marker symbol is centered on its data point)

Range -0.5 to +0.5, where 0.5 represents one-half of the original marker size.

Interaction The specified offset is also applied to the marker symbol that is displayed in the legend.

Note A positive offset moves the marker symbol bounding box to the right. A negative offset moves it to the left.

ROTATE=*number*
specifies the angle of rotation for the marker symbol in degrees. Positive angles are measured in the counterclockwise direction, and negative angles are measured in the clockwise direction.

Default 0 (no rotation is performed)

Note An angle that exceeds 360 degrees in absolute value can be specified.

SCALE=*double*
specifies a scale factor for the marker symbol as a percentage. The scale factor is applied to the character's height.

Default 1.0 (100%)

Range Greater than zero. Very small (for example, 0.1) or very large (for example, 4) scale factors can make the markers invisible or truncated, respectively.

VOFFSET=*offset*
specifies a vertical offset for the marker symbol.

Default 0 (the marker symbol is centered on its data point)

Range -0.5 to +0.5, where 0.5 represents one-half of the original marker size.

Interaction The specified offset is also applied to the marker symbol that is displayed in the legend.

Note A positive offset moves the marker symbol bounding box up. A negative offset moves it down.

Details

The SYMBOLIMAGE statement defines a custom marker symbol from an image that is stored in an image file. The image file must exist on the local file system. URL access is not supported. The supported image formats are GIF, JPG, and PNG.

The custom marker symbol can then be specified in any plot statement that supports the MARKERATTRS= option. These plots include the following:

- marker-based plots such as dot and scatter plots.
- line-based plots that enable the addition of markers. These plots include line plots, fit plots (loess, PBspline, regression), needle, series, and step plots.

The custom marker symbol can also be specified for the value of the MARKERSYMBOL variable in an attribute map.

Symbol specifications are not validated at compile time. An invalid specification might not generate a warning when the output is rendered and might create unexpected results.

You can use the SIZE= and TRANSPARENCY= suboptions of the MARKERATTRS= option to modify the appearance of markers that are created by the SYMBOCHAR statement. The COLOR= and WEIGHT= suboptions have no effect on these markers.

The FILLEDOUTLINEDMARKERS= option also has no effect on these markers.

When you use the SCALE= option, you might need to test the output with multiple trials before you find the best output. This is true especially if you also specify the marker size (using the MARKERATTRS= option in the plot statement).

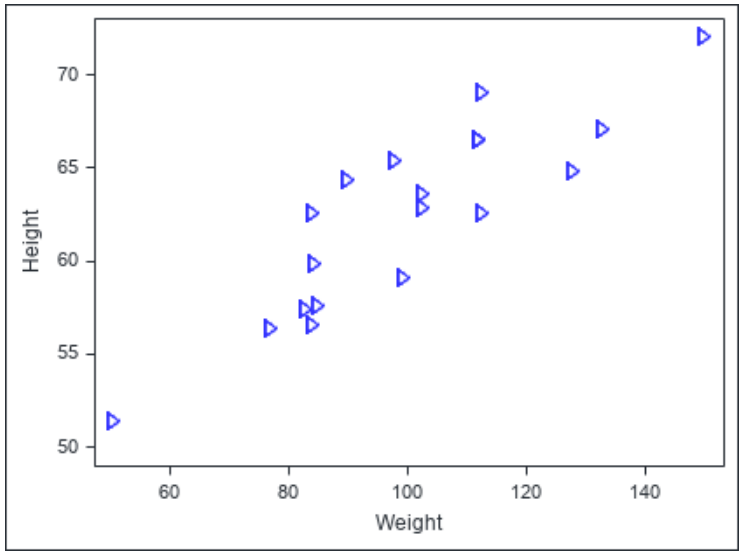
When rendering the graphics output, the procedure performs the following steps:

1. makes adjustments for the HOFFSET and VOFFSET values, if they are specified
2. scales the markers as specified by the SCALE= option
3. clips the markers to the size that is specified in the MARKERATTRS= option in the plot statement
4. rotates the markers, if ROTATE= is specified

Examples

Example 1: Simple SYMBOLIMAGE Example

This example specifies the image file for a triangle (▽). The image is rotated 90 degrees. The scatter plot statement references the symbol name and specifies the marker size.




```
proc sgplot data=sashelp.class;
  symbolimage name=sym1 image="c:\temp\triangle.png" / rotate=90;
  scatter x=weight y=height / markerattrs=(symbol=sym1 size=10pt);
run;
```

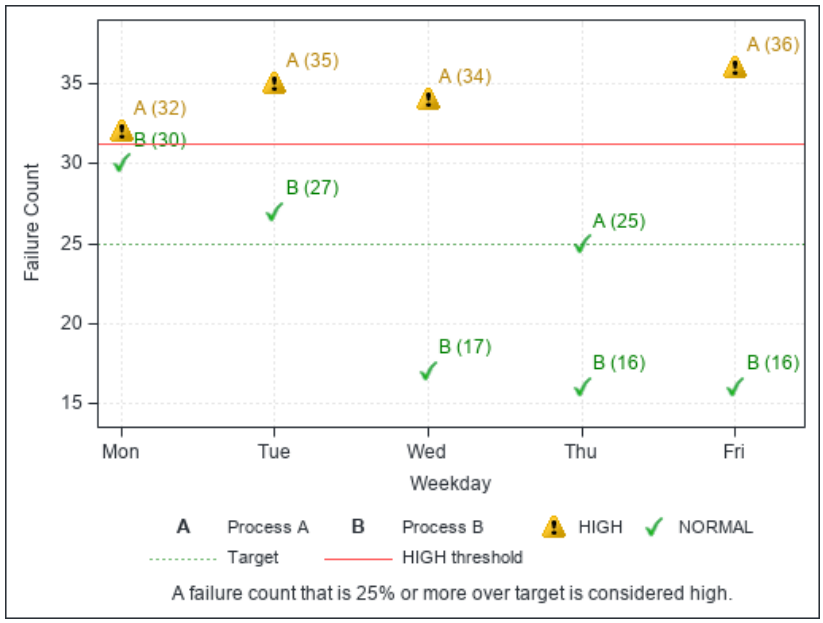
Example 2: SYMBOLIMAGE Used with an SG Attribute Map

This example generates different marker symbols for the failure rate in two production processes. The example uses an SG attribute map to assign colors and symbols for each grouping of the variable STATUS.

In the example, the following images are used:

✓	indicates a failure count that is no more than 25% over the target failure count
⚠	indicates a failure count that is over the target failure count by more than 25%

The procedure uses two SYMBOLIMAGE statements to identify the marker symbols used in the plot. The example then shows how to use the symbols in a scatter plot.




```
/* Summarize the data in SASHELP.FAILURE to compute the
   failure count by day and process */
proc sort data=sashelp.failure out=temp;
   by day process;
run;

proc summary data=temp;
   by day process;
   var count;
   output out=failure sum=count;
run;

/* Determine the daily status for each process based
   on the target failure count. */
%let target=25;
data failure;
   length status details $10;
   set failure;
   if (count <= (&target*1.25)) then status="NORMAL";
   if (count > (&target*1.25)) then status="HIGH";
   sympos = count+2;
   details = substr(process, index(process, " ")+1, 1) || " (" ||
      strip(put(count, 3.)) || ")";
   label day="Weekday" count="Failure Count" status="Status";
run;

/* Create the attribute map that assigns
   symbols for each grouping of SIZE. */
data myattrmap;
  length markercolor $15;
  input ID $ value  $ markersymbol $ markercolor $;
  datalines;
myid  NORMAL normal green
myid  HIGH   high   darkgoldenrod
;
run;
```

```
/*  Specify the marker symbols used in the scatter plot,
    and apply the SG attribute map. */
footnote "A failure count that is 25% or more over target is
considered high.";
proc sgplot data=failure dattrmap=myattrmap;
  legenditem type=text name="A" / text="A" label="Process A"
textattrs=(weight=bold);
  legenditem type=text name="B" / text="B" label="Process B"
textattrs=(weight=bold);
  symbolimage name=normal image='&_workdir\icons
\symbolimgGreenCheck.gif';
  symbolimage name=high image='&_workdir\icons\symbolimgAlert.gif';
  scatter x=day y=count / name="scatter" attrid=myid datalabel=details
     markerattrs=(size=12pt) datalabelattrs=(size=9pt)
     group=status;
  refline &target / name="target" lineattrs=(color=green pattern=dot)
     legendlabel="Target";
  refline %sysevalf(&target*1.25) / name="HIGH"
     lineattrs=(color=lightred pattern=solid)
     legendlabel="HIGH threshold";
  yaxis offsetmax=0.12 grid gridattrs=(color=lightgray pattern=dot);
  xaxis offsetmax=0.1 valuesformat=downame3. grid
     gridattrs=(color=lightgray pattern=dot);
  keylegend "A" "B" "scatter" "target" "HIGH" / across=4;
  format day downame3.;
run;
footnote;
```

TEXT Statement

Displays the associated text values at (X, Y) locations in the graph.

Note: The text can be numbers or characters.

Tip: Use the TEXT statement rather than the SCATTER statement with the MARKERCHAR= option when you want more control over the appearance of the text. The TEXT statement enables you to rotate the text to any angle, manage the text position, split the text into multiple lines, display a bounding box around the text, add a back-light effect to the text, and so on.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Text Plots" on page 36

Syntax

TEXT X=*variable* Y=*variable* TEXT=*variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BACKFILL
: displays filled bounding boxes around the text.

BACKLIGHT<=*number*>
: specifies that the text should have a back light of a contrasting color.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

CONTRIBUTEOFFSETS=ALL | NONE | (*axis-offset-list*)
: specifies whether this plot's space requirements contribute to the calculation of the axis offsets.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all markers from discrete X or Y values.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

OUTLINE
: displays outlined bounding boxes around the text.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the color of the outlines around the text boxes.

PAD=*dimension*<*units*> | *(pad-options)*
: specifies the amount of extra space that is added inside the text-marker border.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

CLUSTERAXIS=X | Y
: specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.

X2AXIS
: assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
: assigns the Y variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped text markers.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Legend options

LEGENDLABEL="*text-string*"
specifies a label that identifies the markers from the plot in the legend.

Plot options

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Text options

POSITION=*position-option* | *character-variable*
specifies the position of the text with respect to the location of the data point.

ROTATE=*number* | *numeric-column*
specifies the angle of rotation in degrees for the text markers.

SIZEMAX=*numeric-value*<*units*>
specifies the maximum font size for the largest text marker when a response variable is used to size the text-marker font.

SIZEMAXRESPONSE=*number*
specifies the response value that corresponds to the maximum font size for text markers.

SIZEMIN=*numeric-value*<*units*>
specifies the minimum font size for text markers when a response variable is used to size the font for text values.

SIZERESPONSE=*numeric-variable*
specifies a response variable that is used to determine the font size for each text value.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the text is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

SPLITPOLICY=NONE | SPLIT | SPLITALWAYS
: specifies a policy for avoiding collisions among the text markers.

SPLITWIDTH=*width-in-characters*
: specifies the maximum width of each split line, expressed as a character count.

STRIP
: specifies that leading and trailing blanks should be stripped from the marker text before it is displayed.

TEXTATTRS=*style-element*<(*text-options*)> | (*text-options*)
: specifies the color and font properties of the marker text.

VCENTER=BBOX | BASELINE
: specifies whether the text is vertically centered with respect to the text bounding box or the text baseline.

Required Arguments

X=*variable*
specifies the variable for the X axis.

Y=*variable*
specifies the variable for the Y axis.

TEXT=*variable*
specifies the variable for the text values that are used for the markers.

Tip If the TEXT= variable contains one or more long character strings, large axis offsets might result. In that case, use the OFFSETMIN= and OFFSETMAX= options in the axis statement to override the automatic axis offsets. See "Plot Axes" on page 1784.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BACKFILL
displays filled bounding boxes around the text.

Tips Use the OUTLINE option to specify outlined boxes around the text.

Use the TRANSPARENCY= option to change the transparency of the backfill and the text. To change the transparency of only the backfill, use the TRANSPARENCY= suboption of the FILLATTRS= OPTION.

BACKLIGHT<=*number*>
specifies that the text should have a back light of a contrasting color. The effect is applied to the marker text only. *number* specifies the degree of the back-light effect.

The following figures show the effect of applying back light to the text. In these examples, both BACKFILL and OUTLINE have also been specified.

BACKLIGHT=0	BACKLIGHT=1

The back light is based on text color. For dark colors, a white back-light effect is used. For lighter colors, a black back-light effect is used. The following figures show the back-light effects when full back light is applied (BACKLIGHT=1). In the first two examples, BACKFILL and OUTLINE have been specified. The third example shows green text against a white background.

Black Text	Gray Text	Green Text

Defaults 0.5 when no GROUP= or COLORRESPONSE= option is used.

0.75 when the GROUP= or COLORRESPONSE= option is used.

Range 0.0–1.0, where 0.0 specifies no effect and 1.0 specifies maximum effect

Note This option is most effective when text color has a low level of contrast with the background. It is also effective when the background is cluttered.

CLUSTERAXIS=X | Y
specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.

Default Uses the discrete axis for clustering groups when only one axis is discrete. Uses the X axis for clustering if both axes are discrete or interval.

Interaction The GROUPDISPLAY= option must be set to CLUSTER for this option to have any effect. The GROUP= option must also be used.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interaction This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorAltRamp style element

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

CONTRIBUTEOFFSETS=ALL | NONE | (*axis-offset-list*)
specifies whether this plot's space requirements contribute to the calculation of the axis offsets. This value determines which axis offsets can be affected by the plot.

The plot statement can implement an offset that prevents clipping of any text strings that appear at the ends of the axes. The offset is based on the longest string. If the character lengths vary significantly, this can result in wasted space when the shorter strings appear near the ends of the axes. In that case, you can use the CONTRIBUTEOFFSETS= option to modify or eliminate this plot's contribution to the offset calculations in order to reclaim that space.

ALL
the space requirements for this plot are contributed to the axis offset calculations.

NONE
the space requirements for this plot are not contributed to the axis offset calculations.

(*axis-offset-list*)
a space-delimited list of specific contributions that this plot makes to the axis offset calculations.

The list is one or more of the following values enclosed in parentheses:

XMAX the space requirements for this plot are contributed to the X-axis offset calculation for the maximum end.

XMIN the space requirements for this plot are contributed to the X-axis offset calculation for the minimum end.

YMAX the space requirements for this plot are contributed to the Y-axis offset calculation for the maximum end.

YMIN the space requirements for this plot are contributed to the Y-axis offset calculation for the minimum end.

Default ALL

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all markers from discrete X or Y values.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset)

Requirement This option is applicable only when the X or Y axis is discrete.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect unless BACKFILL is also specified.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interactions When the COLORRESPONSE= option is specified with the GROUP= option, the GROUP= option is ignored.

When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped text markers.

CLUSTER
grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interaction This option is ignored unless GROUP= is specified.

LEGENDLABEL="*text-string*"
specifies a label that identifies the markers from the plot in the legend. By default, the label of the Y variable or the group value for each marker is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Interaction This option has no effect unless GROUP= is also specified.

OUTLINE
displays outlined bounding boxes around the text.

Tip Use the BACKFILL option to display filled bounding boxes around the text.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the color of the outlines around the text boxes. You can specify the outline color by using a style element or by specifying the COLOR= option. Only option COLOR= is honored. Options PATTERN= and THICKNESS= are ignored. If you specify a style element, you can also specify the COLOR= option to override the color specification in the current style.

For a description of the line COLOR= option, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor attribute of the GraphOutlines style element in the current style.

For grouped data, the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless OUTLINE is also specified.

PAD=*dimension*<*units*> | *(pad-options)*
specifies the amount of extra space that is added inside the text-marker border.

dimension
specifies a dimension to use for the extra space at the left, right, top, and bottom of the text-marker border.

(*pad-options*)
specifies a space-separated list of one or more pad options enclosed in parentheses.

Specify one or more of the following name-value-pair options:

LEFT=*dimension*<*units*> TOP=*dimension*<*units*>
RIGHT=*dimension*<*units*> BOTTOM=*dimension*<*units*>

Default Padding is a fraction of the font height.

Note Sides that are not assigned padding are padded with the default amount.

Tips This option is meaningful only when you also specify OUTLINE, BACKFILL, or both.

Use *pad-options* to create non-uniform padding.

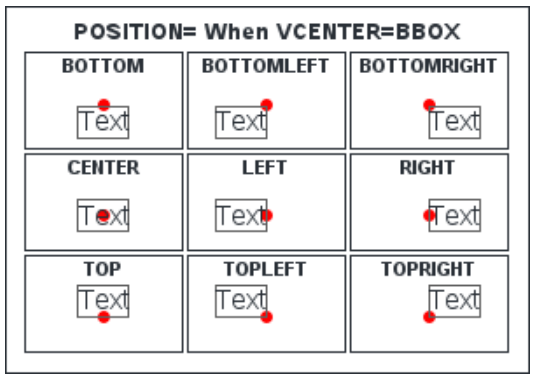
Note The default units for *dimension* are pixels. If you want to specify values in other units, then you must specify the desired units with the value. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

POSITION=*position-option* | *character-variable*
specifies the position of the text with respect to the location of the data point. You can either specify a position option or specify the name of a variable that contains one or more position options.

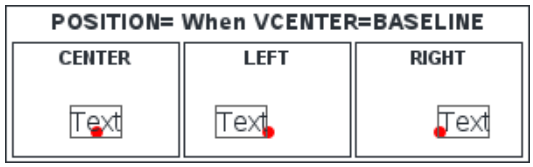
Here are the position options:

BOTTOM	CENTER	TOP
BOTTOMLEFT	LEFT	TOPLEFT
BOTTOMRIGHT	RIGHT	TOPRIGHT

The VCENTER= option specifies whether the position is relative to the text bound box or the text baseline. By default, the positions are relative to the text bounding box. The following figure shows the effect of each of these values on the position of an outlined text when VCENTER=BBOX is in effect. The red dot indicates the data-point location.



When CENTER, LEFT, or RIGHT is specified, and VCENTER=BASELINE is in effect, the positions are relative to the text baseline as shown in the following figure.



Default CENTER

Interaction When *character-variable* is specified and SPLITJUSTIFY= is also specified, the position values specified in *character-variable* are ignored and the text is positioned BOTTOMRIGHT.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

ROTATE=*number* | *numeric-column*
specifies the angle of rotation in degrees for the text markers. Positive angles are measured in the counterclockwise direction, and negative angles are measured in clockwise direction. An angle that exceeds 360 degrees in absolute value can be used.

Default 0

SIZEMAX=*numeric-value<units>*
specifies the maximum font size for the largest text marker when a response variable is used to size the text-marker font. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

By default, the font size of the text values that are associated with the maximum response variable value is set to the value specified by this option.

Default Three times the size specified in the GraphDataText style element for the maximum response variable value marker.

Interactions The SIZERESPONSE= option must be specified for this option to have any effect.

The SIZEMAXRESPONSE= option specifies the response value at which the maximum font size for a text marker is reached. The font size for all text values that exceed the SIZEMAXRESPONSE= value is set to the value specified in this option.

Tips Use the SIZEMAXRESPONSE= option to specify the response value at which the maximum font size for a text marker is reached.

Use the SIZEMIN= option to specify the minimum font size for text markers.

SIZEMAXRESPONSE=*number*
specifies the response value that corresponds to the maximum font size for text markers.

Default The maximum value in the response column specified in the SIZERESPONSE= option.

Interaction The SIZERESPONSE= option must be specified for this option to have any effect.

Note When this option is used with the SIZEMAX= option, the response value is set at the maximum size. Any response values larger than SIZEMAXRESPONSE= are constrained to the SIZEMAX= font size.

SIZEMIN=*numeric-value<units>*
specifies the minimum font size for text markers when a response variable is used to size the font for text values. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The size specified in the GraphDataText style element for the minimum response column value marker.

Interaction The SIZERESPONSE= option must be specified for this option to have any effect.

Tip Use the SIZEMAX= option to specify the maximum text size.

SIZERESPONSE=*numeric-variable*
specifies a response variable that is used to determine the font size for each text value.

Default The size specified in the GraphDataText style element for all text values.

Notes When the variable value for an observation is 0, the font size for the text value for that observation is set to the SIZEMIN= option value.

When the variable value for an observation is negative or missing, the text value for that observation is not displayed in the text plot. However, that observation still contributes to the axis ranges, legend, and so on.

When all the variable values are 0 or missing, this option is ignored. In that case, the default font size is used for all of the text values.

Tip Use the SIZEMIN= and SIZEMAX= options to limit the minimum and maximum font size for the text values.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITPOLICY=NONE | SPLIT | SPLITALWAYS
specifies a policy for avoiding collisions among the text markers.

NONE
does not split the text for text markers that collide.

SPLIT
splits the marker text at a split character only if a split is needed at that character in order to make the text fit the available space.

No split occurs at split characters that occur where a split is not needed. If the text does not contain any of the specified split characters, then a split does not occur. In that case, if the text does not fit the available space, then it might collide with the adjoining text markers.

See See the SPLITCHAR= option for information about specifying the split characters.

SPLITALWAYS
splits the marker text at every occurrence of a split character.

If the text does not contain any of the specified split characters, then a split does not occur.

See See the SPLITCHAR= option for information about specifying the split characters

Default NONE

SPLITWIDTH=*width-in-characters*
specifies the maximum width of each split line, expressed as a character count. When a width is specified, the marker text is split unconditionally after every *width-in-characters* characters.

Default Uses the width of the longest inter-split-character substring.

Restriction This option has effect only when SPLITPOLICY=SPLIT.

STRIP
specifies that leading and trailing blanks should be stripped from the marker text before it is displayed.

Default Blanks are not stripped

Tip Stripping the blanks from numeric value strings helps center each string relative to its data point.

TEXTATTRS=*style-element*<(*text-options*)> | (*text-options*)
specifies the color and font properties of the marker text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the GraphDataText style element in the current style.

For grouped data, the Font attribute of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interactions When this option's COLOR= suboption is used with the GROUP= option, the color of all of the text markers is specified by the COLOR= suboption.

This option's COLOR= suboption overrides the COLORRESPONSE= option. In that case, if a continuous legend is requested for the plot, the legend is not drawn.

Note When the POSITION= option is specified, the data label font size might be reduced in some cases to avoid overlapping labels.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Restriction Data tips in text plots are not supported in the SVG output format. To generate a text plot with data tips, use a bitmapped output format such as PNG. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP OUTPUTFMT=PNG;
```

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted "text strings" enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

VCENTER=BBOX | BASELINE
specifies whether the text is vertically centered with respect to the text bounding box or the text baseline.

BBOX
vertically centers the text with respect to its bounding box.

BASELINE
vertically centers the text with respect to the text baseline.

If the text is split into multiple lines, the text is centered on the baseline of the last line of text.

Restriction This option is valid only when POSITION= is set to CENTER, LEFT, or RIGHT. If POSITION= is set to any other value, VCENTER=BBOX is used instead.

Default BBOX

Tip Use the POSITION= option to specify the text position with respect to the text bounding box or the text baseline.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

VBAR Statement

Creates a vertical bar chart that summarizes the values of a category variable.

Restrictions: The VBAR statement can be combined only with other categorization plot statements in the SGPLOT procedure. See "Plot Compatibility" on page 1780.

When category plots are overlaid, all of the plots must use the same category variable.

Tips: Bar charts can be combined with basic plot types using the HBARBASIC and VBARBASIC statements.

The VBAR statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Examples: "About Bar Charts" on page 53

"Example 10: Creating a Bar-Line Chart" on page 1671

Syntax

VBAR *category-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
: specifies whether the baseline is displayed.

FILL

NOFILL
: specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLENDCOLOR=*color*
: specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
specifies the thickness of the bars when the category data is numeric.

NOZEROBARS
suppresses zero-length bars.

OUTLINE

NOOUTLINE
specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

X2AXIS
assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the response variable to the secondary (right) vertical axis.

Bar options

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each bar.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELFITPOLICY=*policy-value*
specifies a policy for avoiding collisions among the bar labels when bar labels are displayed and DATALABELPOS= is DATA.

DATALABELPOS=DATA | BOTTOM | TOP
specifies the location of the data label.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend.

SEGLABEL
displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

STATLABEL

NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels.

Limit options

LIMITATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITS=BOTH | LOWER | UPPER
specifies which limit lines to display.

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR.

Plot options

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

MISSING
accepts a missing value as a valid category value.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Statistics options

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the vertical axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the bars. The variable generates the midpoints to which each observation in the data set contributes.

Interval bar charts are supported when the category axis is set to TYPE=LINEAR.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default .05

Interactions This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA= value that you specify is used for all of the plots.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Defaults .8

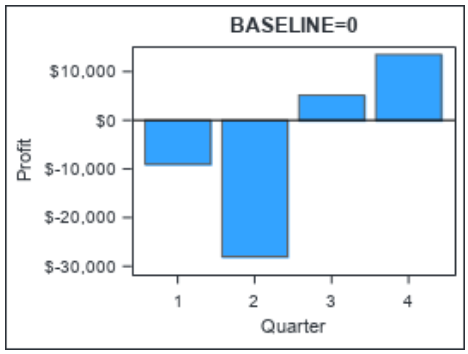
1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interactions If GROUPDISPLAY=STACK is specified, this option is ignored.

When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged. By default, the plot is sorted in ascending order based on the category values. You can change the order so that it is based on the response values.

Specify one of the following values:

RESPASC
sorts by the response values in ascending order.

RESPDESC
sorts by the response values in descending order.

Default The plot is sorted in ascending order based on the category values.

Restrictions This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Uniform scaling and response sorting cannot occur on the same axis. If the UNIFORM= option is used in the SGPLOT statement, the UNIFORM option is ignored for the sorted response axis and a note is generated in the log. The UNIFORM= option is applied to the other axes and groups if requested. Note that the UNIFORM= option can selectively apply scaling to only the X or Y axis

Interactions When a group variable is used with the CATEGORYORDER= option, the category order is not affected by the value of the groups. The categories are always sorted by the response statistic at a category level.

When this option and the GROUPORDER= option are both specified, the GROUPORDER= option is ignored.

Notes If CATEGORYORDER= is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

When two or more observations have the same response values, there is no guarantee that the tied values will be sub-sorted.

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified and when GROUPDISPLAY=CLUSTER.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Requirement The COLORRESPONSE= option must also be specified, or COLORSTAT= must be FREQ or PCT. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

When COLORRESPONSE= is specified, the following values are valid:

SUM
sum values for the color response

MEAN
mean values for the color response

MEDIAN
median values for the color response

Default SUM when option COLORRESPONSE= is also specified.

Requirement The COLORRESPONSE= option must also be specified. If COLORRESPONSE= is not specified, COLORSTAT= is ignored.

Note This option is independent of the STAT= and RESPONSE= options.

DATALABEL
DATALABEL=*variable*
displays a label for each bar.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITS or LIMITSTAT, then the limit labels are also shown. Limit labels are also shown when you use DATALABEL without a variable.

This option is supported with the GROUP= option only when GROUPDISPLAY=CLUSTER is specified. Otherwise, the label is not shown. The labels are shown if you also specify the DATALABELPOS= option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=*policy-value*
specifies a policy for avoiding collisions among the bar labels when bar labels are displayed and DATALABELPOS= is DATA. Select one of the following values:

NONE
does not rotate the bar labels.

Labels that are too long overlap.

ROTATE
rotates the text 90 degrees, but only if collisions occur.

SPLIT
splits the labels at the character or characters specified in the SPLITCHAR= option.

No split occurs at split characters that occur where a split is not needed. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Tips Use the SPLITCHAR= option to specify a split character.

The split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

SPLITALWAYS
always splits the values at the character or characters specified in the SPLITCHAR= option.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Tips Use the SPLITCHAR= option to specify a split character.

The split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

Defaults ROTATE if the column is numeric.

SPLIT if the column is character.

Requirement For this option to have any effect, the DATALABEL= option must also be specified and DATALABELPOS=DATA must be in effect.

Note When DATALABELPOS= is TOP or BOTTOM, the default label fit policy for XAXISTABLE applies.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

DATALABELPOS=DATA | BOTTOM | TOP
specifies the location of the data label. Specify one of the following values:

DATA
places the label on the data primitives (at the upper boundary of the bars).

BOTTOM
places the label below the bars on the bottom edge of the data area.

TOP
places the label above the bars on the top edge of the data area.

Default DATA

Interactions This option displays limit information when limits are specified. When limits are specified, the default data label position is BOTTOM.

This option displays group values for each category when GROUP= is also specified.

This option displays response values for each overlaid chart.

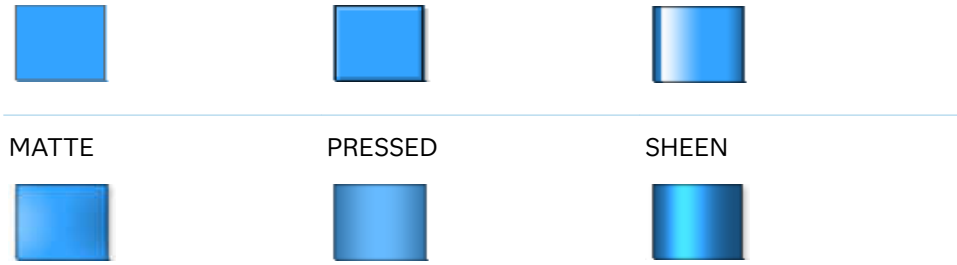
This option has no effect unless you also specify the DATALABEL option.

When you specify a value of TOP or BOTTOM for this option, the splitting or rotation of data labels is not supported.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.48 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS



Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

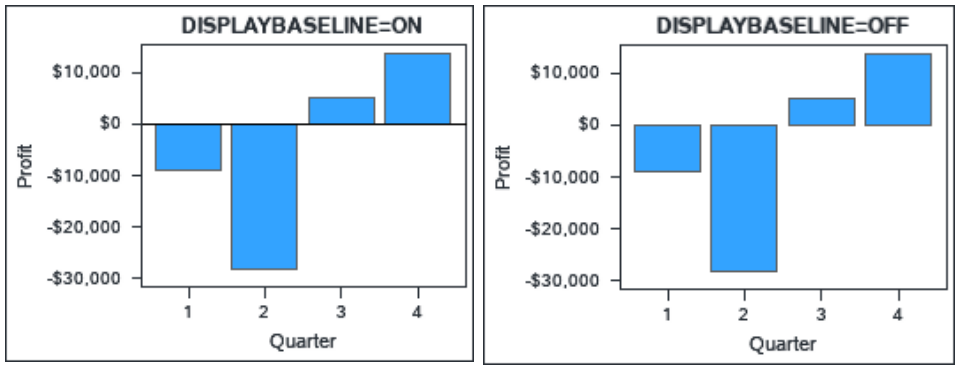
DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.



Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL
specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

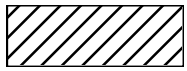
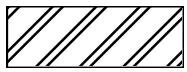
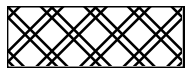
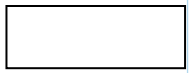
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.49 Fill Patterns

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions

For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note

This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interactions If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

When FREQ= and MISSING are both specified, the MISSING option has no effect.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP= is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default ASCENDING

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

INTERVALBARWIDTH=*numeric-value<units>*
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is "Frequency".

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITS= or LIMITSTAT= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITS=BOTH | LOWER | UPPER
specifies which limit lines to display. Limits are displayed as heavier line segments with a serif at the end extending from each bar. By default, no limits are displayed unless you specify either the LIMITS= or LIMITSTAT= option. If you specify the LIMITSTAT= option only, then LIMITS=BOTH is the default. Specify one of the following values:

BOTH
adds lower and upper limit lines to the plot.

LOWER
adds lower limit lines to the plot.

UPPER
adds upper limit lines to the plot.

By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interactions Limit lines are displayed only when you specify STAT=MEAN.

If you use the GROUP= option in the plot statement, the LIMITS= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines. Specify one of the following statistics:

CLM
confidence limits

STDDEV
standard deviation

STDERR
standard error

Default If you specify the LIMITS= option, then the default for LIMITSTAT= is CLM. Otherwise, there is no default.

Interactions If you specify the LIMITSTAT= option only, then the default value for the LIMITS= option is BOTH.

Limits lines are displayed only when you specify STAT=MEAN.

If you use the GROUP= option in the plot statement, the LIMITSTAT= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

MISSING

accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"

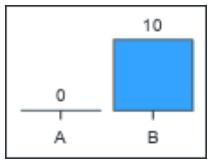
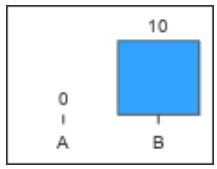
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS

suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

NUMSTD=*n*

specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR. You can specify any positive number, including decimals.

Default 1

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphOutlines style element
- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

This option is ignored if the DATASKIN= option is also specified.

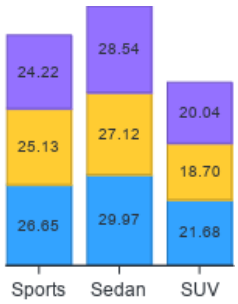
RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the vertical axis.

SEGLABEL
displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

NONE
no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP
does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN
drops any bar segment label that does not fit within its segment.

The label width must not exceed the bar width, and the text height must not exceed the segment height.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear

consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP

specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM

specifies the statistic for the vertical axis. Specify one of the following:

FREQ

the frequencies of the response or category variable.

The frequencies are calculated as follows:

- If you specify the RESPONSE= option, FREQ calculates the frequency of the response variable.
- If you do not specify the RESPONSE= option, FREQ calculates the frequency of the category variable.

MEAN

the mean of the response variable.

Interaction For STAT=MEAN to take effect, you must also specify the RESPONSE= option.

MEDIAN

the median of the response variable.

Interaction For STAT=MEDIAN to take effect, you must also specify the RESPONSE= option.

PERCENT

the percentage of the response or category variable.

The percentage is calculated as follows:

- If you specify the RESPONSE= option, PERCENT calculates the percentage of the sum of the response variable.
- If you do not specify the RESPONSE= option, PERCENT calculates the percentage of the frequency of the category variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the PCTLEVEL= option in the PROC SGPLOT statement.

You can use the PCTNDEC= option in the SGPLOT procedure statement to control the number of decimals to be used when calculating the percent values.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level are zero.

SUM

the sum of the response variable.

This is the default value when you specify the RESPONSE= option.

Interaction For this value to take effect, you must also specify the RESPONSE= option.

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Restriction If you do not also specify the RESPONSE= option, then only the FREQ or PERCENT statistic is calculated (FREQ is the default). If you specify RESPONSE=, then you can use any of the statistics.

Interaction When the graph is generated, the statistic is appended to the variable name in the axis label and the legend (if it is created). However, if a label has been assigned to the variable, then the label appears in the axis label and legend instead of the statistic.

STATLABEL | NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels. STATLABEL forces the statistic to be displayed. NOSTATLABEL removes the statistic from the axis and legend labels.

Normally, the procedure displays the statistic along with the name of the response variable. However, when a custom label is assigned to the response variable, the procedure does not display the statistic. In each case, you can control whether the statistic is displayed.

Defaults The statistic is displayed for the response variable.

When a custom label is assigned to the response variable, the statistic is not displayed.

Interactions This option has no effect unless the RESPONSE= option is specified.

This option has no effect if you specify the axis label using the LABEL= option in an AXIS statement.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces the information that is displayed by default. However, if RESPONSE= is not specified and TIP= is other than NONE, the computed frequency is added to the specified tips.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

If your categorical chart is overlaid with other categorical charts, each chart can specify its own URL variable. The URL variable specified for each chart overlay is honored. This capability affects the following charts: DOT, HBAR, HLINE, VBAR, VLINE.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction If your plot is overlaid with other categorization plots that also specify WEIGHT=, then the first WEIGHT variable that you specified is used for all of the plots.

X2AXIS
assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the response variable to the secondary (right) vertical axis.

VBARBASIC Statement

Creates a vertical bar chart that is compatible with other categorization charts as well as basic plots, such as scatter and series plots, and box plots.

Notes: When using the VBARBASIC statement with a procedure that uses the BY statement, axes are not guaranteed to be uniform across BY groups.

The bars are sorted in data order based on the category values. The sort order cannot be changed. This arrangement of the bars differs from the VBAR statement, which by default sorts bars in alphabetical order based on the category values.

Tip: The VBARBASIC statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Bar Charts" on page 53

Syntax

VBARBASIC *category-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
: specifies whether the baseline is displayed.

FILL

NOFILL
specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency.

FILLENDCOLOR=*color*
specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
specifies the thickness of the bars when the category data is numeric.

NOZEROBARS
suppresses zero-length bars.

OUTLINE

NOOUTLINE
specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

X2AXIS
assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the response variable to the secondary (right) vertical axis.

Data tip options

TIP=(*role-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*

specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL
displays the bar statistic value for each bar.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL option.

DATALABELFITPOLICY=NONE
specifies that no fit policy is implemented for the bar labels.

DATALABELFORMAT=*format*
specifies the text format used to display the bar label.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend.

SEGLABEL
displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Plot options

MISSING
accepts a missing value as a valid category value.

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Statistics options

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors.

STAT=FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic for the vertical axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the bars. The variable generates the midpoints to which each observation in the data set contributes.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Defaults .8

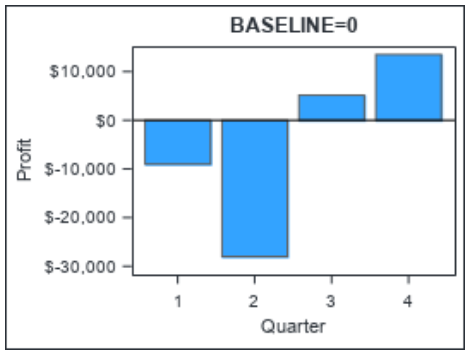
1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interaction When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified and when GROUPDISPLAY=CLUSTER.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Requirement The COLORRESPONSE= option must also be specified, or COLORSTAT= must be FREQ or PCT. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Interaction This option is ignored if COLORSTAT=FREQ or COLORSTAT=PCT.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

COLORSTAT=FREQ | PCT | SUM | MEAN | MEDIAN
specifies the statistic to use for computing the response colors. Specify one of the following values, depending on whether the COLORRESPONSE= option is also specified. When COLORRESPONSE= is not specified, the following values are valid:

FREQ
frequency count

PCT
percentages between 0 and 100

Alias PERCENT

When COLORRESPONSE= is specified, the following values are valid:

SUM
sum values for the color response

MEAN
mean values for the color response

MEDIAN
median values for the color response

Default SUM when option COLORRESPONSE= is also specified.

Requirement The COLORRESPONSE= option must also be specified. If COLORRESPONSE= is not specified, COLORSTAT= is ignored.

Note This option is independent of the STAT= and RESPONSE= options.

DATALABEL
displays the bar statistic value for each bar. For grouped clustered bars, each bar is labeled with the summarized value of the bar. For grouped stacked bars, the segmented bar is labeled with the accumulated, summarized value of all the bar segments.

Default No label is shown

Tip The font and color attributes for the label are specified by the DATALABELATTRS= option. The text format is specified by the DATALABELFORMAT= option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=NONE
specifies that no fit policy is implemented for the bar labels. By default, if the data labels collide, the fit policy is to rotate the data labels 90 degrees, and if they still collide, drop all of the data labels. As a result, the labels might not be visible. To show the labels regardless of how they fit, specify DATALABELFITPOLICY=NONE.

Default Rotate the labels 90 degrees, and if they still collide, drop all of the labels.

Interaction This option has no effect unless DATALABEL is also specified.

DATALABELFORMAT=*format*
specifies the text format used to display the bar label.

Default The column format assigned to the RESPONSE= column, or BEST6 if no format is assigned.

Interaction This option has no effect unless DATALABEL= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.50 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS

MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

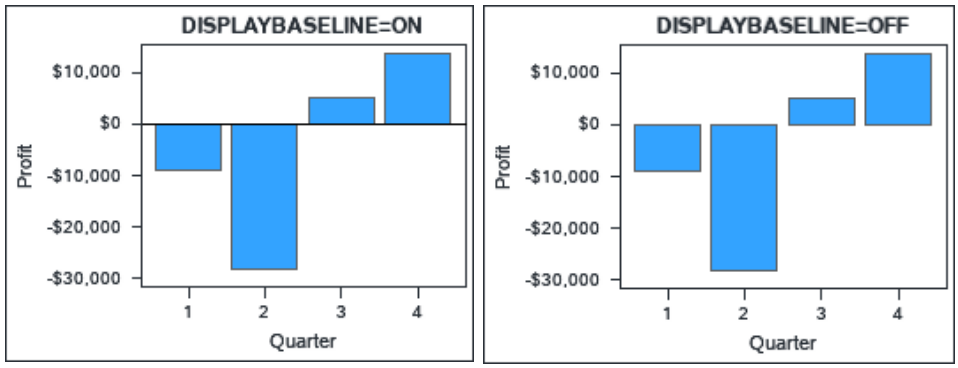
DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.



Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)

specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interaction This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*

specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

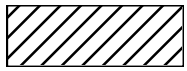
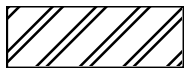
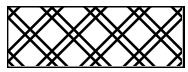
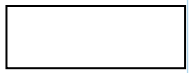
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.51 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions

For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note

This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, “Using Attribute Maps to Control Visual Attributes,” on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP= is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use

default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INTERVALBARWIDTH=*numeric-value<units>*
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable and the computed statistic (SUM or MEAN) is used. If the RESPONSE= option is not used, the legend label is "Frequency".

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

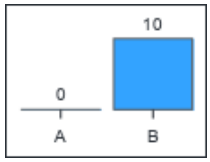
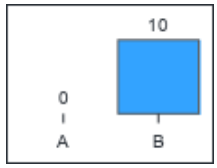
MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS
suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see “Line Attributes and Patterns” on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphOutlines style element
- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the

ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

This option is ignored if the DATASKIN= option is also specified.

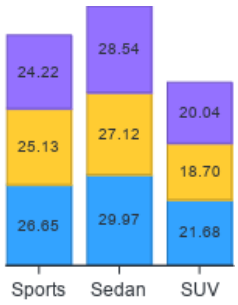
RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed for each value on the horizontal axis.

SEGLABEL
displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

NONE
no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP
does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN
drops any bar segment label that does not fit within its segment.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

STAT=FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic for the vertical axis. For bar charts with no RESPONSE= variable, the following values are valid:

FREQ
frequency count

PERCENT
percentages between 0 and 100

Alias PCT

PROPORTION
proportions between 0 and 1

For bar charts with a RESPONSE= variable, the following values are valid:

SUM
sum values for the response

MEAN
mean values for the response

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Interaction When this option is used with the GROUP=group option, the specified statistic is computed for each segment that is created for the unique group values.

TIP=(*role-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*role-list*)
a space-separated list of unique chart roles enclosed in parentheses.

The roles for TIP are CATEGORY, GROUP, and RESPONSE. Specify one or more of these role names in *role-list*. Data tips are displayed using the data obtained from the specified roles.

Note: You must specify the GROUP and RESPONSE roles for the chart in order to use those roles for data tips.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(category response)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option. Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the response variable to the secondary (right) vertical axis.

VBARPARM Statement

Creates a vertical bar chart based on a pre-summarized response value for each unique value of the category variable.

Requirements: Parameterized bar charts require a response variable in addition to the category variable. The response variable contains pre-summarized computed values such as a sum or a mean for each unique value of the category variable. This is the key distinction between VBARPARM and VBAR.

The data must contain only one response value per unique category variable. If more than one value is found, a warning is written to the SAS log, and the graph might produce unpredictable results.

Interaction: The VBARPARM statement can be combined only with other basic plot statements in the SGPLOT procedure. See "Plot Compatibility" on page 1780.

Notes: You can also assign variables to the upper and lower limits.

The bars are sorted in data order based on the category values. The sort order cannot be changed. This arrangement of the bars differs from the VBAR statement, which by default sorts bars in alphabetical order based on the category values.

Tip: The VBARPARM statement can produce fill patterns for bars. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

Example: "About Bar Charts" on page 53

Syntax

VBARPARM CATEGORY=*category-variable* RESPONSE=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
: specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the baseline.

COLORMODEL=*style-element* | (*color-list*)
: specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
: specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all bars from the category midpoints.

DISPLAYBASELINE=AUTO | ON | OFF
: specifies whether the baseline is displayed.

FILL

NOFILL
: specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
: specifies the fill color and transparency.

FILLENDCOLOR=*color*
: specifies a fill color that is applied to the chart at the baseline.

FILLPATTERN
: enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
: specifies the line-fill color and pattern.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

INTERVALBARWIDTH=*numeric-value*<*units*>
specifies the thickness of the bars when the category data is numeric.

NOZEROBARS
suppresses zero-length bars.

OUTLINE

NOOUTLINE
specifies whether the bars have outlines.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline.

X2AXIS
assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the response variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*

displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELFITPOLICY=*policy-value*
specifies a policy for avoiding collisions among the bar labels when bar labels are displayed and DATALABELPOS= is DATA.

DATALABELPOS=DATA | BOTTOM | TOP
specifies the location of the data label.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend.

SEGLABEL
displays a label inside each segment of a stacked bar.

SEGLABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the text properties of the bar segment label text.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Limit options

LIMITATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITLOWER=*numeric-variable*
specifies values for the lower endpoints on the limit lines.

LIMITUPPER=*numeric-variable*
specifies values for the upper endpoints on the limit lines.

Plot options

MISSING
accepts a missing value as a valid category value.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

CATEGORY=*category-variable*
specifies the variable that categorizes the data. All values are treated as discrete values. The input data for this variable should contain unique values. When the category values are not unique, a warning is logged, and multiple bars are superimposed at the duplicated category values. The CATEGORY axis is always discrete.

Interval bar charts are supported when the category axis is set to TYPE=LINEAR.

RESPONSE=*numeric-variable*
specifies a numeric response variable. The input data is expected to be pre-summarized computed values (sum, mean, and so on).

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars.

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

If this option is not specified, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Defaults .8

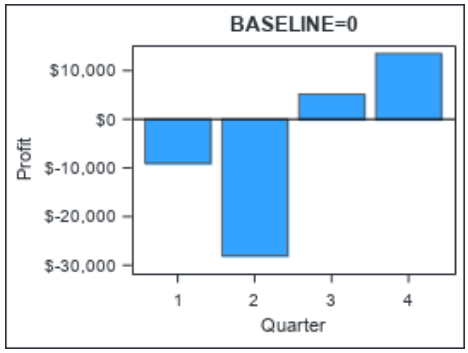
1.0 when the GROUP option is specified and GROUPDISPLAY=CLUSTER

Range 0.0 (narrowest) to 1.0 (widest)

Interaction When the GROUP option is specified, the bar width is determined by the maximum number of bars in any one group cluster. All bars are drawn with the same width. The cluster is positioned symmetrically around the midpoint.

Tip When the GROUP option is specified and GROUPDISPLAY=CLUSTER, you can use the CLUSTERWIDTH= option to control the space between clusters. For example, to remove the space between clusters, set CLUSTERWIDTH=1.

BASELINE=*numeric-value*
specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, even when this option is not specified. In that case, the default value is used. When this option is specified, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.



Default 0

Interaction When a logarithmic response axis is used and BASELINE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINE= to a positive value.

Tips The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, specify DISPLAYBASELINE=OFF. Or, use the BASELINEATTRS= option to set the line thickness to 0.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the maximum width. Specify a value from 0.0 (narrowest) to 1.0 (widest).

CLUSTERWIDTH is the fraction of the midpoint spacing used by all bars that are clustered around a midpoint (category value). The bar width is applied to the

maximum bar spacing divided by the maximum number of bars in any one cluster.

Default 0.8

Interactions This option is applicable only when the GROUP option is specified and when GROUPDISPLAY=CLUSTER.

The default cluster widths can vary for different types of plots. If your graph contains plot overlays, the default cluster width for the primary plot statement is applied to all the other overlaid plots that use default values. If you specify the cluster width for a plot, then your specified value is honored.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example
```
colormodel=TwoColorRamp
```

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The ThreeColorRamp style element for filled bars. The ThreeColorAltRamp style element for unfilled bars.

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

DATALABEL
DATALABEL=*variable*
displays a label for each data point.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITLOWER or LIMITUPPER, then the limit labels are also shown. Limit labels are shown when you use DATALABEL (without a variable).

This option is supported with the GROUP= option only when GROUPDISPLAY=CLUSTER is specified. Otherwise, the label is not drawn. The labels are shown if you also specify the DATALABELPOS= option.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELFITPOLICY=*policy-value*
specifies a policy for avoiding collisions among the bar labels when bar labels are displayed and DATALABELPOS= is DATA. Select one of the following values:

NONE
does not rotate the bar labels.

Labels that are too long overlap.

ROTATE
rotates the text 90 degrees, but only if collisions occur.

SPLIT
splits the labels at the character or characters specified in the SPLITCHAR= option.

No split occurs at split characters that occur where a split is not needed. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Tips Use the SPLITCHAR= option to specify a split character.

The split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

SPLITALWAYS
always splits the values at the character or characters specified in the SPLITCHAR= option.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Tips Use the SPLITCHAR= option to specify a split character.

The split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

Defaults ROTATE if the column is numeric.

SPLIT if the column is character.

Requirement For this option to have any effect, the DATALABEL= option must also be specified and DATALABELPOS=DATA must be in effect.

Note When DATALABELPOS= is TOP or BOTTOM, the default label fit policy for XAXISTABLE applies.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

DATALABELPOS=DATA | BOTTOM | TOP
specifies the location of the data label. Specify one of the following values:

DATA
places the label on the data primitives (at the upper boundary of the bars).

BOTTOM
places the label below the bars on the bottom edge of the data area.

TOP
places the label above the bars on the top edge of the data area.

Default DATA

Interactions This option displays limit information when limits are specified. When limits are specified, the default data label position is BOTTOM.

This option displays group values for each category when GROUP= is also specified.

This option displays response values for each overlaid chart.

This option has no effect unless you also specify the DATALABEL option.

This option does not support the splitting or rotation of data labels.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.52 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all bars from the category midpoints.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset), where 0.5 represents half the distance between category ticks.

Interaction If you specify the REVERSE option in the axis statement, then the offset direction is also reversed.

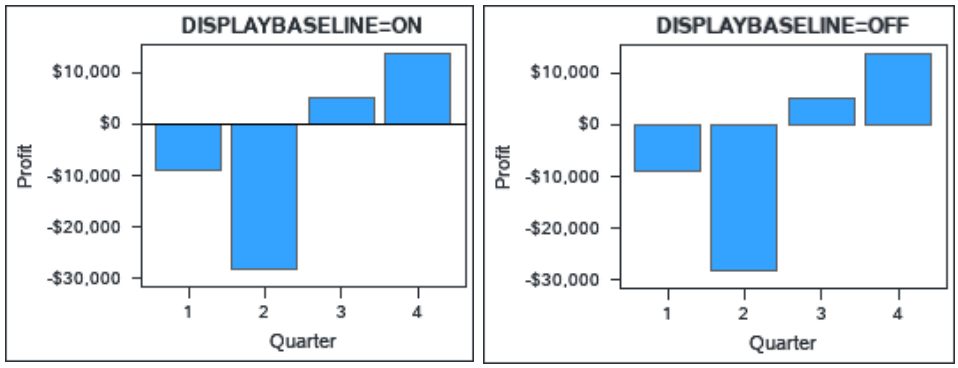
DISPLAYBASELINE=AUTO | ON | OFF
specifies whether the baseline is displayed. Specify one of the following values:

AUTO
displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

In that case, the graph shows both positive and negative bars. If the baseline intercept is equal to the response-axis minimum or maximum value, or is outside of the axis range, the baseline is not displayed.

ON
always displays the baseline.

OFF
hides the baseline.



Default ON

Tip To change the baseline position, use the BASELINE= option. To change the appearance of the baseline, use the BASELINEATTRS= option.

FILL | NOFILL

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)

specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLENDCOLOR=*color*

specifies a fill color that is applied to the chart at the baseline. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The GraphWall style element Color attribute.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless FILLTYPE=GRADIENT.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

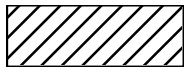
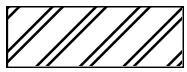
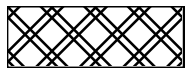
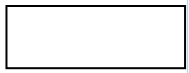
COLOR=*color*
specifies a color to use for the fill lines.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.53 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See "ATTRPRIORITY=COLOR | NONE" on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FILLTYPE=SOLID | GRADIENT | ALPHAGRADIENT
specifies the fill type that is applied to the chart.

SOLID
each bar is filled with the color that is assigned to the bar fill area.

GRADIENT
displays the color from the applied style at the end of the bar gradually changing to the color of the wall at the baseline of the bar.

This option does not affect transparency. Only the color is affected.

Interactions To change the color of the wall, use the FILLENDCOLOR= option.

Data skin SHEEN cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins in that case.

ALPHAGRADIENT
a gradient is used to determine the fill color.

Each bar is filled with a color and transparency gradient. The gradient transitions from the user-specified transparency at the end of the bar to fully transparent at the baseline.

Interaction Data skin SHEEN cannot be used when FILLTYPE=ALPHAGRADIENT is in effect. You can use one of the other data skins in that case.

Tips If your graph contains grid lines or reference lines, this option enables you to see those lines through the transparent portion of the bars.

Use the TRANSPARENCY= chart option, or the TRANSPARENCY= suboption in FILLATTRS=, to set the initial transparency in the gradients.

Default SOLID

Interaction This option has no effect if NOFILL is also specified.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, “Using Attribute Maps to Control Visual Attributes,” on page 1897.

Tips ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

When GROUP= is used with COLORRESPONSE= and FILL is enabled, use FILLPATTERN to distinguish between the group values.

GROUPDISPLAY=STACK | CLUSTER
specifies how to display grouped bars.

STACK
groups are overlaid without any clustering.

All data elements for a given group value are drawn at the exact coordinate, on top of one another. Each group is represented by unique visual attributes

derived from the GraphData1... GraphData*n* style elements in the current style.

CLUSTER
displays group values as separate adjacent bars that replace the single category bar.

Each set of group values is centered at the midpoint tick mark for the category.

Default STACK

Interaction This option is ignored unless GROUP= is specified.

Tip The distance between the group elements in a cluster is controlled by CLUSTERWIDTH=.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted

values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

INTERVALBARWIDTH=*numeric-value<units>*
specifies the thickness of the bars when the category data is numeric. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default The default thickness of the bar is derived from the minimum interval between the data values along the category axis.

Requirement This option is applicable only when the category axis is a linear axis or a TIME axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the bar chart in the legend. By default, the label of the RESPONSE= variable is used. If there is no response variable label, the name of the response variable is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The ContrastColor, LineStyle, and LineThickness attributes of the GraphError style element in the current style.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITLOWER= or LIMITUPPER= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITLOWER=*numeric-variable*
specifies values for the lower endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default The lower segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions If LIMITUPPER= is also specified, then the plot displays the lower and upper segments of the limit lines.

If you use the GROUP= option in the plot statement, the LIMITLOWER= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITUPPER=*numeric-variable*
specifies values for the upper endpoints on the limit lines. Limit lines are displayed as line segments with a serif at the end.

Default The upper segments of the limit lines are not displayed. (Limit lines are displayed only if either LIMITLOWER= or LIMITUPPER= is specified.)

Interactions If LIMITLOWER= is also specified, then the plot displays the lower and upper segments of the limit lines.

If you use the GROUP= option in the plot statement, the LIMITUPPER= option has no effect unless you also specify GROUPDISPLAY=CLUSTER.

The appearance of the limit lines can be controlled by the LIMITATTRS= option.

Tip To display limit labels, specify DATALABEL (without a variable).

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

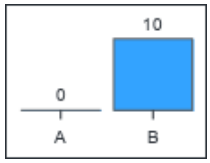
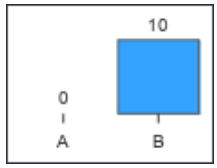
NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOZEROBARS
suppresses zero-length bars. A zero-length bar has a bar length of 0. When this option is specified, zero-length bars are not drawn. The following figure shows a

simple example. In the figure, the graph border, axis line, and bar-chart baseline are suppressed for clarity.

Default	NOZEROBARS Specified
	

Note If BASELINE= is specified, a zero-length bar value equals the baseline.

Tip This option is useful when the bar chart baseline is suppressed.

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

OUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the bar outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828. Note, however, that you cannot specify a line pattern for the bar outline.

Defaults For ungrouped data, the GraphOutlines style element in the current style. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data:

- when only OUTLINE is displayed, the ContrastColor and LineStyle attributes of a GraphData1–GraphData*n* style element, and the LineThickness attribute of the GraphOutlines style element

- when FILL, FILLPATTERN, or both are displayed with OUTLINE, the ContrastColor attribute of a GraphData1–GraphData*n* style element, and the LineThickness and LineStyle attributes of the GraphOutlines style element

Interactions This option has no effect if NOOUTLINE is also specified.

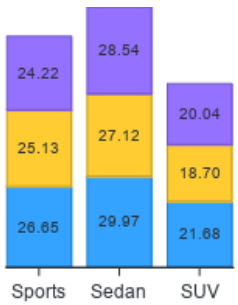
This option is ignored if the DATASKIN= option is also specified.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SEGLABEL
displays a label inside each segment of a stacked bar. For a grouped bar chart when GROUPDISPLAY=STACK, this option displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following fragment that summarizes miles-per-gallon for different makes of vehicles.



Tips For a grouped bar chart when GROUPDISPLAY=STACK, to display a label for each bar segment and a label for the entire bar, specify both SEGLABEL and DATALABEL.

Use the SEGLABELATTRS= option to modify the appearance of the label text.

Use the SEGLABELFORMAT= option to modify the format of the segment labels.

Use the SEGLABELFITPOLICY= option to specify how the labels fit in the segments.

SEGLABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the text properties of the bar segment label text. You can specify the appearance by using a style element or by specifying specific options. If you

specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The GraphDataText style element.

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFITPOLICY=NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

NONE
no attempt is made to fit each segment label within its bar.

Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

NOCLIP
does not clip bar segment labels that extend beyond the plot area.

Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

THIN
drops any bar segment label that does not fit within its segment.

The label width must not exceed the bar width, and the text height must not exceed the segment height.

Default THIN

Interaction This option has no effect unless SEGLABEL is also specified.

SEGLABELFORMAT=*format*
specifies the text format used to display the bar segment labels.

Default The segment-label format is derived from the format that is assigned to the response column or BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction This option has no effect unless SEGLABEL is also specified.

Note When a segment-label format is specified with this option, the segment labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide. If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Notes When TIP= is not specified, the CATEGORY, RESPONSE, COLORRESPONSE, and GROUP variables are displayed, if applicable, when you hover over a bar. The ERRORUPPER or ERRORLOWER variable is displayed when you hover over an error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a bar. No variables are displayed when you hover over an error bar or error bar cap.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see “Using the ODS GRAPHICS Statement” on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the response variable to the secondary (right) vertical axis.

VBOX Statement

Creates a vertical box plot that shows the distribution of your data.

Restriction: This plot has compatibility restrictions when used with other plots. See Table 13.2 on page 1781.

See: "Visual Description of Box Plot Percentile Boundaries" on page 1479

Example: "About Box Plots" on page 47

Syntax

VBOX *numeric-analysis-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

BOXWIDTH=*numeric-value*
specifies the width of the box.

CAPSCALE=*numeric-value*
specifies a positive number to use as a multiplier for changing the default width of the whisker caps.

CAPSHAPE=BRACKET | LINE | SERIF | NONE
specifies the shape of the whisker cap lines.

CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
specifies that a connect line joins a statistic from box to box.

CONNECTATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the lines that connect multiple boxes.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset all boxes from the discrete tick marks.

EXTREME
specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified.

FILL

NOFILL
specifies whether the boxes are filled with color.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency.

FILLPATTERN
enables the display of line fill patterns for the plot.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern.

INTERVALBOXWIDTH=*numeric-value*<*units*>
specifies the box width when an interval category (X) variable is specified.

LINEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the box outlines.

MEANATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the marker that represents the mean in the box.

MEDIANATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the line that represents the median.

NOCAPS
hides the cap lines for the whiskers.

NOMEAN
hides the mean marker.

NOMEDIAN
hides the median line.

NOOUTLIERS
hides the outliers from the plot.

NOTCHES
specifies that the boxes be notched.

OUTLIERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the marker that represents the outliers.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

WHISKERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the whisker and cap lines.

WHISKERPCT=*number*
specifies the whisker length, in percentile units.

Axis options

X2AXIS
assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the analysis variable to the secondary (right) vertical axis.

Data tip options

TIP=(*role-list*) | NONE
specifies the information to display when the cursor is positioned over a box or whisker in the box plot.

TIPFORMAT=(*format-list*)

applies formats to the list of data tip roles that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the midpoint spacing.

GROUP=*variable*
specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped boxes.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

Label options

DATALABEL

DATALABEL=*variable*
adds data labels for the outlier markers.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

LABELFAR
specifies that only the far outliers have data labels.

LEGENDLABEL="*text-string*"
specifies a label that identifies the box plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

CATEGORY=*category-variable*
specifies the category variable for the plot.

DISPLAYSTATS=ALL | (*statistics-list*)
specifies statistics to be displayed for each box.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

MISSING
accepts a missing value as a valid category value.

PERCENTILE=1 | 2 | 3 | 4 | 5
specifies a method for computing the percentiles for the plot.

SPREAD

relocates outlier points that have identical values to prevent overlapping.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Argument

numeric-analysis-variable
specifies the analysis variable for the plot. If you do not specify the CATEGORY= option, then one box is created for the analysis variable.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BOXWIDTH=*numeric-value*
specifies the width of the box. Specify a value between 0.0 (0% of the available width) and 1.0 (100% of the available width).

Defaults 0.4

When GROUP is specified, the default box width is 0.6.

CAPSCALE=*numeric-value*
specifies a positive number to use as a multiplier for changing the default width of the whisker caps. For example, CAPSCALE=2 doubles the default cap width and CAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction This option has no effect if NOCAPS is specified or if CAPSHAPE=NONE.

Tips The thickness (height) of a SERIF, LINE, or BRACKET cap honors the line thickness specified in the WHISKERATTRS= option. The cap pattern is always solid.

Use the CAPSHAPE= option to specify the line shape of the caps.

CAPSHAPE=BRACKET | LINE | SERIF | NONE
specifies the shape of the whisker cap lines. Specify one of the following values:

BRACKET
displays a straight line with brackets.

LINE
displays a straight line.

SERIF
displays a short straight line.

NONE
does not display a cap.

Default SERIF

CATEGORY=*category-variable*
specifies the category variable for the plot. A box plot is created for each distinct value of the category variable.

If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete.

The default ordering of the tick values is ascending formatted order of the *category-variable*. To change this ordering, you can use the DISCRETEORDER=DATA option in the category axis statement.

CLUSTERWIDTH=*numeric-value*
specifies the cluster width as a ratio of the midpoint spacing. Specify a value from 0.1 (narrowest) to 1.0 (widest).

Default 0.7

Interaction This option is applicable only when a GROUP is in effect and the category axis is discrete.

CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
specifies that a connect line joins a statistic from box to box.

Interaction This option applies only when the CATEGORY option is used to generate multiple boxes.

Tip You can use the CONNECTATTRS option to specify attributes for the connect line.

CONNECTATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the lines that connect multiple boxes. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphConnectLine style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interactions This option takes effect only if the CONNECT= option is also specified.

This option is ignored if the GROUP= option is also specified.

Examples
```
CONNECTATTRS=(Color="light green" Pattern=MediumDash Thickness=4)
```

This example specifies a style element:
```
CONNECTATTRS=GraphData3
```

DATALABEL
DATALABEL=*variable*
adds data labels for the outlier markers. If you specified a variable, then the values for that variable are used for the data labels. If you did not specify a variable, then the values of the analysis variable are used.

Note This option has no effect unless the plot contains outlier points.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

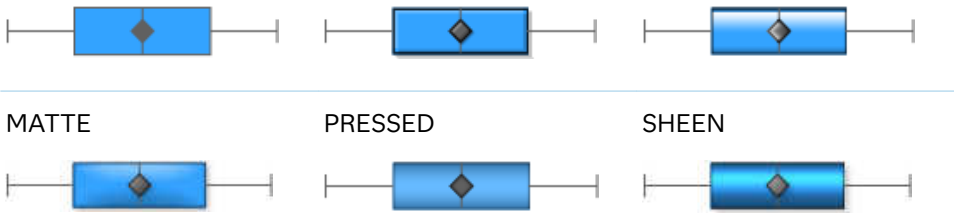
Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all filled boxes. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.54 *DATASKIN Options for Box Plots*

NONE	CRISP	GLOSS



Though the figure shows horizontal boxes, the data skin for vertical boxes is identical.

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interaction If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*

specifies an amount to offset all boxes from the discrete tick marks.

Specify a value from -0.5 (left offset) to +0.5 (right offset). If you specify a value outside of this range, an error message appears in the SAS log and the graph is not produced.

Default 0.0 (no offset)

DISPLAYSTATS=ALL | (*statistics-list*)

specifies statistics to be displayed for each box.

ALL

displays all available statistics.

See the *statistics-list*.

statistics-list

specifies a space-separated list of statistics enclosed in parentheses.

Specify one or more of the following:

DATAMAX maximum data value that includes not only the maximum whisker values but also the maximum outlier values. The DATAMAX value is greater than or equal to the MAX value and is always represented in the axis range.

DATAMIN minimum data value that includes not only the minimum whisker values but also the minimum outlier values. The DATAMIN value is less than or equal to the MIN value and is always represented in the axis range.

IQR	interquartile range (Q3–Q1).
MAX	maximum data value below the box upper fence.
MEAN	mean data value for the box.
MEDIAN	median data value for the box.
MIN	minimum data value above the box lower fence.
N	number of observations for the box.
Q1	lower quartile (25th percentile) for the box.
Q3	upper quartile (75th percentile) for the box.
RANGE	range of the data (MAX–MIN).
STD	standard deviation of the data for the box.
SUMWGT	sum of the weights for the box.

Example `displaystats=(mean median)`

Restriction This option is ignored if the GROUP= option is also specified.

EXTREME
specifies that the whiskers can extend to the maximum and minimum values for the analysis variable, and that outliers are not identified. When you do not specify the EXTREME option, the whiskers cannot be longer than 1.5 times the length of the box.

FILL | NOFILL
specifies whether the boxes are filled with color. The FILL option shows the fill color. The NOFILL option hides the fill color.

Default FILL

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Transparency is the same as the setting for the plot's TRANSPARENCY= option.

Interactions This option has no effect if you specify the NOFILL option.

This suboption overrides the plot's TRANSPARENCY= option for the fill.

FILLPATTERN
enables the display of line fill patterns for the plot.

Interaction
If you specify a group, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement. Example:

```
ods graphics / attrpriority=none;
```

Accessibility note
This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Tips
Use the FILLPATTERNATTRS= option to specify the appearance of the fill pattern.

Use the NOFILL option to suppress the plot's fill color.

FILLPATTERNATTRS=*style-element*<(*options*)> | (*options*)
specifies the line-fill color and pattern. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

style-element
specifies the name of a style element.

options
specifies fill-pattern style options.

Specify one or both of the following options:

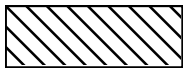
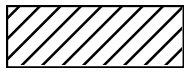
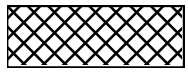
COLOR=*color*
specifies a color to use for the fill lines.

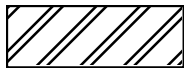
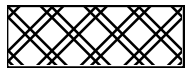
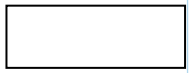
You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. With grouped data, the COLOR= setting has the effect of holding the fill color constant across all group values.

PATTERN=*line-pattern*
specifies the type of line pattern to use for the fill.

line-pattern can be a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for crosshatch) and a line-identification number, 1–5, or E if no fill pattern is desired. The following table shows the patterns for each of the possible combinations.

Table 10.55 *Fill Patterns*

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L1		R1		X1	

Pattern Name	Example	Pattern Name	Example	Pattern Name	Example
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	
E[1, 3]		S[2, 3]			

1 No fill pattern (empty)
2 Solid fill
3 Valid starting with 2026.08

With grouped data, the PATTERN= setting has the effect of holding the fill pattern constant across all group values.

Interactions For ODS styles that use color-priority attribute rotation, such as Illuminate and HTMLBlue, to change the fill pattern for each group value, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement. See “ATTRPRIORITY=COLOR | NONE” on page 107.

For this option to take effect, the FILLPATTERN option must be specified.

Accessibility note This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

FREQ=*numeric-variable*

specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interaction If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY
specifies how to display grouped boxes.

CLUSTER
the boxes are drawn adjacent to each other.

OVERLAY
all the boxes for a given group value are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1... GraphData*n* style elements in the current style.

Defaults CLUSTER for a discrete category axis

OVERLAY for a linear axis

Interaction This option is ignored unless GROUP= is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default DATA

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

INTERVALBOXWIDTH=*numeric-value*<*units*>
specifies the box width when an interval category (X) variable is specified. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Alias INTBOXWIDTH=

Restriction The axis type for the category axis must be LINEAR, and the variable must be numeric.

Example
```
proc sgplot data=sashelp.class;
  vbox weight / category=height intboxwidth=20 ;
  xaxis type=linear;
run;
```

LABELFAR
specifies that only the far outliers have data labels. Far outliers are points whose distance from the box is more than three times the length of the box.

Note This option has no effect if you do not specify the DATALABEL option, or if there are no far outliers.

LEGENDLABEL="*text-string*"
specifies a label that identifies the box plot in the legend. By default, the label of the analysis variable is used.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the box outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style

element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

Interaction This option is ignored if the DATASKIN= option is also specified.

MEANATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker that represents the mean in the box. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphBoxMean style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

MEDIANATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the line that represents the median. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphBoxMedian style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

Interaction This option is ignored if the NOMEDIAN option is also specified.

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOCAPS
hides the cap lines for the whiskers.

Interaction Using several options that hide box features can cause the NOCAPS option to be ignored. For example, if you use NOCAPS, NOFILL, NOMEAN, NOMEDIAN, and NOOOUTLIERS in the same statement, the NOCAPS option might be ignored.

NOMEAN
hides the mean marker.

NOMEDIAN
hides the median line.

NOOUTLIERS
hides the outliers from the plot.

NOTCHES
specifies that the boxes be notched. See "Notches" on page 1480.

OUTLIERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker that represents the outliers. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphOutlier style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

PERCENTILE=1 | 2 | 3 | 4 | 5
specifies a method for computing the percentiles for the plot.

Default 5 (empirical distribution function with averaging)

See "Calculating Percentiles" on page 1481

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPREAD
relocates outlier points that have identical values to prevent overlapping.

Note This option has no effect if your data does not contain two or more outliers with identical values for the analysis variable.

TIP=(*role-list*) | NONE
specifies the information to display when the cursor is positioned over a box or whisker in the box plot.

(*role-list*)
a space-separated list of unique box plot roles enclosed in parentheses.

The box plot roles for TIP are CATEGORY, N, STD, MIN, MAX, MEAN, MEDIAN, Q1, and Q3. Specify one or more of these role names in *role-list*. Data tips are displayed using the data obtained from the specified roles.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(mean median)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip roles that you specify in the TIP= option. Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the data that appears in data tips.

A one-to-one correspondence exists between the *format-list* and the *role-list* that is specified for the TIP= option. A format must be provided for each role, using the same order as the *role-list*. If you do not want to apply a format to a role, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL option to assign labels to the list of roles.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip roles that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the data that appears in data tips.

A one-to-one correspondence exists between the *label-list* and the *role-list* that is specified for the TIP= option. A label must be provided for each role, using the same order as the *role-list*. If you do not want to apply a custom label to a role, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each role that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of roles.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

WHISKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the whisker and cap lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphBoxWhisker style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor, LineStyle, and LineThickness.

WHISKERPCT=*number*
specifies the whisker length, in percentile units. When this option is specified, *number* is used as the low percentile, and 100–*number* is used as the high percentile.

Here are some examples of values and their effect:

0 specifies the high and low extremes

10 specifies the 10th percentile low and the 90th percentile high

25 specifies the 25th percentile low and the 75th percentile high

Default The whiskers are drawn from the box to the most extreme point that is less than or equal to 1.5 times the IQR

Range 0–25

Notes When this option is specified, fences and far outliers are not drawn.

When this option is set to 25, no whiskers are drawn because the box extends from the 25th to the 75th percentile.

X2AXIS
assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the analysis variable to the secondary (right) vertical axis.

Details

Statement Summary

The plot displays a single box if only the analysis variable is provided. The plot displays multiple boxes if a category variable is also provided and that variable has more than one unique value.

The ANALYSIS variable is displayed on the vertical axis. The axis for the analysis column is always LINEAR.

By default for numeric or character columns, the CATEGORY= axis is TYPE=DISCRETE. You can override the default and set TYPE=LINEAR in the axis statement, provided that the category column is numeric.

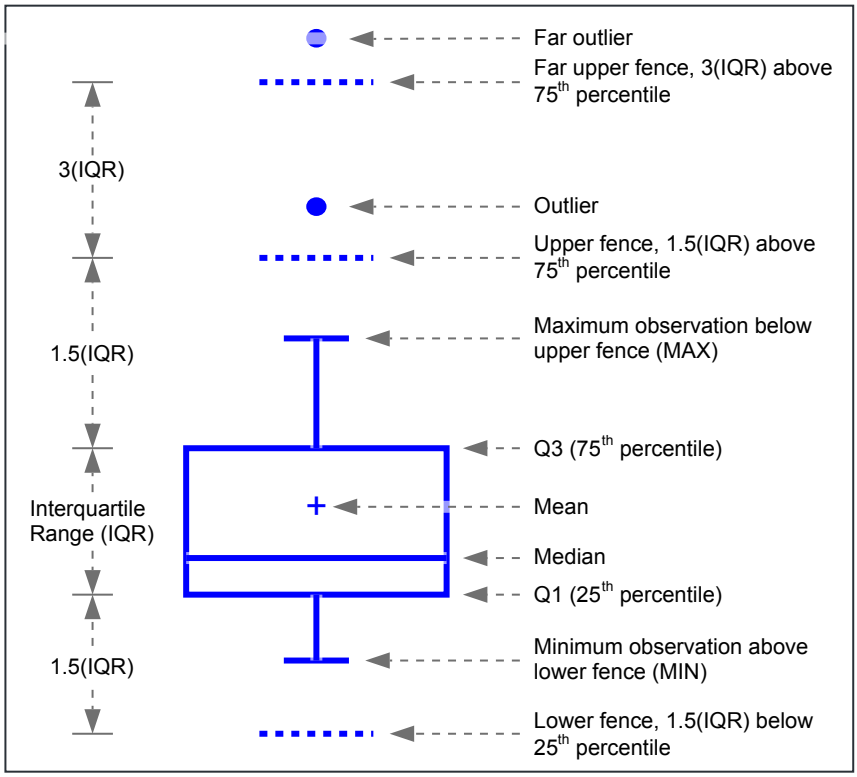
If you explicitly set the category axis type to LINEAR and use a numeric category variable, the box plot becomes an interval plot. Otherwise, the box plot is discrete. For the interval case, you can use the INTBOXWIDTH= option to specify the box width.

Two basic box plot representations can be drawn: a schematic (Tukey) box plot and a skeletal box plot. See the EXTREME option for details.

Visual Description of Box Plot Percentile Boundaries

Box plots display the distribution of data by using a rectangular box and whiskers. Whiskers are lines that indicate a data range outside of the box.

Figure 10.9 Parts of a Box Plot



In the previous figure, the bottom and top edges of the box indicate the intra-quartile range (IQR). That is the range of values between the first and third quartiles (the 25th and 75th percentiles). The marker inside the box indicates the mean value. The line inside the box indicates the median value.

The elements that are outside the box are dependent on your options. By default, the whiskers that extend from each box indicate the range of values that are outside of the intra-quartile range. However, they are close enough not to be considered outliers (a distance less than or equal to 1.5*IQR). If you specify the EXTREME option, then the whiskers indicate the entire range of values, including outliers.

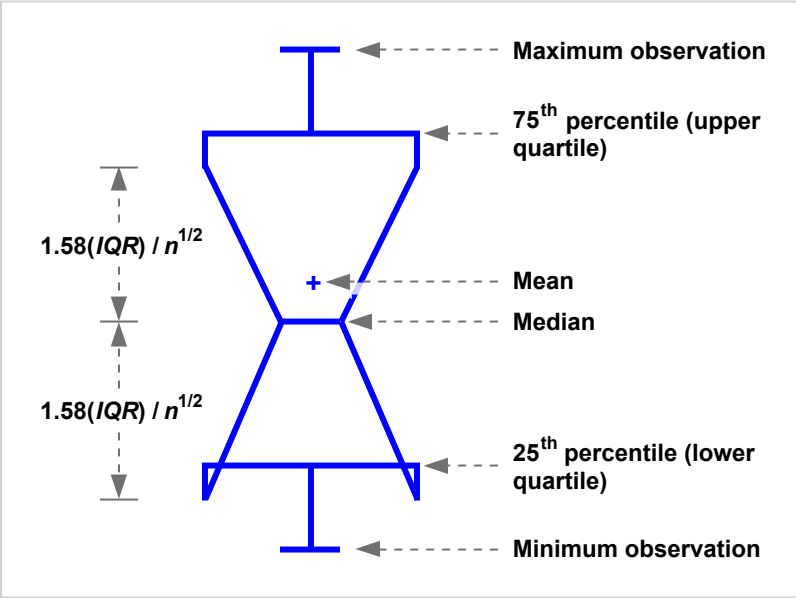
Outliers are observations that are more extreme than the upper and lower fences ($\pm$ 1.5 IQR). Outliers that are beyond upper and lower far fences ($\pm$ 3 IQR) are called FAR OUTLIERS. By default, outliers are indicated by markers. If you specify the DATALABEL= option, then the outlier points have data labels. If you also specify the LABELFAR option, then only outliers that are 3*IQR from the box have data labels.

Notches

In a notched box plot, the endpoints of the notches are at the following computed locations:

$$median \pm 1.58\,(IQR/\sqrt{N})$$

In the equation, IQR (IQR=Q3-Q1) is the interquartile range and *n* is the sample size. The medians (central lines) of the two boxes are significantly different at approximately the 0.05 level if the corresponding notches do not overlap.



Note: In some cases, the notches in the box plot can extend beyond DATAMIN and DATAMAX.

Calculating Percentiles

You can specify one of five definitions for computing the percentiles with the PERCENTILE= option. Let n be the number of nonmissing values for a variable, and let $X_1, X_2, ..., X_n$ represent the ordered values of the variable. X_1 is the smallest value, X_2 is the next smallest, and X_n is the largest value. Let the tth percentile be y, set:

$$p = \frac{t}{100}$$

and let:

$$np = j + g$$

when PERCENTILE=1, 2, 3, or 5, or let:

$$(n+1)p = j + g$$

when PERCENTILE=4, where j is the integer part of np, and g is the fractional part of np. Then the PERCENTILE= option defines the tth percentile, y, as described in the following table:

Percentile	Description	Equation	Notes
1	Weighted average at X_{np}	$y = (1-g)x_j + gx_{j+1}$	x_0 is taken to be x_1

Percentile	Description	Equation	Notes
2	Observation numbered closest to *np*	$y = x_j$	Used when $g < \frac{1}{2}$ or when $g = \frac{1}{2}$ and *j* is even
		$y = x_{j+1}$	Used when $g = \frac{1}{2}$ and *j* is odd or when $g > \frac{1}{2}$
3	Empirical distribution function	$y = x_j$	Used when $g = 0$
		$y = x_{j+1}$	Used when $g > 0$
4	Weighted average aimed at $x_{(n+1)p}$	$y = (1-g)x_j + gx_{j+1}$	x_{n+1} is taken to be x_n
5	Empirical distribution function with averaging	$y = \frac{1}{2}(x_j + x_{j+1})$	Used when $g = 0$
		$y = x_{j+1}$	Used when $g > 0$

VECTOR Statement

Creates a vector plot that draws arrows from a point of origin to each data point.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Vector Plots" on page 39

Syntax

VECTOR X=*numeric-variable* Y=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ARROWDIRECTION=IN | OUT | BOTH
specifies the location of the arrowheads for the vectors.

ARROWHEADSHAPE=*shape*

specifies the shape of the arrowheads for the vectors.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

LINEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the vector line.

NOARROWHEADS
removes the arrowheads from the vectors.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

THICKMAX=*numeric-value*<*units*>
specifies the maximum line thickness when a response variable is used to determine the line thickness.

THICKMAXRESP=*numeric-value*
specifies the response value that corresponds to the maximum line thickness.

THICKRESP=*numeric-variable*
specifies a response variable that is used to map a line thickness to each group value.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
specifies a variable that is used to group the data.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point.

LEGENDLABEL="*text-string*"
specifies a label that identifies the vector plot in the legend.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Plot options

XORIGIN=*numeric-value* | *numeric-variable*
specifies the X coordinate of the origin for the vectors.

YORIGIN=*numeric-value* | *numeric-variable*
specifies the Y coordinate of the origin for the vectors.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

X=*numeric-variable*
specifies a numeric variable for the X axis.

Y=*numeric-variable*
specifies numeric variable for the Y axis.

Optional Arguments

ARROWDIRECTION=IN | OUT | BOTH
specifies the location of the arrowheads for the vectors. Specify one of the following:

IN
places the arrowheads at the origin of the vector.

OUT
places the arrowheads at the ending point of the vector.

BOTH
places arrowheads at both the origin and ending point of the vector.

Default OUT

ARROWHEADSHAPE=*shape*
specifies the shape of the arrowheads for the vectors. Specify one of the following:

OPEN
resembles the letter "V".

Example:

CLOSED
an outline of a triangle.

Example:

FILLED
a solid triangle.

Example:

BARBED
a solid triangle with an indent at the base.

Example:

Default OPEN

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.

ENDCOLOR specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorAltRamp style element

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Tip The color ramp is specified by the COLORMODEL= option.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, then the values of the Y variable are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=*position*
specifies the location of each data label with respect to its data point. *position* can be one of the following values:

BOTTOM	BOTTOMLEFT	BOTTOMRIGHT
CENTER	LEFT	RIGHT
TOP	TOPLEFT	TOPRIGHT

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. Specify one of the following:

Table 10.56 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

GROUP=*variable*
specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

LEGENDLABEL="*text-string*"
specifies a label that identifies the vector plot in the legend. By default, the label of the Y variable or the group value for each plot element is used.

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the vector line. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOARROWHEADS
removes the arrowheads from the vectors.

NOMISSINGGROUP
specifies that missing values of the group variable are not included in the plot.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THICKMAX=*numeric-value<units>*
specifies the maximum line thickness when a response variable is used to determine the line thickness. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

By default, this option determines the thickness of the line that represents the maximum response column value.

Default Ten times the thickness that is specified by the GraphDataDefault style element LineThickness attribute.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The THICKMAXRESP= option specifies the response value at which this maximum line thickness is reached. The line thickness for response values that exceed the THICKMAXRESP= value are set to the value that is specified by this option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

THICKMAXRESP=*numeric-value*
specifies the response value that corresponds to the maximum line thickness.

Default The maximum value in the response column that is specified in the THICKRESP= option.

Interactions The THICKRESP= option must be specified for this option to have any effect.

The thickness for all lines that exceed the maximum response value is set to the value specified in the THICKMAX= option.

If the line thickness that is calculated from the THICKMAX= and THICKMAXRESP= option values is less than 0.5 for a line, that line is not drawn.

THICKRESP=*numeric-variable*
specifies a response variable that is used to map a line thickness to each group value.

Default The GraphDataDefault style element LineThickness attribute.

Restriction The THICKRESP= values are assumed to be constant for each group value. If the THICKRESP column has multiple values for a single GROUP value, only one of the THICKRESP= values is used for that group.

Requirement The GROUP= option must be specified with the THICKRESP= option. Otherwise, the THICKRESP= option is ignored.

Interactions When the column values are all zero, all negative, or all missing, this option is ignored. In that case, the default line thickness is used for all of the lines.

The THICKNESS= suboption of the LINEATTRS= option overrides this option for the line thickness attribute.

See "Example 12: Series Plot with Line-Thickness Response and Arrowheads" on page 1673 for an example of how to use this option.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Restriction The specified transparency is not applied to missing values. Missing values are always opaque.

X2AXIS
assigns the X variable to the secondary (top) horizontal axis.

XORIGIN=*numeric-value* | *numeric-variable*
specifies the X coordinate of the origin for the vectors. You can specify either a numeric value or a numeric variable.

Default 0

Y2AXIS
assigns the Y variable to the secondary (right) vertical axis.

YORIGIN=*numeric-value* | *numeric-variable*
specifies the Y coordinate of the origin for the vectors. You can specify either a numeric value or a numeric variable.

Default 0

VLINE Statement

Creates a vertical line chart (the line is horizontal).

Restrictions: The VLINE statement can be combined only with other categorization plot statements in the SGPLOT procedure. See "Plot Compatibility" on page 1780. and "Plot Axes" on page 1784.

When category plots are overlaid, all of the plots must use the same category variable.

Tip: You can use the VLINE statement with the VBAR statement to create a bar-line chart.

See: "Marker Fills and Outlines" on page 1805

Examples: "About Line Charts" on page 61

"Example 10: Creating a Bar-Line Chart" on page 1671

Syntax

VLINE *category-variable* <*/options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

BREAK
: breaks the line whenever the computed statistic for a category value is missing.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
: specifies a special effect to be used on the plot.

DISCRETEOFFSET=*numeric-value*
: specifies an amount to offset all lines from discrete category values.

LINEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the lines in the line plot.

TRANSPARENCY=*value*
: specifies the degree of transparency for the plot.

Axis options

X2AXIS
: assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS
: assigns the response variable to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
: specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

CLUSTERWIDTH=*numeric-value*
: specifies the width of the group clusters as a fraction of the midpoint spacing.

GROUP=*variable*
: specifies a variable that is used to group the data.

GROUPDISPLAY=CLUSTER | OVERLAY
: specifies how to display grouped lines.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the ordering of the groups within a category.

Label options

CURVELABEL
: adds a label to the curve or curves.

CURVELABEL="*text-string*"
: specifies a label for the curve when only one curve is drawn.

CURVELABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the CURVELABEL= option.

CURVELABELLOC=OUTSIDE | INSIDE
: specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

CURVELABELPOS=AUTO | END | MAX | MIN | START
: specifies the location of the curve label.

DATALABEL

DATALABEL=*variable*
: displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the labels in the plot when you use the DATALABEL= option.

DATALABELPOS=DATA | BOTTOM | TOP
: specifies the location of the data label.

LEGENDLABEL="*text-string*"
: specifies the label that identifies the line plot in the legend.

SPLITCHAR="*character-list*"
: specifies one or more characters on which the text for curve labels is always split.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

STATLABEL

NOSTATLABEL
: specifies whether the response variable statistic is displayed in the axis and legend labels.

Limit options

LIMITATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the limit lines in the plot.

LIMITCAPSCALE=*positive-number*
: specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif).

LIMITS=BOTH | LOWER | UPPER
: adds limit lines to the plot.

LIMITSTAT=CLM | STDDEV | STDERR
: specifies the statistic for the limit lines.

NOLIMITCAPS
: suppresses the serif cap on limit lines, if limit lines are displayed.

NUMSTD=*n*
: specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

MARKERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

MARKERS
adds markers to the plot.

Plot options

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged.

FREQ=*numeric-variable*
specifies a variable for the frequency count for each observation in the input data.

MISSING
accepts a missing value as a valid category value.

RESPONSE=*response-variable*
specifies a numeric response variable for the plot.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Statistics options

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the vertical axis.

Required Argument

category-variable
specifies the variable whose values determine the categories of data represented by the lines.

Optional Arguments

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits. Specify a number between 0.00 (100% confidence) and 1.00 (0% confidence).

Default .05

Interactions This option has no effect if you do not specify LIMITSTAT=CLM.

If your plot is overlaid with other categorization plots, then the first ALPHA= value that you specify is used for all of the plots.

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BREAK
breaks the line whenever the computed statistic for a category value is missing. For example, if a response variable is used and it has all missing values for a certain category value, the SUM or MEAN for this category value is missing. By default in such cases, the response value for the previous category is joined to the response value for the next category value by a line segment. If BREAK is specified, this segment is not drawn.

Note The observation is excluded from the graph when there is a missing value for the FREQ variable.

CATEGORYORDER=RESPASC | RESPDESC
specifies the order in which the categories are arranged. By default, the plot is sorted in ascending order based on the category values. You can change the order so that it is based on the response values.

Specify one of the following values:

RESPASC
sorts by the response values in ascending order.

RESPDESC
sorts by the response values in descending order.

Default The plot is sorted in ascending order based on the category values.

Restrictions This option takes effect only when the plot statement specifies a response variable and the axis for that variable is numeric. If the axis is not numeric, an error is generated and a message is written to the SAS log.

Uniform scaling and response sorting cannot occur on the same axis. If the UNIFORM= option is used in the SGPLOT statement, the UNIFORM option is ignored for the sorted response axis and a note

is generated in the log. The UNIFORM= option is applied to the other axes and groups if requested. Note that the UNIFORM= option can selectively apply scaling to only the X or Y axis

Interactions When a group variable is used with the CATEGORYORDER= option, the category order is not affected by the value of the groups. The categories are always sorted by the response statistic at a category level.

When this option and the GROUPORDER= option are both specified, the GROUPORDER= option is ignored.

Notes If CATEGORYORDER= is specified in multiple statements, the procedure sorts by the last statement in which it is specified.

When two or more observations have the same response values, there is no guarantee that the tied values will be sub-sorted.

CLUSTERWIDTH=*numeric-value*
specifies the width of the group clusters as a fraction of the midpoint spacing. Specify a value from 0.0 (narrowest) to 1.0 (widest).

Default 0.8

Interaction This option is applicable only when the GROUP option is specified, when GROUPDISPLAY=CLUSTER, and when the category axis is discrete.

CURVELABEL
adds a label to the curve or curves. For ungrouped data, the Y variable label is used as the curve label. For grouped data, the group values are used as curve labels.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABEL="*text-string*"
specifies a label for the curve when only one curve is drawn.

Restriction This option is ignored when the GROUP= option is in effect. To label the curves in that case, use CURVELABEL instead.

Interactions If the plot uses a broken linear or time axis, the curve label might not align properly with the curve in some cases.

If you specify VALUES=, MAX=, or MIN= in an axis statement, the points used to determine the position of the curve label might fall outside the graph area. In this case, the curve label might not be displayed, or its position might not be correct.

By default, CURVELABEL= removes a plot from an automatic legend because the curve label is considered redundant information. However, you can override that behavior by specifying the NAME= option in the plot statement. Then include the plot name in a KEYLEGEND statement. If your procedure has more than one plot statement, do this for each plot that you want included in the legend.

CURVELABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the CURVELABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element in the current style.

For grouped data, the FontFamily, FontSize, FontWeight, and FontStyle attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless the CURVELABEL option is also specified.

Examples
```
CURVELABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
CURVELABELATTRS=GraphTitleText
```

CURVELABELLOC=OUTSIDE | INSIDE
specifies whether the curve label is placed inside the plot axes (INSIDE) or outside of the plot axes (OUTSIDE).

Default INSIDE

CURVELABELPOS=AUTO | END | MAX | MIN | START
specifies the location of the curve label. Specify one of the following values:

AUTO
places the curve label outside the plot area near the end of the curve along unused axes whenever possible (typically Y2 or X2).

Interaction This value takes effect only when CURVELABELLOC=OUTSIDE.

END
places the curve label at the last point on the curve.

MAX
places the label at the part of the curve closest to the maximum X-axis value.

MIN
places the label at the part of the curve closest to the minimum X-axis value.

START
places the curve label at the first point on the curve.

Defaults END if CURVELABELLOC=INSIDE

AUTO if CURVELABELLOC=OUTSIDE

Interactions This option has no effect unless the CURVELABEL option is also specified.

The START and END suboptions take effect only when CURVELABELLOC=INSIDE.

DATALABEL
DATALABEL=*variable*
displays a label for each data point.

- If you specify a variable, then the values of that variable are used for the data labels.

  If the values of the label variable are not the same for each category, then the rendered label is indeterminate. In that case, either the first or last-occurring values are displayed. With data that is stored in a table, the minimum values are displayed.

- If you do not specify a variable, then the values of the calculated response are used for the data labels. If you do not specify a response variable, then the frequency counts of the category variable are displayed.

Interactions When DISCRETEOFFSET= and DATALABELPOS= are both specified, the DISCRETEOFFSET= option has no effect on the data labels. Only the plot is offset. If you remove DATALABELPOS= from the plot statement, the data labels are offset.

If you also specify LIMITS or LIMITSTAT, then the limit labels are also shown. Limit labels are shown when you use DATALABEL (without a variable).

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, for the label font, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style. For the label color, when DATALABELPOS=DATA, the ContrastColor attribute of a GraphData1–GraphData*n* style element. Otherwise, the Color attribute of the GraphDataText style element.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family="Arial" Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATALABELPOS=DATA | BOTTOM | TOP
specifies the location of the data label. Specify one of the following values:

DATA
places the label at the data points of the lines.

BOTTOM
places the label below the lines on the bottom edge of the data area.

TOP
places the label above the lines on the top edge of the data area.

Default DATA

Interactions This option displays limit information when limits are specified. When limits are specified, the default data label position is BOTTOM.

This option displays group values for each category when GROUP= is also specified.

This option displays response values for each overlaid chart.

This option has no effect unless you also specify the DATALABEL option.

When you specify a value of TOP or BOTTOM for this option, the splitting of data labels (using the SPLITCHAR= option) is not supported.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot. The data skin affects all plot lines. The data skin affects all line marker symbols, if they are displayed. If the

marker symbol is not filled, then the data skin is applied to the outlines. Specify one of the following:

Table 10.57 *DATASKIN Options for Lines*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

If you specify markers with the plot, then the data skin affects the markers as well.

Table 10.58 *DATASKIN Options for Markers*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

See "Using Data Skins" on page 1851

DISCRETEOFFSET=*numeric-value*

specifies an amount to offset all lines from discrete category values. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

Requirement This option is applicable only when the category axis is discrete.

FILLEDOUTLINEDMARKERS

specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled"

suffix in its name. Otherwise, this option is ignored. See marker symbols on page 1834.

Interaction This option has no effect unless MARKERS is also specified.

Tip Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

FREQ=*numeric-variable*

specifies a variable for the frequency count for each observation in the input data. Each observation is repeated *n* times for computational purposes, where *n* is the value of the numeric variable.

Restrictions If the value of FREQ= for a given observation is missing or is less than 1, that observation is not used in the analysis.

If the value is not an integer, only the integer portion is used.

Interactions If your plot is overlaid with other categorization plots, then the first FREQ variable that you specified is used for all of the plots.

When FREQ= and MISSING are both specified, the MISSING option has no effect.

GROUP=*variable*

specifies a variable that is used to group the data. The plot elements for each group value are automatically distinguished by different visual attributes.

Restriction If you specify a group variable in a category chart, and the procedure contains more than one category chart statement, all of the charts must specify the same GROUP variable. If you do not specify the same GROUP= option for all of the category charts, then an error is generated.

Interaction When the procedure contains both computed and non-computed plot statements, the assignment of group attributes does not always align between the plots. If you need to control the group attribute assignment for these types of plots, consider using an attribute map. For more information, see Chapter 17, "Using Attribute Maps to Control Visual Attributes," on page 1897.

Tip ODS Graphics limits the number of groups to 1000. Use the GROUPMAX= option in the ODS GRAPHICS statement to change the maximum number of groups that can be processed.

GROUPDISPLAY=CLUSTER | OVERLAY

specifies how to display grouped lines.

CLUSTER

grouped items are drawn adjacent to each other.

OVERLAY
grouped items are drawn at the exact coordinate and might overlap.

Each group is represented by unique visual attributes derived from the GraphData1–GraphData*n* style elements in the current style.

Default OVERLAY

Interaction This option is ignored unless GROUP= is specified.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in data order of the group variable.

REVERSEDATA
orders the groups within a category in the reverse data order of the group variable.

ASCENDING
orders the groups within a category in ascending order of the group variable.

DESCENDING
orders the groups within a category in descending order of the group variable.

Default ASCENDING

Interactions The DATA and REVERSEDATA values have no effect when you are using data that is stored in a table.

This option is ignored unless GROUP= is specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER=. The order in the legend can be changed using the SORTORDER= option in the KEYLEGEND statement.

The default group order can vary for different types of plots. If your graph contains plot overlays, the default group order for the first plot statement is applied to all the other overlaid plots that use default values. If you specify the group order for a plot, then your specified value is honored.

Note The ASCENDING and DESCENDING settings linguistically sort the group values within each category for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

Tip The category order can be changed using the DISCRETEORDER= option in the axis statement for the category axis.

LEGENDLABEL="*text-string*"
specifies the label that identifies the line plot in the legend. By default, the label of the response variable is used. If there is no response variable label, then the name of the response variable and the computed statistic (SUM or MEAN) are used. If you do not specify a response variable, then the legend label is "Frequency".

Interaction The LEGENDLABEL= option has no effect if you also specify the GROUP= option in the same plot statement.

LIMITATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the limit lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults For ungrouped data, the ContrastColor, LineStyle, and LineThickness attributes attributes of the GraphError style element in the current style.

For grouped data, the LineStyle and LineThickness attributes of the GraphError style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

LIMITCAPSCALE=*positive-number*
specifies a positive number to use as a multiplier for changing the default width of the limit-bar cap (serif). For example, LIMITCAPSCALE=2 doubles the default cap width and LIMITCAPSCALE=0.5 reduces the default cap width by half.

Default 1

Interaction For this option to have any effect, either LIMITS= or LIMITSTAT= must be specified.

Tip The thickness (height) of the serif cap honors the line thickness specified in the LIMITATTRS= option. The cap pattern is always solid.

LIMITS=BOTH | LOWER | UPPER
adds limit lines to the plot. Specify one of the following values:

BOTH
adds lower and upper limit lines to the plot.

LOWER
adds lower limit lines to the plot.

UPPER
adds upper limit lines to the plot.

By default, no limit lines are displayed. However, if you specify the LIMITSTAT= option, then the default is BOTH.

Interaction Limit lines are displayed only when you specify STAT= MEAN.

Tip To display limit labels, specify DATALABEL (without a variable).

LIMITSTAT=CLM | STDDEV | STDERR
specifies the statistic for the limit lines. Specify one of the following statistics:

CLM
confidence limits

STDDEV
standard deviation

STDERR
standard error

Default If you specify the LIMITS= option, then the default for LIMITSTAT= is CLM. Otherwise, there is no default.

Interaction If you specify the LIMITSTAT= option, then the default value for the LIMITS= option is BOTH.

Tip To display limit labels, specify DATALABEL (without a variable).

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the lines in the line plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and MarkerSymbol.

Interactions This option has no effect unless you also specify the MARKERS option.

Suboption COLOR= is ignored when option FILLEDOUTLINEDMARKERS is also specified. In that case, use option MARKERFILLATTRS= to specify the marker color.

Suboption COLOR= in option LINEATTRS= overrides the default marker color.

MARKERFILLATTRS=*style-element* | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default — For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style. For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

Requirement — Option FILLEDOUTLINEDMARKERS must be specified for MARKERFILLATTRS= to have any effect.

Interaction — This option overrides any color that is specified with the MARKERATTRS= option.

Tip — You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See — For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default — For ungrouped data, the GraphOutlines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are ContrastColor and LineThickness

Requirement — Option FILLEDOUTLINEDMARKERS must be specified for MARKEROUTLINEATTRS= to have any effect.

Tip — You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKERS
adds markers to the plot.

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

Interaction This option has no effect when FREQ= is specified.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOLIMITCAPS
suppresses the serif cap on limit lines, if limit lines are displayed.

Interaction The appearance of the limit lines is controlled by the LIMITATTRS= option.

NUMSTD=*n*
specifies the number of standard units for the limit lines, when you specify LIMITSTAT=STDDEV or LIMITSTAT=STDERR. You can specify any positive number, including decimals.

Default 1

RESPONSE=*response-variable*
specifies a numeric response variable for the plot. The summarized values of the response variable are displayed on the vertical axis.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for curve labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless either CURVELABEL or DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip If you specify data labels and curve labels, this option affects both types of labels. If you do not want to split both types with the same split character, consider using an overlaid plot in your graph. You can then split data labels in one plot and curve labels in the other.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the vertical axis. Specify one of the following:

FREQ
the frequencies of the response or category variable.

The frequencies are calculated as follows:

- If you specify the RESPONSE= option, FREQ calculates the frequency of the response variable.

- If you do not specify the RESPONSE= option, FREQ calculates the frequency of the category variable.

MEAN
the mean of the response variable.

Interaction For STAT=MEAN to take effect, you must also specify the RESPONSE= option.

MEDIAN
the median of the response variable.

Interaction For STAT=MEDIAN to take effect, you must also specify the RESPONSE= option.

PERCENT
the percentage of the response or category variable.

The percentage is calculated as follows:

- If you specify the RESPONSE= option, PERCENT calculates the percentage of the sum of the response variable.
- If you do not specify the RESPONSE= option, PERCENT calculates the percentage of the frequency of the category variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the PCTLEVEL= option in the PROC SGPLOT statement.

You can use the PCTNDEC= option in the SGPLOT procedure statement to control the number of decimals to be used when calculating the percent values.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level are zero.

SUM
the sum of the response variable.

This is the default value when you specify the RESPONSE= option.

Interaction For this value to take effect, you must also specify the RESPONSE= option.

Defaults SUM when you also specify the RESPONSE= option.

FREQ when do not specify the RESPONSE= option.

Restriction If you do not also specify the RESPONSE= option, then only the FREQ or PERCENT statistic is calculated (FREQ is the default). If you specify RESPONSE=, then you can use any of the statistics.

Interaction When the graph is generated, the statistic is appended to the variable name in the axis label and the legend (if it is created). However, if a label has been assigned to the variable, then the label appears in the axis label and legend instead of the statistic.

STATLABEL | NOSTATLABEL
specifies whether the response variable statistic is displayed in the axis and legend labels. STATLABEL forces the statistic to be displayed. NOSTATLABEL removes the statistic from the axis and legend labels.

Normally, the procedure displays the statistic along with the name of the response variable. However, when a custom label is assigned to the response variable, the procedure does not display the statistic. In each case, you can control whether the statistic is displayed.

Defaults The statistic is displayed for the response variable.

When a custom label is assigned to the response variable, the statistic is not displayed.

Interactions This option has no effect unless the RESPONSE= option is specified.

This option has no effect if you specify the axis label using the LABEL= option in an AXIS statement.

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Notes When TIP= is not specified, the category, RESPONSE, and GROUP variables are displayed, if applicable, when you hover over a line or marker. The category and upper limit or lower limit variables are displayed when you hover over an error bar or error bar cap.

When TIP= is specified, the specified variables are displayed when you hover over a line or marker. No variables are displayed when you hover over an error bar or error bar cap.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

If your categorical chart is overlaid with other categorical charts, each chart can specify its own URL variable. The URL variable specified for each chart overlay is honored. This capability affects the following charts: DOT, HBAR, HLINE, VBAR, VLINE.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

WEIGHT=*numeric-variable*
specifies a variable that contains values to be used as weights for the calculations. Each observation is weighted by the value of the specified numeric variable.

Requirement The values of the weight variable must be greater than zero. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Interaction If your plot is overlaid with other categorization plots that also specify WEIGHT=, then the first WEIGHT variable that you specified is used for all of the plots.

X2AXIS
assigns the category variable to the secondary (top) horizontal axis.

Y2AXIS
assigns the response variable to the secondary (right) vertical axis.

WATERFALL Statement

Creates a waterfall chart computed from input data.

Interaction: A waterfall chart accumulates response values in data order. Any change in the order of the X-axis values from the data order can adversely affect the waterfall chart. For example, the X-axis value order can change when the waterfall chart is overlaid with other plots.

See: "Plot Compatibility" on page 1780 for information about overlaying this plot with other plot types

Example: "About Waterfall Charts" on page 64

Syntax

WATERFALL CATEGORY=*variable* RESPONSE=*numeric-variable* </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set.

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width.

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline.

COLORGROUP=*variable*
specifies a variable that is used to determine the visual attributes of the transaction bars.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
specifies a special effect to be used on the plot.

FILL

NOFILL
specifies whether the bars are filled.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency of the transaction bars.

OUTLINE

NOOUTLINE
specifies whether the bars have outlines.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*value*
specifies the degree of transparency for the plot.

Axis options

X2AXIS
assigns the variables that are assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS
assigns the variables that are assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

Data tip options

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Final bar options

FINALBARATTRS=*style-element* | (COLOR=*color*)
specifies the appearance of the final bar.

FINALBARTICKVALUE="*text-string*"
specifies the tick value used on the X axis for the final bar, if displayed.

Initial bar options

INITIALBARATTRS=*style-element* | (COLOR=*color*)
specifies the appearance of the initial bar, if displayed.

INITIALBARTICKVALUE="*text-string*"
specifies the tick value used on the X axis for the initial bar, if displayed.

INITIALBARVALUE=*number*
specifies the value of the initial bar.

Label options

DATALABEL
displays a label for each data point.

DATALABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option.

Plot options

MISSING
accepts a missing value as a valid category value.

STAT=MEAN | SUM
specifies the statistic for the response axis.

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

Plot reference options

NAME="*text-string*"
assigns a name to a plot statement.

Required Arguments

CATEGORY=*variable*
specifies the variable for the category values. Duplicated category values are summarized into a unique value. All values are treated as discrete.

RESPONSE=*numeric-variable*
specifies the variable for the response values.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

BARWIDTH=*numeric-value*
specifies the width of the bars as a ratio of the maximum possible width. The maximum width is equal to the distance between the center of each bar and the centers of the adjacent bars. Specify a value from 0.0 (narrowest) to 1.0 (widest).

For example, if you specify a width of `1`, then there is no space between the bars. If you specify a width of `.5`, then the width of the bars is equal to the space between the bars.

Default .8

BASELINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the baseline. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default The GraphAxisLines style element in the current style.

Notes The baseline is always drawn by default.

When *style-element* is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip To suppress the baseline, set the line thickness to 0 as follows:

```
baselineattrs=(thickness=0)
```

COLORGROUP=*variable*
specifies a variable that is used to determine the visual attributes of the transaction bars. The transaction bars for each group value are automatically distinguished by different visual attributes.

Note When the COLORGROUP= option is specified with the COLORRESPONSE= option, the COLORGROUP= option is ignored.

Tips The visual attributes of the initial and final bars are controlled by the INITIALBARATTRS= and FINALBARATTRS= options.

The visual attributes of the connecting line is controlled by the GraphConnectingLine style element.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used to determine response colors.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The ThreeColorRamp style element

Requirement The COLORRESPONSE= option must also be specified. Otherwise, COLORMODEL= is ignored.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color ramp. The color ramp represents the range of unique response values and is displayed in a gradient legend. The colors are applied to the transaction bars in the waterfall chart.

Note When the COLORRESPONSE= option is specified with the COLORGROUP= option, the COLORGROUP= option is ignored.

Tips The color ramp is specified by the COLORMODEL= option.

When the bars are filled, the bar outline attributes are controlled by the GraphOutlines style element.

The visual attributes of the initial and final bars are controlled by the INITIALBARATTRS= and FINALBARATTRS= options.

The visual attributes of the connecting line is controlled by the GraphConnectingLine style element.

See "GRADLEGEND Statement" on page 1003

"Using Gradient Color Legends" on page 1793

DATALABEL
displays a label for each data point. The values of the calculated response are used for the data labels.

DATALABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the labels in the plot when you use the DATALABEL= option. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Default The Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

Interaction This option has no effect unless the DATALABEL option is also specified.

Examples
```
DATALABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
DATALABELATTRS=GraphLabelText
```

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

specifies a special effect to be used on the plot. The data skin affects all filled bars. The effect that a data skin has on a filled area depends on the skin type, the graph style, and the color of the skinned element. Most of the skins work best with lighter colors over a medium to large filled area. Specify one of the following:

Table 10.59 *DATASKIN Options for Filled Areas*

NONE	CRISP	GLOSS
MATTE	PRESSED	SHEEN

Default NONE

Restriction The ODS GRAPHICS option DATASKINMAX= controls the maximum number of graphics elements that are skinned in a plot statement. The maximum number is 200 by default. When this limit is exceeded for a plot, the specified data skin is not applied. Use the DATASKINMAX= option to increase or decrease the maximum limit.

Interactions This option has no effect if the applied style uses a fill pattern and FILL is disabled for the style. If the style enables both FILL and FILLPATTERN, data skins work. For more information about fill patterns, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865. For instructions about viewing the contents of style templates, see "About Styles and Style Elements" on page 1841.

If you also specify NOFILL, then the data skin is applied to the outlines.

See "Using Data Skins" on page 1851

FILL | NOFILL

specifies whether the bars are filled. The FILL option shows the fill color for the bars. The NOFILL option hides the fill color for the bars.

Default FILL

Interactions Specifying FILL also hides the outlines.

If NOFILL and NOOUTLINE are both specified, then both options are ignored.

FILLATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill color and transparency of the transaction bars. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Defaults For ungrouped data, the Color attribute of the GraphDataDefault style element in the current style.

For grouped data, the Color attribute of a GraphData1–GraphData*n* style element in the current style.

The fill is completely opaque (TRANSPARENCY=0).

Interaction This option has no effect if you also specify the NOFILL option.

Tip Use the INITIALBARATTRS= option to control the appearance of the initial bar and the FINALBARATTRS= option to control the appearance of the final bar.

Examples Specify a style element:
```
FILLATTRS=GraphData3
```

Specify a color name and transparency:
```
FILLATTRS=(COLOR="steelblue" transparency=0.5)
```

FINALBARATTRS=*style-element* | (COLOR=*color*)
specifies the appearance of the final bar. You can specify the fill color by using the COLOR= suboption. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. You can specify the fill color and the appearance of the final bar outline by using a style element.

Default The GraphFinal style element in the current style.

Interaction This option has no effect if you specify the NOFILL option.

FINALBARTICKVALUE="*text-string*"
specifies the tick value used on the X axis for the final bar, if displayed.

Default "Final"

INITIALBARATTRS=*style-element* | (COLOR=*color*)
specifies the appearance of the initial bar, if displayed. You can specify the fill color by using the COLOR= suboption. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808. You can specify the fill color and the appearance of the final bar outline by using a style element.

Default The GraphInitial style element in the current style.

Interactions This option has no effect if you specify the NOFILL option.

This option has no effect unless INITIALBARVALUE is also specified.

INITIALBARTICKVALUE="*text-string*"
specifies the tick value used on the X axis for the initial bar, if displayed.

Default "Initial"

Interaction This option has no effect unless INITIALBARVALUE is also specified.

INITIALBARVALUE=*number*
specifies the value of the initial bar. When specified, the first transaction bar to be drawn starts at that initial value instead of zero.

Default If the INITIALBARVALUE option is not specified, the initial bar is not included in the chart and the first transaction bar is drawn from zero.

MISSING
accepts a missing value as a valid category value. By default, observations with missing values are ignored. If more than one chart is specified in the procedure, the MISSING option affects the calculations for all of the charts.

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

OUTLINE | NOOUTLINE
specifies whether the bars have outlines. The OUTLINE option shows the outlines. The NOOUTLINE option hides the outlines.

Default OUTLINE

Interactions Specifying OUTLINE also hides the fill color, unless FILL is also specified..

If NOOUTLINE and NOFILL are both specified, then both options are ignored.

Tips For ungrouped data, the outline appearance is controlled by the GraphOutlines style element. If COLORRESPONSE= is specified, the outline color is controlled by the color ramp that is specified by the COLORMODEL= option.

For grouped data, if only outlines are displayed, the outline thickness is controlled by the GraphOutlines style element, and the color and line style are controlled by a GraphData1–GraphData*n*

style element. If outlines and fill are displayed, the outline thickness and line style are controlled by the GraphOutlines style element, and the color is controlled by a GraphData1–GraphData*n* style element.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

STAT=MEAN | SUM
specifies the statistic for the response axis. Specify one of the following statistics:

MEAN
the mean of the response variable.

SUM
the sum of the response variable.

Default SUM

TIP=(*variable-list*) | NONE
specifies the data tip information to be displayed when the cursor is positioned over the graphics element.

(*variable-list*)
a space-separated list of variables enclosed in parentheses.

Data tips are displayed using the data obtained from the specified variables.

NONE
suppresses the data tips from this plot.

Requirement You must specify the IMAGEMAP option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

Interaction This option replaces all of the information that is displayed by default.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Note A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F2.0

Tip Use the TIPLABEL= option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a custom label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option. When AUTO is used, the label is obtained from the variable.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*value*
specifies the degree of transparency for the plot. The transparency that you specify applies to all aspects of the plot statement.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies an HTML page to be displayed when parts of the plot are selected.

character-variable
specifies a variable that contains a valid HTML page reference (HREF) for each plot element that is to have an active link.

Example https://www.sas.com/en_us/home.html

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP;
```

For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Tip Use the DRILLTARGET= option in the ODS GRAPHICS statement in order to specify the window that displays the drill-down output.

X2AXIS
assigns the variables that are assigned to the primary (bottom) horizontal axis to the secondary (top) horizontal axis.

Y2AXIS
assigns the variables that are assigned to the primary (left) vertical axis to the secondary (right) vertical axis.

XAXIS Statement

Specifies the axis options for the X axis.

Notes: All XAXIS options should be included in a single XAXIS statement. Otherwise, some axis options might not be applied.

You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

XAXIS *options*;

Summary of Optional Arguments

Appearance options

COLORBANDS=NONE | EVEN | ODD
: specifies the display of alternating wall-color bands corresponding to the discrete axis bins.

COLORBANDSATTRS=*style-element*<(***options***)> | (*options*)
: specifies the fill appearance of the color band.

DISPLAY=ALL | NONE | (*options*)
: specifies which features of the axis are displayed.

GRID
: creates grid lines at each tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the grid lines.

MINORGRID
: creates grid lines at each minor tick on the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the minor grid lines.

TICKSTYLE=OUTSIDE | INSIDE | ACROSS | INBETWEEN
: specifies the placement of the major tick marks in relation to the axis line.

Axis options

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED
: specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis.

INTEGER
: specifies that only integers are used for tick mark values.

INTERVAL=*time-interval*
: specifies the tick interval for a time axis.

INTERVALMULT=*positive-integer*
: specifies a multiplier to apply to the time interval that is in effect for the axis.

LOGBASE=2 | 10 | e
: specifies the base value for the logarithmic scale.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
: specifies how to scale and format the values for the major tick marks for logarithmic axes.

LOGVTYPE=EXPANDED | EXPONENT
: specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options.

MAX=*numeric-value*
: specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

MIN=*numeric-value*
: specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

MINOR
adds minor tick marks to a linear, log, or time axis.

MINORCOUNT=*numeric-value*
specifies the number of minor tick marks for the axis.

MINORINTERVAL=*time-interval*
specifies the time interval between minor ticks.

NOTIMESPLIT
prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX=*numeric-value*
specifies an offset that follows the highest data value on the axis.

OFFSETMIN=*numeric-value*
specifies an offset that precedes the lowest data value on the axis.

RANGES=(*start–end <start2–end2 startN–endN ...>*)
specifies the ranges for a broken axis.

REFTICKS

REFTICKS=(*options*)
adds tick marks to the axis that is opposite from the specified axis.

REVERSE
specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX=*numeric-value*
Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN=*numeric-value*
Specifies a threshold for displaying one more tick mark at the low end of the axis.

TYPE=DISCRETE | LINEAR | LOG | TIME
specifies the type of axis.

Text options

FITPOLICY=*policy-value*
specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally.

LABEL="*text-string*"
specifies a label for the axis.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the axis labels.

LABELPOS=CENTER | DATACENTER | LEFT | RIGHT
specifies the position of the axis label.

SPLITCHAR="*character-list*"
splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the axis tick value labels.

VALUES=(*values-list*) | (*"string-list"*)
specifies the values for the ticks on an axis.

VALUESDISPLAY=
specifies the text that is to be displayed for the tick values that are defined in the VALUES= option.

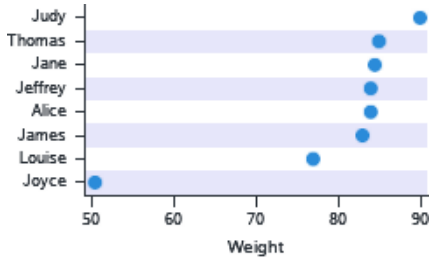
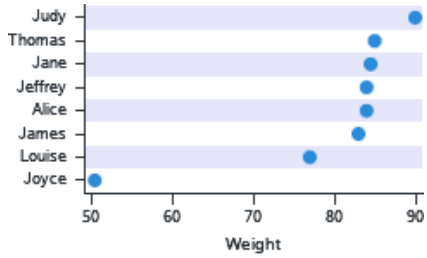
VALUESFORMAT=DATA | *SAS-format*
specifies how to format the values for major tick marks.

VALUESHINT
specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option.

VALUESROTATE=DIAGONAL | DIAGONAL2 | VERTICAL
specifies how the tick values are rotated on the axis.

Optional Arguments

COLORBANDS=NONE | EVEN | ODD
specifies the display of alternating wall-color bands corresponding to the discrete axis bins. The following images show the results of ODD and EVEN settings:

COLORBANDS=ODD	COLORBANDS=EVEN
	

Default NONE

Restriction This option applies to discrete axes only.

Interaction Specifying this option for more than one axis in the graph might have unexpected results. The order in which color bands are drawn might not match the order in which the axis options are specified.

Note The full width of a color band is the distance between midpoints. When no axis offsets are specified, the first band begins at one-half of the midpoint distance, and the last band ends at one-half of the midpoint distance. When axis offsets are specified, the first and last color bands on the axis might extend into their adjacent offsets by as much as half the color-band width.

Tip Use the COLORBANDSATTRS= option to customize the color bands.

COLORBANDSATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill appearance of the color band. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interaction This option has no effect unless COLORBANDS= is also specified.

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED
specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis. Specify one of the following values:

DATA
places the values in the order in which they appear in the data.

Note: This value is not supported with data that is stored in a table.

FORMATTED
sorts the formatted values in ascending character order.

UNFORMATTED
sorts the unformatted values in ascending character order.

Default UNFORMATTED

DISPLAY=ALL | NONE | (*options*)
specifies which features of the axis are displayed.

ALL
displays all of the features of the axis.

NONE
specifies that none of the features of the axis are displayed.

(*options*)
specifies one or more axis features to hide.

Specify one or more of the following:

NOLABEL
hides the axis label

NOLINE
hides the axis line

Tips This value hides the axis line but has no effect on the graph's wall border, part of which occupies the same area as the axis line. To hide the wall border, specify NOBORDER in the PROC SGPLOTstatement.

This value has no effect on baselines. For plots that support a baseline, such as bar charts and needle plots, you might need to suppress the baseline. In the plot statement, use the BASELINEATTRS= option to set the line thickness to 0.

NOTICKS
hides the tick marks on the axis

NOVALUES
hides the tick mark values on the axis

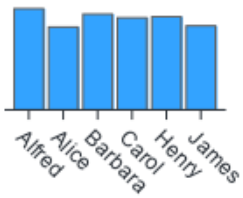
Default ALL

Example DISPLAY=(NOTICKS NOVALUES)

FITPOLICY=*policy-value*
specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally. Select one of the following values:

NONE
does not split the values.

ROTATE
rotates the value text 45 degrees.



> **TIP** Use VALUESROTATE= to specify how the tick values are rotated on the axis.

Note: With a time axis, you must specify NOTIMESPLIT for this option to have any effect.

ROTATETHIN
attempts to use ROTATE, and then THIN to fit the values.

Note: With a time axis, you must specify NOTIMESPLIT for this option to have any effect.

SPLIT
splits the values at the character or characters specified in the SPLITCHAR= option.

No split occurs at split characters that occur where a split is not needed. In that case, the split character is displayed with the text value. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Interaction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

SPLITALWAYS
always splits the values at the character or characters specified in the SPLITCHAR= option. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Interaction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

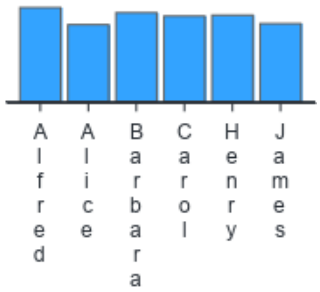
Example "Fit Policies for Axes" on page 1803

SPLITROTATE
attempts to use SPLIT, and then ROTATE to fit the values.

This is the default for discrete axes.

Note: This option has no effect on time axes.

STACKEDALWAYS
stacks the values vertically regardless of tick value collision.



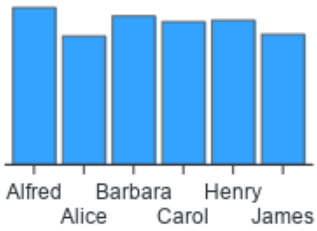
Interaction This option works only with a discrete axis.

STACKEDALWAYSTHIN
stacks the values vertically regardless of tick value collision.

If a collision is detected after stacking, then THIN as needed.

Interaction This option works only with a discrete axis.

STAGGER
shifts the values up and down.



Interaction When used with a time axis, this option has no effect unless you also specify NOTIMESPLIT in the axis statement.

STAGGERROTATE
attempts to use STAGGER, and then ROTATE to fit the values.

Interaction When used with a time axis, this option has no effect unless you also specify NOTIMESPLIT in the axis statement.

STAGGERTHIN
attempts to use STAGGER, and then THIN to fit the values.

Interaction When used with a time axis, this option has no effect unless you also specify NOTIMESPLIT in the axis statement.

THIN
removes some of the values from the axis.

This is the default for linear and time axes.

Defaults SPLITROTATE for discrete axes.

THIN for linear and time axes.

Restriction This option does not affect logarithmic axes.

See "Fit Policies for Axes" on page 1803

GRID
creates grid lines at each tick on the axis.

Interaction Grid lines are not displayed when you specify the COLORBANDS= option. The color bands take the place of grid lines.

Tip You can specify the MINORGRID option to create grid lines at each minor tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless GRID is also specified.

Examples `GRIDATTRS=(color=green pattern=longdash thickness=2)`

Here is an example that specifies a style element:
`GRIDATTRS=GraphAxisLines`

INTEGER
specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL=*time-interval*
specifies the tick interval for a time axis. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

Specify one of the following values:

Table 10.60 *Time Intervals*

INTERVAL	Unit	Tick interval	Default tick value format
AUTO	DATE, TIME, or DATETIME	automatically chosen	automatically chosen
SECOND	TIME or DATETIME	second	TIME8.
MINUTE	TIME or DATETIME	minute	TIME8.
HOUR	TIME or DATETIME	hour	TIME8.
DAY	DATE or DATETIME	day	DATE9.
TENDAY	DATE or DATETIME	10 days	DATE9.
WEEK	DATE or DATETIME	7 days	DATE9.
SEMIMONTH	DATE or DATETIME	1st and 16th of each month	DATE9.

INTERVAL	Unit	Tick interval	Default tick value format
MONTH	DATE or DATETIME	month	MONYY7.
QUARTER	DATE or DATETIME	3 months	YYQC6.
SEMIYEAR	DATE or DATETIME	6 months	MONYY7.
YEAR	DATE or DATETIME	year	YEAR4.

Default AUTO

Tip If unwanted thinning of tick marks occurs when you specify INTERVAL=, try using minor intervals instead. For example, you might specify MINOR MINORINTERVAL=MONTH instead of INTERVAL=MONTH. As a result, in the example monthly tick marks are displayed even though the major values are thinned.

INTERVALMULT=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the axis.

Alias INTERVALMULTIPLIER

Default 1

Restriction This option applies to time axes only.

Interaction This option is ignored if VALUES=*values-list* is used.

Tip Use the INTERVAL= option to specify a different time interval.

Examples To specify 3-month intervals:

```
INTERVAL=MONTH INTERVALMULT=3
```

To specify 10-year intervals:

```
INTERVAL=YEAR INTERVALMULT=10
```

LABEL="*text-string*"
specifies a label for the axis.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LABELPOS=CENTER | DATACENTER | LEFT | RIGHT
specifies the position of the axis label.

CENTER
centers the axis label in the axis area (including any offsets).

The label is positioned below the tick values for the X axis or above axis values for the X2 axis.

DATACENTER
centers the axis label in the axis tick display area (excluding any offsets).

The label is positioned below the tick values for the X axis or above the axis values for the X2 axis.

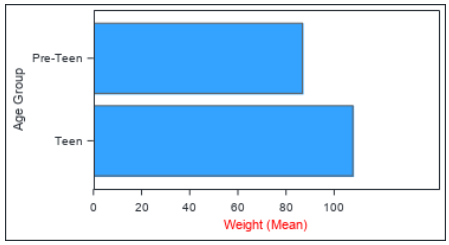
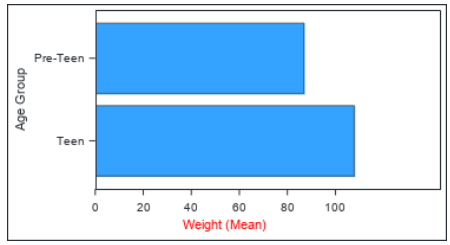
LEFT
positions the label to the left of the axis area.

The label is centered vertically in the axis area.

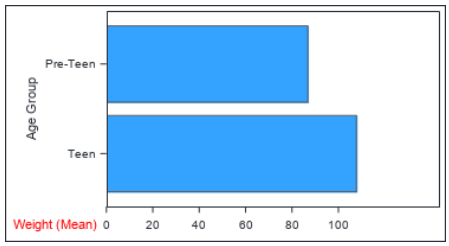
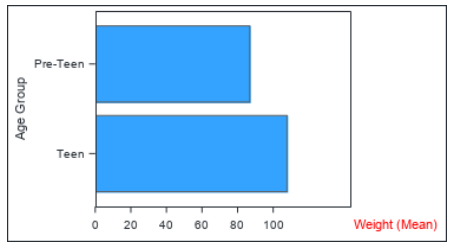
RIGHT
positions the label to the right of the axis area.

The label is centered vertically in the axis area.

The following figure shows the CENTER and DATACENTER positions for the red X-axis label “Weight (Mean).” An axis offset is applied to the maximum end of the axis in order to demonstrate the difference between CENTER and DATACENTER. CENTER centers the labels on the entire axis area, including the offset. DATACENTER centers the labels on the tick display areas, which does not include the offset.

CENTER	DATACENTER
	

The next figure shows the LEFT and RIGHT positions for the same axis labels.

LEFT	RIGHT
	

Default CENTER

LOGBASE=2 | 10 | e
specifies the base value for the logarithmic scale.

Default 10

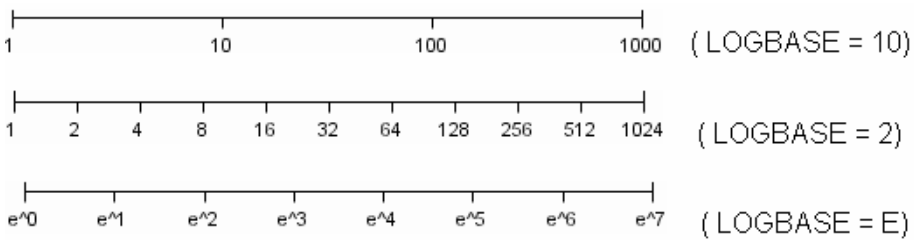
Interaction This option has no effect unless you also specify TYPE=LOG.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND
places the tick marks at integer powers of the base.

For example, if you specified `LOGBASE=2`, the tick marks might be at 1, 2, 4, 8, 16. See Figure 10.32 on page 1535 .

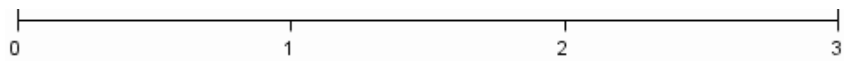
Figure 10.10 *Graph Axes with LOGEXPAND*



LOGEXPONENT
places the tick marks at integer powers of the base, but identifies the values by the exponent.

For example, if you specified `LOGBASE=10`, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See Figure 10.33 on page 1535 .

Figure 10.11 *An Axis with LOGEXPONENT*



LINEAR
places the tick marks at uniform linear intervals, but spaces them logarithmically.

In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See Figure 10.34 on page 1536 .

Figure 10.12 *An Axis with LINEAR*

Default LOGEXPAND

Interaction This option has no effect unless you also specify TYPE=LOG.

LOGVTYPE=EXPANDED | EXPONENT
specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options. This option enables you to choose your preferred way of specifying log-axis values regardless of the LOGSTYLE= option value.

Specify one of the following values:

EXPANDED
the values are interpreted as integer powers of the base (decimal numbers).

EXPONENT
the values are interpreted as integer exponents of the base.

Default EXPANDED

Interaction This option has no effect unless you also specify TYPE=LOG. You must also specify values for the VALUES= option or the MIN= and MAX= options or all of them.

Tip This option is particularly useful when the log axis is an odd base (such as base E) or the axis log style is EXPONENT.

Examples The following example specifies MIN= and MAX= as exponent values instead of expanded values on an expanded Base 10 log axis. This results in Y-axis tick values of 10, 100, 1000, 10000, and 100000.

```
yaxis type=log logbase=10 logstyle=logexpand
   logvtype=exponent
   min=1 max=5;
```

The following example specifies VALUES= as a list of expanded values instead of exponent values on an exponent Base 10 log axis. This results in X-axis tick values of 1, 2, 3, 4, and 5.

```
xaxis type=log logbase=10 logstyle=logexponent
   logvtype=expanded
   values=(10 100 1000 10000 100000);
```

MAX=*numeric-value*

specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES=option and you do not also specify the VALUESHINT option.

This option does not determine the maximum axis tick value displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the maximum value is expanded or interpreted as an exponent.

Tip The maximum axis tick value might differ from the MAX= value. The MAX= and MIN= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values. To display the MAX= value as the maximum tick value, use the VALUES= option.

MIN=*numeric-value*

specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

This option does not determine the minimum axis tick value displayed. The THRESHOLDMIN= value is used to determine the minimum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the minimum value is expanded or interpreted as an exponent.

Tip The minimum axis tick value might differ from the MIN= value. The MIN= and MAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values. To display the MIN= value as the minimum tick value, use the VALUES= option.

MINOR

adds minor tick marks to a linear, log, or time axis.

Restriction This option has no effect on discrete axes.

Interaction This option has no effect if you specify the VALUES= option.

Tip Use MINORCOUNT= to specify the number of tick marks.

MINORCOUNT=*numeric-value*
specifies the number of minor tick marks for the axis. This value determines the number of minor tick marks for each interval on the axis.

Restriction This option applies to linear and log axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

MINORGRID
creates grid lines at each minor tick on the axis.

Interaction This option has no effect unless GRID is also specified for the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the minor grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphMinorGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless MINORGRID is also specified.

Tip You can use GRIDATTRS= to change the appearance of the major grid lines.

Examples
```
MINORGRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:
```
MINORGRIDATTRS=GraphAxisLines
```

MINORINTERVAL=*time-interval*
specifies the time interval between minor ticks. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

For information about the intervals that you can select, see Table 10.139 on page 1532.

Default AUTO

Restriction This option applies to time axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

NOTIMESPLIT
prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction This option applies to time axes only.

OFFSETMAX=*numeric-value*
specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the maximum end of the axis to the highest data value or highest tick value, whichever is greater.

For a discrete axis, the offset is measured from the maximum end of the axis to the midpoint of the last category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the maximum end of the axis. The sum of the minimum and maximum offsets does not exceed 50% of the axis length.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See "Adjusting Axis Offsets" on page 1788

OFFSETMIN=*numeric-value*
specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the beginning of the axis to the lowest data value or lowest tick value, whichever is less.

For a discrete axis, the offset is measured from the beginning of the axis to the midpoint of the first category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the minimum end of the axis. The sum of the minimum and maximum offsets does not exceed 50% of the axis length.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See "Adjusting Axis Offsets" on page 1788

RANGES=(*start–end <start2–end2 startN–endN ...>*)
specifies the ranges for a broken axis.

start
specifies the start of a range.

Start can be one of the following:

- a number (linear axis only).
- the keyword MIN specifies the minimum data value.

- a SAS time, date, or date-time constant (time axis only).

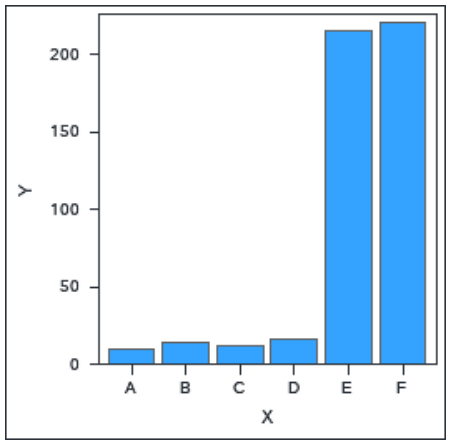
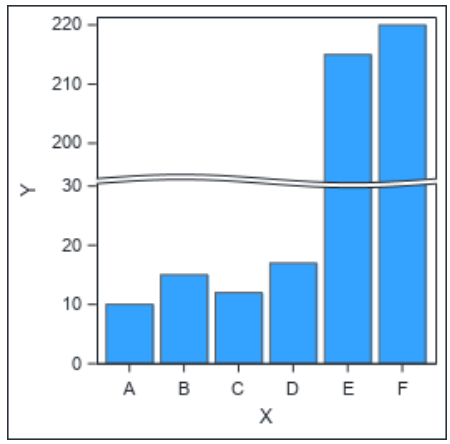
end

specifies the end of a range.

End can be one of the following:

- a number (linear axis only).
- the keyword MAX specifies the maximum data value.
- a SAS time, date, or date-time constant (time axis only).

The following figure shows a linear axis, broken into ranges 0–30 and 195–220. Although the figure shows the split range on a vertical axis, the same concept applies to a horizontal axis.

RANGES Not Specified	RANGES Specified
	

As shown in the figure, break lines are drawn to indicate the break in the axis.

Restrictions This option is valid for linear and time axes only.

Only one axis can be broken. If this option is specified for both axes, then it is honored for the vertical axis and ignored for the horizontal axis.

If this option is specified for both axes in the same direction (X and X2 or Y and Y2), then only the primary axis is broken.

When plots are associated with both the X and X2 axes or with both the Y and Y2 axes, neither axis can be broken.

Requirements All of the ranges must be enclosed in parentheses.

You must specify each range as a starting value, a hyphen, and an ending value. You must separate adjacent ranges with a space.

Interactions When this option is specified, axis options THRESHOLDMIN=, THRESHOLDMAX=, MIN=, and MAX= are ignored.

When this option is specified, data tips are not displayed, and the plot URL= option is ignored.

When data labels are used in the graph, the data label font size might be reduced in order to avoid overlapping labels and markers. When a range is specified, the data label font size is not scaled during label placement.

Notes When this option is specified, data-clipping might occur for the following graphics elements: plot markers and marker characters, box plot outlier markers, fixed-position data labels, needle fringe plots, reference lines and drop lines on the broken axis, axis tables, and relative bubble plots.

Curve label positions are based on the non-broken axis data range. When curve labels are specified with a broken axis, the curve label positions might not be ideal.

Tip To control the axis break symbol, use the AXISBREAK= option in the STYLEATTR statement.

Examples
```
ranges=(10-500 1000-5000 10000-50000)

ranges=('01Jan2001'd-'01May2003'd '01Jan2005'd-'01Oct2005'd)
```

REFTICKS
REFTICKS=(*options*)
adds tick marks to the axis that is opposite from the specified axis. You can also specify *options*:

LABEL
in addition to the tick marks, displays the axis label.

NOTICKS
suppresses the tick marks.

VALUES
in addition to the tick marks, displays the values that are represented by the tick marks.

Note This option has no effect if the target axis already contains data.

REVERSE
specifies that the tick values are displayed in reverse (descending) order.

SPLITCHAR="*character-list*"
splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect. When FITPOLICY=SPLIT is in effect, the text is split on occurrences of the specified split character or characters where it is required to fit the tick marks. When FITPOLICY=SPLITALWAYS is in effect, the text is split on all occurrences of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Restriction This option has no effect unless the axis is discrete.

Interactions This option has no effect unless FITPOLICY= is specified as either SPLIT, SPLITALWAYS, or SPLITROTATE. SPLITROTATE is the default for discrete X axes.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Default CENTER

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THRESHOLDMAX=*numeric-value*
Specifies a threshold for displaying one more tick mark at the high end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

THRESHOLDMIN=*numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

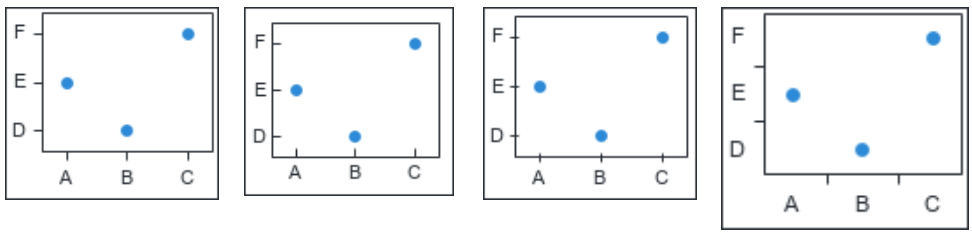
Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

TICKSTYLE=OUTSIDE | INSIDE | ACROSS | INBETWEEN

specifies the placement of the major tick marks in relation to the axis line. The figure shows the tick display for each value.

OUTSIDE	INSIDE	ACROSS	INBETWEEN



OUTSIDE

displays the major tick marks outside of the axis frame.

INSIDE
displays the major tick marks inside the axis frame.

ACROSS
displays the major tick marks across the axis line.

INBETWEEN
displays the major tick marks between the discrete data values and outside of the axis frame.

Interaction This value is available only with discrete axes. The option is ignored if specified with other axis types.

Default OUTSIDE

Restriction This option applies to the major tick marks only. It does not affect the minor tick marks.

Interaction This option has effect only if the DISPLAY= option setting includes tick marks, which it does by default.

Note This option has no effect on the placement of the tick values, which are always outside the axis frame.

Tip You can use the SG annotation facility to draw inside minor tick marks in your graph. See Part 7, "SG Annotation," on page 1933. The post Creating an axis in GTL with both major and minor tick marks shown inside on the Graphics Programming community page provides an example for the SGPLOT procedure.

TYPE=DISCRETE | LINEAR | LOG | TIME
specifies the type of axis. Specify one of the following values:

DISCRETE
specifies an axis with discrete values.

If a character variable is assigned to an axis, then the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable. Bar charts support a linear category axis.

LINEAR
specifies a linear scale for the axis.

This is the default axis type for numeric variables, except when the data is discrete, or when the numeric variable has a date or time format.

LOG
specifies a logarithmic scale for the axis.

This axis type is never a default.

Restriction A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

Interactions Use the LOGSTYLE= option to specify the scale and format for the tick values.

Use the LOGBASE= option to specify the base value.

Use the LOGVTYPE= option to specify how the values that are provided in the VALUES= option and the MIN= and MAX= options are interpreted.

TIME
specifies a time scale for the axis.

If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

For ungrouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

VALUES=(*values-list*) | ("*string-list*")
specifies the values for the ticks on an axis. The syntax for this option varies depending on the type of axis.

- VALUES=(*values-list*) specifies tick values for linear, time, and logarithmic axes.
- VALUES=(*"string-list"*) specifies tick values for discrete axes. The values can be character or numeric.

values-list
specifies tick values for linear, time, and logarithmic axes.

Values in *values-list* can be specified as a simple list of values in the following form:

value <...*value-n*>

The values must be listed in either ascending or descending order. The values can also be specified as a range of values of a specific increment in the following form:

value-1 TO *value-2* BY *increment-value*

By default the increment value is 1. You can use a negative integer for increment to specify a value list in descending order.

You can use both forms when specifying values:

<*value ... value-n* >*value-1* TO *value-2* BY *increment-value* <*value ... value-n*>

In all forms, multiple *n* values can be separated by blanks or commas. Here are some examples:

- Place tick marks at 2, 4, and 6 on a linear axis:

```
values=(2 4 6)
```

- Place tick marks at 6, 4, and 2 on a linear axis:

```
values=(6,4,2)
```

- Place tick marks at 25MAY08, 04JUL08, and 23AUG08 on a time axis:

```
values=("25MAY08"d "04JUL08"d "23AUG08"d)
```

- Place tick marks from 2 to 10 at an increment of 2 on a linear axis:

```
values=(2 to 10 by 2)
```

- Place tick marks from 50 to 10 at an increment of -5 on a linear axis:

```
values=(50 to 10 by -5)
```

If the specified range is not evenly divisible by the increment value, the highest value displayed on the axis is the last incremental value below the ending value for the range. For example, this value list produces a maximum axis value of 9:

```
values=(0 to 10 by 3)
```

Restrictions This option has no effect on discrete axes.

If your VALUES= option creates more than 1000 values, then the option has no effect.

Interactions For logarithmic axes, use the LOGVTYPE option to control whether the values are expanded or interpreted as exponents.

If a custom format is applied to the value, the raw value is plotted first and then formatted.

Tips Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

For graphs such as histograms, the final bin might be truncated at the maximum data value. If you need to expand the axis in order to show all of the data, use the VALUESHINT option.

"string-list"

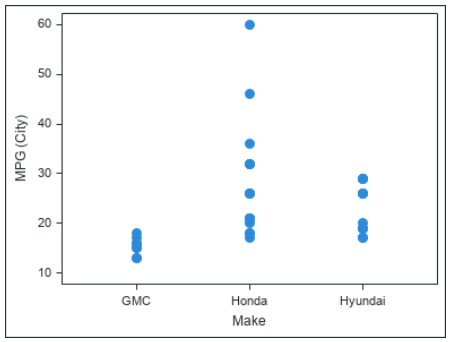
specifies a space-separated list of string values enclosed in parentheses for a discrete axis.

Each value in the list must be enclosed in quotation marks. Numeric values must also be enclosed in quotation marks.

Only the tick values that are included in the string list are displayed on the axis. The values are displayed in the order in which they are listed. The data values that are not in the list are dropped. The list can be a subset of the data values. It can also contain values that are not included in the actual data. A tick value that is not included in the data appears on the axis, but no data is represented at its tick mark.

The following example specifies the values for an X axis. Similar values can be provided for the Y axis.

Table 10.61 *Axis with Specified Character Values*




```
proc sgplot data=sashelp.cars;
xaxis values=("GMC" "Honda" "Hyundai")
   offsetmin=0.2 offsetmax=0.2;
scatter x=make y=mpg_city;
run;
```

Restrictions

This option has no effect on linear, time, or logarithmic axes.

Each value must be enclosed in quotation marks and separated from adjacent values by a blank space.

Notes

If the string list contains duplicate values, the first occurrence of the duplicated value in the list is honored while the remaining instances are ignored.

The axis data can be character or numeric.

If a numeric variable has an associated format, the specified values must use the same format.

Tip

Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

Examples

The following example specifies the axis tick values Sedan, Sports, Wagon, and SUV:

```
values=("Sedan" "Sports" "Wagon" "SUV")
```

The following example specifies the axis tick values 10, 20, 30, and 40:

```
values=("10" "20" "30" "40")
```

The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

> **TIP** Along the axis, numeric tick values are arranged in ascending order while character values are arranged in the order in which they are used in the graph. In some cases, the resulting tick-value order might not be desirable, especially if the graph consists of multiple plots or if the data contains missing values. This option is useful in that case. You can use this option to set the order of the axis tick values.

VALUESDISPLAY=
specifies the text that is to be displayed for the tick values that are defined in the VALUES= option. The list of values must be enclosed in parentheses. Each value must be enclosed in quotation marks and separated from adjacent values by a blank space. Numeric values must also be enclosed in quotation marks.

Restriction This option applies only to discrete and linear axes.

Interaction This option should be used with the VALUES= option. The number of items in the list for this option should equal the number of items in the list for the VALUES= option.

Example The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

VALUESFORMAT=DATA | *SAS-format*
specifies how to format the values for major tick marks. This option supports discrete and logarithmic axes.

DATA
uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

SAS-format
specifies a format to apply to the major tick values.

Restriction This option currently honors most, but not every, SAS format. For more information, see "Format Support in ODS Graphics" in *SAS Graph Template Language: Reference*.

Notes If you specify a format that significantly reduces precision, then, because of tick-value rounding, the plot data elements might not align properly with the axis tick values. In that case, specify a tick-value format with a higher precision.

A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F5.1

Alias TICKVALUEFORMAT

Restriction For discrete axes, only character formats are supported.

Interactions This option is ignored when LOGSTYLE=LOGEXPONENT.

When LOGSTYLE =LOGEXPAND, this option is honored for the base 10 and base 2 logarithmic scales, and is ignored for the base E scale.

When LOGSTYLE =LINEAR, this option is honored for the base 10, base 2, and base E logarithmic scales.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option. Tick values specified by the VALUES= option are treated as suggestions for the tick values that actually appear, so that the actual data range of the axis is preserved. If the *values-list* contains values that are less than the data minimum, or greater than the data maximum, then those values are ignored.

Interaction This option has no effect unless you also specify the VALUES= option.

VALUESROTATE=DIAGONAL | DIAGONAL2 | VERTICAL

specifies how the tick values are rotated on the axis.

DIAGONAL

rotates the tick values diagonally.

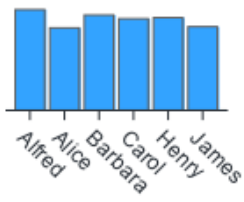
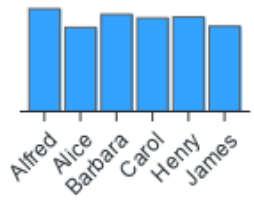
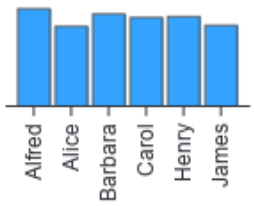
DIAGONAL2

rotates the tick values to a reverse diagonal angle.

VERTICAL

rotates the tick values vertically.

The values are always drawn from bottom to top.

DIAGONAL	DIAGONAL2	VERTICAL
		

Default DIAGONAL

Interactions This option works through the axis fit policy. If there are no tick collisions, no rotation occurs.

With a time axis, you must specify NOTIMESPLIT for this option to have any effect.

X2AXIS Statement

Specifies the axis options for the X2 axis, which is displayed on the top side of the graph.

Notes:

All X2AXIS options should be included in a single X2AXIS statement. Otherwise, some axis options might not be applied.

You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

X2AXIS *options*;

Summary of Optional Arguments

Appearance options

COLORBANDS=NONE | EVEN | ODD
: specifies the display of alternating wall-color bands corresponding to the discrete axis bins.

COLORBANDSATTRS=*style-element*<(***options***)> | (*options*)
: specifies the fill appearance of the color band.

DISPLAY=ALL | NONE | (*options*)
: specifies which features of the axis are displayed.

GRID
: creates grid lines at each tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the grid lines.

MINORGRID
: creates grid lines at each minor tick on the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the minor grid lines.

TICKSTYLE=OUTSIDE | INSIDE | ACROSS | INBETWEEN
: specifies the placement of the major tick marks in relation to the axis line.

Axis options

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED
: specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis.

INTEGER
: specifies that only integers are used for tick mark values.

INTERVAL=*time-interval*
: specifies the tick interval for a time axis.

INTERVALMULT=*positive-integer*
: specifies a multiplier to apply to the time interval that is in effect for the axis.

LOGBASE=2 | 10 | e
: specifies the base value for the logarithmic scale.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
: specifies how to scale and format the values for the major tick marks for logarithmic axes.

LOGVTYPE=EXPANDED | EXPONENT
: specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options.

MAX=*numeric-value*
: specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

MIN=*numeric-value*
: specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

MINOR
: adds minor tick marks to a linear, log, or time axis.

MINORCOUNT=*numeric-value*
: specifies the number of minor tick marks for the axis.

MINORINTERVAL=*time-interval*
: specifies the time interval between minor ticks.

NOTIMESPLIT
: prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX=*numeric-value*
: specifies an offset that follows the highest data value on the axis.

OFFSETMIN=*numeric-value*
: specifies an offset that precedes the lowest data value on the axis.

RANGES=(*start–end <start2–end2 startN–endN ...>*)
: specifies the ranges for a broken axis.

REFTICKS

REFTICKS=(*options*)
: adds tick marks to the axis that is opposite from the specified axis.

REVERSE
: specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX=*numeric-value*
: Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN=*numeric-value*
: Specifies a threshold for displaying one more tick mark at the low end of the axis.

TYPE=DISCRETE | LINEAR | LOG | TIME
: specifies the type of axis.

Text options

FITPOLICY=*policy-value*
: specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally.

LABEL="*text-string*"
: specifies a label for the axis.

LABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the axis labels.

LABELPOS=CENTER | DATACENTER | LEFT | RIGHT
: specifies the position of the axis label.

SPLITCHAR="*character-list*"
: splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect.

SPLITCHARNODROP
: specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the value text that is being split.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the axis tick value labels.

VALUES=(*values-list*) | ("*string-list*")
: specifies the values for the ticks on an axis.

VALUESDISPLAY=
: specifies the text that is to be displayed for the tick values that are defined in the VALUES= option.

VALUESFORMAT=DATA | *SAS-format*
: specifies how to format the values for major tick marks.

VALUESHINT
: specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option.

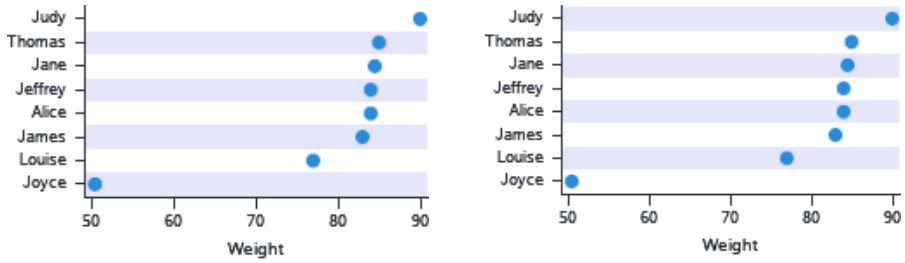
VALUESROTATE=DIAGONAL | DIAGONAL2 | VERTICAL
: specifies how the tick values are rotated on the axis.

Optional Arguments

COLORBANDS=NONE | EVEN | ODD

specifies the display of alternating wall-color bands corresponding to the discrete axis bins. The following images show the results of ODD and EVEN settings:

COLORBANDS=ODD	COLORBANDS=EVEN



Default NONE

Restriction This option applies to discrete axes only.

Interaction Specifying this option for more than one axis in the graph might have unexpected results. The order in which color bands are drawn might not match the order in which the axis options are specified.

Note The full width of a color band is the distance between midpoints. When no axis offsets are specified, the first band begins at one-half of the midpoint distance, and the last band ends at one-half of the midpoint distance. When axis offsets are specified, the first and last color bands on the axis might extend into their adjacent offsets by as much as half the color-band width.

Tip Use the COLORBANDSATTRS= option to customize the color bands.

COLORBANDSATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill appearance of the color band. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interaction This option has no effect unless COLORBANDS= is also specified.

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED
specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis. Specify one of the following values:

DATA
places the values in the order in which they appear in the data.

Note: This value is not supported with data that is stored in a table.

FORMATTED
sorts the formatted values in ascending character order.

UNFORMATTED
sorts the unformatted values in ascending character order.

Default UNFORMATTED

DISPLAY=ALL | NONE | (*options*)
specifies which features of the axis are displayed.

ALL
displays all of the features of the axis.

NONE
specifies that none of the features of the axis are displayed.

(*options*)
specifies one or more axis features to hide.

Specify one or more of the following:

NOLABEL
hides the axis label

NOLINE
hides the axis line

Tips This value hides the axis line but has no effect on the graph's wall border, part of which occupies the same area as the axis line. To hide the wall border, specify NOBORDER in the PROC SGPLOTstatement.

This value has no effect on baselines. For plots that support a baseline, such as bar charts and needle plots, you might need to suppress the baseline. In the plot statement, use the BASELINEATTRS= option to set the line thickness to 0.

NOTICKS
hides the tick marks on the axis

NOVALUES
hides the tick mark values on the axis

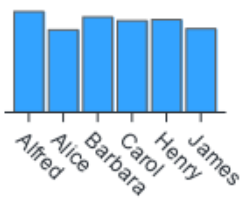
Default ALL

Example DISPLAY=(NOTICKS NOVALUES)

FITPOLICY=*policy-value*
specifies the method that is used to fit tick mark values on a horizontal axis when there is not enough room to draw them normally. Select one of the following values:

NONE
does not split the values.

ROTATE
rotates the value text 45 degrees.



TIP Use VALUESROTATE= to specify how the tick values are rotated on the axis.

Note: With a time axis, you must specify NOTIMESPLIT for this option to have any effect.

ROTATETHIN
attempts to use ROTATE, and then THIN to fit the values.

Note: With a time axis, you must specify NOTIMESPLIT for this option to have any effect.

SPLIT
splits the values at the character or characters specified in the SPLITCHAR= option.

No split occurs at split characters that occur where a split is not needed. In that case, the split character is displayed with the text value. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Interaction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

SPLITALWAYS
always splits the values at the character or characters specified in the SPLITCHAR= option. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Interaction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

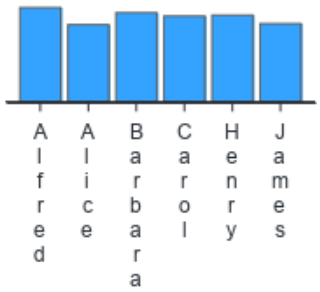
Example "Fit Policies for Axes" on page 1803

SPLITROTATE
attempts to use SPLIT, and then ROTATE to fit the values.

This is the default for discrete axes.

Note: This option has no effect on time axes.

STACKEDALWAYS
stacks the values vertically regardless of tick value collision.



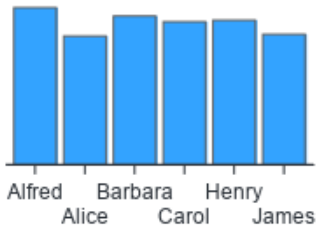
Interaction This option works only with a discrete axis.

STACKEDALWAYSTHIN
stacks the values vertically regardless of tick value collision.

If a collision is detected after stacking, then THIN as needed.

Interaction This option works only with a discrete axis.

STAGGER
shifts the values up and down.



Interaction When used with a time axis, this option has no effect unless you also specify NOTIMESPLIT in the axis statement.

STAGGERROTATE
attempts to use STAGGER, and then ROTATE to fit the values.

Interaction When used with a time axis, this option has no effect unless you also specify NOTIMESPLIT in the axis statement.

STAGGERTHIN
attempts to use STAGGER, and then THIN to fit the values.

Interaction When used with a time axis, this option has no effect unless you also specify NOTIMESPLIT in the axis statement.

THIN
removes some of the values from the axis.

This is the default for linear and time axes.

Defaults SPLITROTATE for discrete axes.

THIN for linear and time axes.

Restriction This option does not affect logarithmic axes.

See "Fit Policies for Axes" on page 1803

GRID
creates grid lines at each tick on the axis.

Interaction Grid lines are not displayed when you specify the COLORBANDS= option. The color bands take the place of grid lines.

Tip You can specify the MINORGRID option to create grid lines at each minor tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless GRID is also specified.

Examples
```
GRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:
```
GRIDATTRS=GraphAxisLines
```

INTEGER
specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL=*time-interval*
specifies the tick interval for a time axis. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

Specify one of the following values:

Table 10.62 *Time Intervals*

INTERVAL	Unit	Tick interval	Default tick value format
AUTO	DATE, TIME, or DATETIME	automatically chosen	automatically chosen
SECOND	TIME or DATETIME	second	TIME8.

INTERVAL	Unit	Tick interval	Default tick value format
MINUTE	TIME or DATETIME	minute	TIME8.
HOUR	TIME or DATETIME	hour	TIME8.
DAY	DATE or DATETIME	day	DATE9.
TENDAY	DATE or DATETIME	10 days	DATE9.
WEEK	DATE or DATETIME	7 days	DATE9.
SEMIMONTH	DATE or DATETIME	1st and 16th of each month	DATE9.
MONTH	DATE or DATETIME	month	MONYY7.
QUARTER	DATE or DATETIME	3 months	YYQC6.
SEMIYEAR	DATE or DATETIME	6 months	MONYY7.
YEAR	DATE or DATETIME	year	YEAR4.

Default AUTO

Tip If unwanted thinning of tick marks occurs when you specify INTERVAL=, try using minor intervals instead. For example, you might specify MINOR MINORINTERVAL=MONTH instead of INTERVAL=MONTH. As a result, in the example monthly tick marks are displayed even though the major values are thinned.

INTERVALMULT=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the axis.

Alias INTERVALMULTIPLIER

Default 1

Restriction This option applies to time axes only.

Interaction This option is ignored if VALUES=*values-list* is used.

Tip Use the INTERVAL= option to specify a different time interval.

Examples To specify 3-month intervals:

```
INTERVAL=MONTH INTERVALMULT=3
```

To specify 10-year intervals:

```
INTERVAL=YEAR INTERVALMULT=10
```

LABEL="*text-string*"
specifies a label for the axis.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples

```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:

```
LABELATTRS=GraphTitleText
```

LABELPOS=CENTER | DATACENTER | LEFT | RIGHT
specifies the position of the axis label.

CENTER
centers the axis label in the axis area (including any offsets).

The label is positioned below the tick values for the X axis or above axis values for the X2 axis.

DATACENTER
centers the axis label in the axis tick display area (excluding any offsets).

The label is positioned below the tick values for the X axis or above the axis values for the X2 axis.

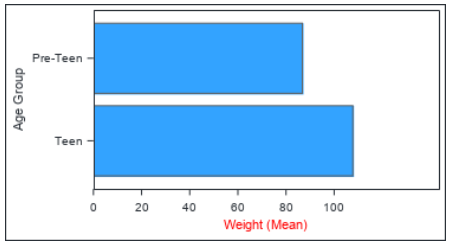
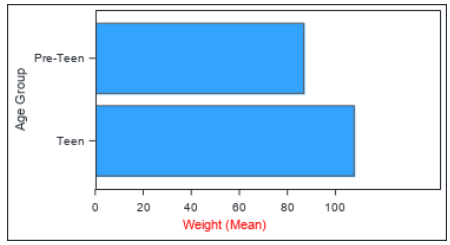
LEFT
positions the label to the left of the axis area.

The label is centered vertically in the axis area.

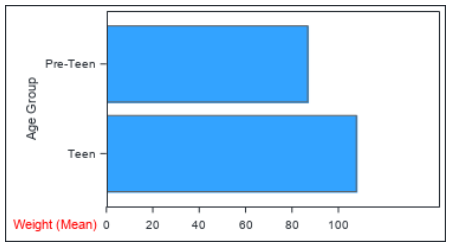
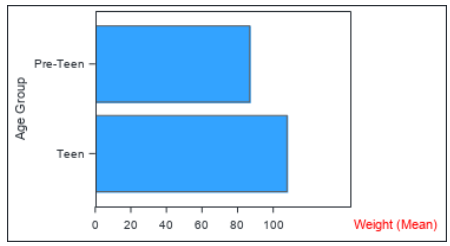
RIGHT
positions the label to the right of the axis area.

The label is centered vertically in the axis area.

The following figure shows the CENTER and DATACENTER positions for the red X-axis label “Weight (Mean).” An axis offset is applied to the maximum end of the axis in order to demonstrate the difference between CENTER and DATACENTER. CENTER centers the labels on the entire axis area, including the offset. DATACENTER centers the labels on the tick display areas, which does not include the offset.

CENTER	DATACENTER
	

The next figure shows the LEFT and RIGHT positions for the same axis labels.

LEFT	RIGHT
	

Default CENTER

LOGBASE=2 | 10 | e
specifies the base value for the logarithmic scale.

Default 10

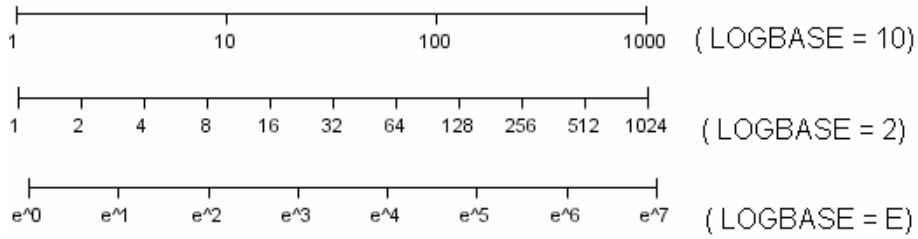
Interaction This option has no effect unless you also specify TYPE=LOG.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND
places the tick marks at integer powers of the base.

For example, if you specified **LOGBASE=2**, the tick marks might be at 1, 2, 4, 8, 16. See Figure 10.35 on page 1561 .

Figure 10.13 *Graph Axes with LOGEXPAND*



LOGEXPONENT
places the tick marks at integer powers of the base, but identifies the values by the exponent.

For example, if you specified **LOGBASE=10**, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See Figure 10.36 on page 1561 .

Figure 10.14 *An Axis with LOGEXPONENT*

LINEAR
places the tick marks at uniform linear intervals, but spaces them logarithmically.

In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See Figure 10.37 on page 1561 .

Figure 10.15 *An Axis with LINEAR*

Default LOGEXPAND

Interaction This option has no effect unless you also specify TYPE=LOG.

LOGVTYPE=EXPANDED | EXPONENT
specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options. This option enables you to choose your preferred way of specifying log-axis values regardless of the LOGSTYLE= option value.

Specify one of the following values:

EXPANDED
the values are interpreted as integer powers of the base (decimal numbers).

EXPONENT
the values are interpreted as integer exponents of the base.

Default EXPANDED

Interaction This option has no effect unless you also specify TYPE=LOG. You must also specify values for the VALUES= option or the MIN= and MAX= options or all of them.

Tip This option is particularly useful when the log axis is an odd base (such as base E) or the axis log style is EXPONENT.

Examples The following example specifies MIN= and MAX= as exponent values instead of expanded values on an expanded Base 10 log axis. This results in Y-axis tick values of 10, 100, 1000, 10000, and 100000.

```
yaxis type=log logbase=10 logstyle=logexpand
   logvtype=exponent
   min=1 max=5;
```

The following example specifies VALUES= as a list of expanded values instead of exponent values on an exponent Base 10 log axis. This results in X-axis tick values of 1, 2, 3, 4, and 5.

```
xaxis type=log logbase=10 logstyle=logexponent
   logvtype=expanded
   values=(10 100 1000 10000 100000);
```

MAX=*numeric-value*
specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES=option and you do not also specify the VALUESHINT option.

This option does not determine the maximum axis tick value displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the maximum value is expanded or interpreted as an exponent.

Tip The maximum axis tick value might differ from the MAX= value. The MAX= and MIN= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values. To display the MAX= value as the maximum tick value, use the VALUES= option.

MIN=*numeric-value*
specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

This option does not determine the minimum axis tick value displayed. The THRESHOLDMIN= value is used to determine the minimum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the minimum value is expanded or interpreted as an exponent.

Tip The minimum axis tick value might differ from the MIN= value. The MIN= and MAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values. To display the MIN= value as the minimum tick value, use the VALUES= option.

MINOR
adds minor tick marks to a linear, log, or time axis.

Restriction This option has no effect on discrete axes.

Interaction This option has no effect if you specify the VALUES= option.

Tip Use MINORCOUNT= to specify the number of tick marks.

MINORCOUNT=*numeric-value*
specifies the number of minor tick marks for the axis. This value determines the number of minor tick marks for each interval on the axis.

Restriction This option applies to linear and log axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

MINORGRID
creates grid lines at each minor tick on the axis.

Interaction This option has no effect unless GRID is also specified for the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the minor grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphMinorGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless MINORGRID is also specified.

Tip You can use GRIDATTRS= to change the appearance of the major grid lines.

Examples
```
MINORGRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:

```
MINORGRIDATTRS=GraphAxisLines
```

MINORINTERVAL=*time-interval*

specifies the time interval between minor ticks. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

For information about the intervals that you can select, see Table 10.141 on page 1557.

Default AUTO

Restriction This option applies to time axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

NOTIMESPLIT

prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction This option applies to time axes only.

OFFSETMAX=*numeric-value*

specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the maximum end of the axis to the highest data value or highest tick value, whichever is greater.

For a discrete axis, the offset is measured from the maximum end of the axis to the midpoint of the last category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the maximum end of the axis. The sum of the minimum and maximum offsets does not exceed 50% of the axis length.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See "Adjusting Axis Offsets" on page 1788

OFFSETMIN=*numeric-value*

specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the beginning of the axis to the lowest data value or lowest tick value, whichever is less.

For a discrete axis, the offset is measured from the beginning of the axis to the midpoint of the first category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the minimum end of the axis. The sum of the minimum and maximum offsets does not exceed 50% of the axis length.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See "Adjusting Axis Offsets" on page 1788

RANGES=(*start–end <start2–end2 startN–endN ...>*)
specifies the ranges for a broken axis.

start
specifies the start of a range.

Start can be one of the following:

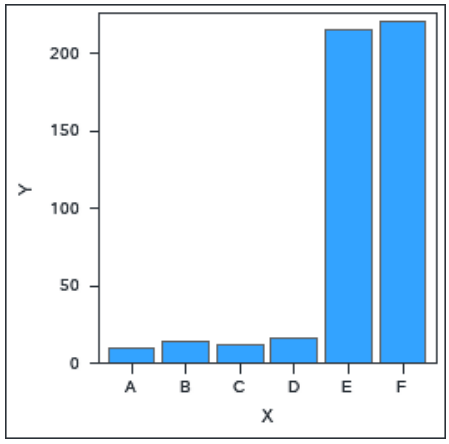
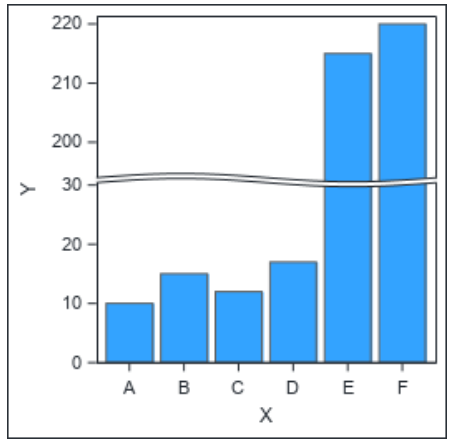
- a number (linear axis only).
- the keyword MIN specifies the minimum data value.
- a SAS time, date, or date-time constant (time axis only).

end
specifies the end of a range.

End can be one of the following:

- a number (linear axis only).
- the keyword MAX specifies the maximum data value.
- a SAS time, date, or date-time constant (time axis only).

The following figure shows a linear axis, broken into ranges 0–30 and 195–220. Although the figure shows the split range on a vertical axis, the same concept applies to a horizontal axis.

RANGES Not Specified	RANGES Specified
	

As shown in the figure, break lines are drawn to indicate the break in the axis.

Restrictions This option is valid for linear and time axes only.

Only one axis can be broken. If this option is specified for both axes, then it is honored for the vertical axis and ignored for the horizontal axis.

If this option is specified for both axes in the same direction (X and X2 or Y and Y2), then only the primary axis is broken.

When plots are associated with both the X and X2 axes or with both the Y and Y2 axes, neither axis can be broken.

Requirements All of the ranges must be enclosed in parentheses.

You must specify each range as a starting value, a hyphen, and an ending value. You must separate adjacent ranges with a space.

Interactions When this option is specified, axis options THRESHOLDMIN=, THRESHOLDMAX=, MIN=, and MAX= are ignored.

When this option is specified, data tips are not displayed, and the plot URL= option is ignored.

When data labels are used in the graph, the data label font size might be reduced in order to avoid overlapping labels and markers. When a range is specified, the data label font size is not scaled during label placement.

Notes When this option is specified, data-clipping might occur for the following graphics elements: plot markers and marker characters, box plot outlier markers, fixed-position data labels, needle fringe plots, reference lines and drop lines on the broken axis, axis tables, and relative bubble plots.

Curve label positions are based on the non-broken axis data range. When curve labels are specified with a broken axis, the curve label positions might not be ideal.

Tip To control the axis break symbol, use the AXISBREAK= option in the STYLEATTR statement.

Examples
```
ranges=(10-500 1000-5000 10000-50000)

ranges=('01Jan2001'd-'01May2003'd '01Jan2005'd-'01Oct2005'd)
```

REFTICKS
REFTICKS=(*options*)
adds tick marks to the axis that is opposite from the specified axis. You can also specify *options*:

LABEL
in addition to the tick marks, displays the axis label.

NOTICKS
suppresses the tick marks.

VALUES
in addition to the tick marks, displays the values that are represented by the tick marks.

Note This option has no effect if the target axis already contains data.

REVERSE
specifies that the tick values are displayed in reverse (descending) order.

SPLITCHAR="*character-list*"
splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect. When FITPOLICY=SPLIT is in effect, the text is split on occurrences of the specified split character or characters where it is required to fit the tick marks. When FITPOLICY=SPLITALWAYS is in effect, the text is split on all occurrences of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Restriction This option has no effect unless the axis is discrete.

Interactions This option has no effect unless FITPOLICY= is specified as either SPLIT, SPLITALWAYS, or SPLITROTATE. SPLITROTATE is the default for discrete X axes.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See “Fit Policies for Axis Tick Values, Curve Labels, and Data Labels” on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Default CENTER

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THRESHOLDMAX=*numeric-value*
Specifies a threshold for displaying one more tick mark at the high end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

THRESHOLDMIN=*numeric-value*
Specifies a threshold for displaying one more tick mark at the low end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

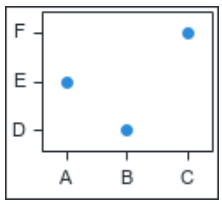
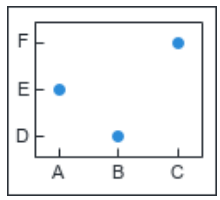
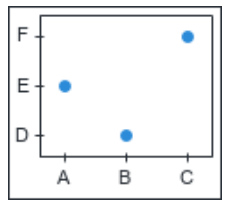
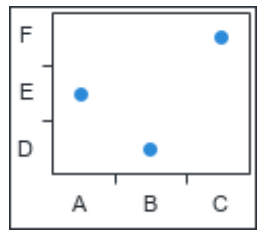
Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

TICKSTYLE=OUTSIDE | INSIDE | ACROSS | INBETWEEN
specifies the placement of the major tick marks in relation to the axis line. The figure shows the tick display for each value.

Note: Although the figure shows tick displays for the primary axes, the same placement in relation to the axis applies when secondary axes are used.

OUTSIDE	INSIDE	ACROSS	INBETWEEN
			

OUTSIDE
displays the major tick marks outside of the axis frame.

INSIDE
displays the major tick marks inside the axis frame.

ACROSS
displays the major tick marks across the axis line.

INBETWEEN
displays the major tick marks between the discrete data values and outside of the axis frame.

Interaction This value is available only with discrete axes. The option is ignored if specified with other axis types.

Default OUTSIDE

Restriction This option applies to the major tick marks only. It does not affect the minor tick marks.

Interaction This option has effect only if the DISPLAY= option setting includes tick marks, which it does by default.

Note This option has no effect on the placement of the tick values, which are always outside the axis frame.

Tip You can use the SG annotation facility to draw inside minor tick marks in your graph. See Part 7, “SG Annotation,” on page 1933. The post Creating an axis in GTL with both major and minor tick marks

shown inside on the Graphics Programming community page provides an example for the SGPLOT procedure.

TYPE=DISCRETE | LINEAR | LOG | TIME
specifies the type of axis. Specify one of the following values:

DISCRETE
specifies an axis with discrete values.

If a character variable is assigned to an axis, then the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable. Bar charts support a linear category axis.

LINEAR
specifies a linear scale for the axis.

This is the default axis type for numeric variables, except when the data is discrete, or when the numeric variable has a date or time format.

LOG
specifies a logarithmic scale for the axis.

This axis type is never a default.

Restriction A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

Interactions Use the LOGSTYLE= option to specify the scale and format for the tick values.

Use the LOGBASE= option to specify the base value.

Use the LOGVTYPE= option to specify how the values that are provided in the VALUES= option and the MIN= and MAX= options are interpreted.

TIME
specifies a time scale for the axis.

If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

For ungrouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element, and the

ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

VALUES=(*values-list*) | (*"string-list"*)

specifies the values for the ticks on an axis. The syntax for this option varies depending on the type of axis.

- VALUES=(*values-list*) specifies tick values for linear, time, and logarithmic axes.
- VALUES=(*"string-list"*) specifies tick values for discrete axes. The values can be character or numeric.

values-list

specifies tick values for linear, time, and logarithmic axes.

Values in *values-list* can be specified as a simple list of values in the following form:

value <...*value-n*>

The values must be listed in either ascending or descending order. The values can also be specified as a range of values of a specific increment in the following form:

value-1 TO *value-2* BY *increment-value*

By default the increment value is 1. You can use a negative integer for increment to specify a value list in descending order.

You can use both forms when specifying values:

<*value* ... *value-n* >*value-1* TO *value-2* BY *increment-value* <*value* ... *value-n*>

In all forms, multiple *n* values can be separated by blanks or commas. Here are some examples:

- Place tick marks at 2, 4, and 6 on a linear axis:

```
values=(2 4 6)
```

- Place tick marks at 6, 4, and 2 on a linear axis:

```
values=(6,4,2)
```

- Place tick marks at 25MAY08, 04JUL08, and 23AUG08 on a time axis:

```
values=("25MAY08"d "04JUL08"d "23AUG08"d)
```

- Place tick marks from 2 to 10 at an increment of 2 on a linear axis:

```
values=(2 to 10 by 2)
```

- Place tick marks from 50 to 10 at an increment of -5 on a linear axis:

```
values=(50 to 10 by -5)
```

If the specified range is not evenly divisible by the increment value, the highest value displayed on the axis is the last incremental value below the ending value for the range. For example, this value list produces a maximum axis value of 9:

```
values=(0 to 10 by 3)
```

Restrictions This option has no effect on discrete axes.

If your VALUES= option creates more than 1000 values, then the option has no effect.

Interactions For logarithmic axes, use the LOGVTYPE option to control whether the values are expanded or interpreted as exponents.

If a custom format is applied to the value, the raw value is plotted first and then formatted.

Tips Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

For graphs such as histograms, the final bin might be truncated at the maximum data value. If you need to expand the axis in order to show all of the data, use the VALUESHINT option.

"string-list"

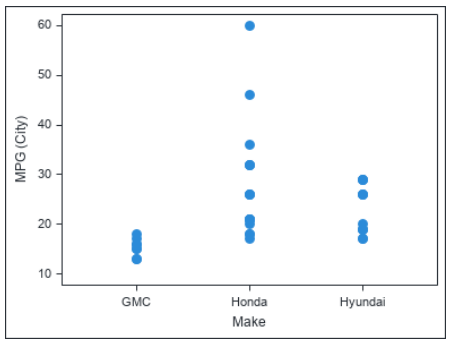
specifies a space-separated list of string values enclosed in parentheses for a discrete axis.

Each value in the list must be enclosed in quotation marks. Numeric values must also be enclosed in quotation marks.

Only the tick values that are included in the string list are displayed on the axis. The values are displayed in the order in which they are listed. The data values that are not in the list are dropped. The list can be a subset of the data values. It can also contain values that are not included in the actual data. A tick value that is not included in the data appears on the axis, but no data is represented at its tick mark.

The following example specifies the values for an X axis. Similar values can be provided for the Y axis.

Table 10.63 *Axis with Specified Character Values*




```
proc sgplot data=sashelp.cars;
xaxis values=("GMC" "Honda" "Hyundai")
   offsetmin=0.2 offsetmax=0.2;
scatter x=make y=mpg_city;
run;
```

Restrictions This option has no effect on linear, time, or logarithmic axes.

Each value must be enclosed in quotation marks and separated from adjacent values by a blank space.

Notes If the string list contains duplicate values, the first occurrence of the duplicated value in the list is honored while the remaining instances are ignored.

The axis data can be character or numeric.

If a numeric variable has an associated format, the specified values must use the same format.

Tip Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

Examples The following example specifies the axis tick values Sedan, Sports, Wagon, and SUV:

```
values=("Sedan" "Sports" "Wagon" "SUV")
```

The following example specifies the axis tick values 10, 20, 30, and 40:

```
values=("10" "20" "30" "40")
```

The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

TIP Along the axis, numeric tick values are arranged in ascending order while character values are arranged in the order in which they are used in the graph. In some cases, the resulting tick-value order might not be desirable, especially if the graph consists of multiple plots or if the data contains missing values. This option is useful in that case. You can use this option to set the order of the axis tick values.

VALUESDISPLAY=

specifies the text that is to be displayed for the tick values that are defined in the VALUES= option. The list of values must be enclosed in parentheses. Each value must be enclosed in quotation marks and separated from adjacent values by a blank space. Numeric values must also be enclosed in quotation marks.

Restriction This option applies only to discrete and linear axes.

Interaction This option should be used with the VALUES= option. The number of items in the list for this option should equal the number of items in the list for the VALUES= option.

Example The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

VALUESFORMAT=DATA | *SAS-format*
specifies how to format the values for major tick marks. This option supports discrete and logarithmic axes.

DATA
uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

SAS-format
specifies a format to apply to the major tick values.

Restriction This option currently honors most, but not every, SAS format. For more information, see "Format Support in ODS Graphics" in *SAS Graph Template Language: Reference*.

Notes If you specify a format that significantly reduces precision, then, because of tick-value rounding, the plot data elements might not align properly with the axis tick values. In that case, specify a tick-value format with a higher precision.

A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F5.1

Alias TICKVALUEFORMAT

Restriction For discrete axes, only character formats are supported.

Interactions This option is ignored when LOGSTYLE=LOGEXPONENT.

When LOGSTYLE =LOGEXPAND, this option is honored for the base 10 and base 2 logarithmic scales, and is ignored for the base E scale.

When LOGSTYLE =LINEAR, this option is honored for the base 10, base 2, and base E logarithmic scales.

VALUESHINT
specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option. Tick values specified by the VALUES= option are treated as suggestions for the tick values that actually appear, so that the actual data range of the axis is preserved. If the *values-list* contains values that are less than the data minimum, or greater than the data maximum, then those values are ignored.

Interaction This option has no effect unless you also specify the VALUES= option.

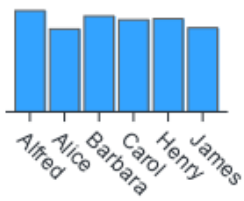
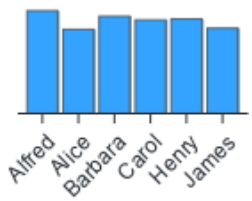
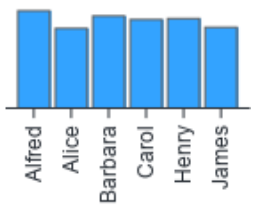
VALUESROTATE=DIAGONAL | DIAGONAL2 | VERTICAL
specifies how the tick values are rotated on the axis.

DIAGONAL
rotates the tick values diagonally.

DIAGONAL2
rotates the tick values to a reverse diagonal angle.

VERTICAL
rotates the tick values vertically.

The values are always drawn from bottom to top.

DIAGONAL	DIAGONAL2	VERTICAL
		

Default DIAGONAL

Interactions This option works through the axis fit policy. If there are no tick collisions, no rotation occurs.

With a time axis, you must specify NOTIMESPLIT for this option to have any effect.

XAXISTABLE Statement

Creates a basic plot that displays data values at specific locations along the horizontal axis.

Requirements: At least one other plot statement must be used in the procedure. This other plot becomes the primary plot. The primary plot provides the default placement of values along the axis table's horizontal axis.

For the axis table, you must specify at least one variable. This variable determines which values are displayed in the axis table.

Interactions: You can specify a variable for the X= option to identify the locations along the axis where the values are displayed. If you do not specify X=, the locations are determined by the X variable in the primary plot. When the primary plot is a categorical chart (bar, line, or dot), the category variable from the chart is used. With the exception of that X (or category) variable and any group variable that might be specified, axis tables are separate plots and are unaware of the options specified in the accompanying plots.

When used with bar charts, line charts, and dot plots, all axis tables must align with the category axis of the chart. If a statement uses the wrong orientation, the statement is rejected with a message in the SAS log. For example, if your procedure has an HBAR statement along with an XAXISTABLE statement, the XAXISTABLE statement is rejected with a message.

Axis tables cannot be used when the primary plot is one of the following plot types: BAND, BLOCK, FRINGE, REG, LOESS, and PBSPLINE. The axis table requires at least one other plot statement that is not one of these types. Otherwise, the axis table is not created and an error is written to the log.

Notes:

The data values can be placed inside or outside of the axis.

You can use multiple axis table statements.

Syntax

XAXISTABLE *variable* <*...variable-n*> </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORGROUP=*variable*
: specifies a variable that is used to determine the color of the table values.

DROPONMISSING
: specifies that the entire axis table is dropped when all of the values are missing.

LOCATION=OUTSIDE | INSIDE
: specifies whether the axis table is placed outside or inside the axis area.

NOMISSINGCHAR
: suppresses the display of the MISSING character (.) for missing numeric values.

PAD=*numeric-value*<*units*> | *(pad-options)*
: specifies the amount of extra space that is added inside the table border.

POSITION=BOTTOM | TOP
: specifies the position of the axis table at the bottom or top of the graph.

SEPARATOR
: creates a separating line between the axis table or axis tables and the plot.

TEXTGROUP=*attribute-map-group-variable*
: specifies the group variable that is used in a discrete attribute map data set to map text attributes to values for each observation.

TEXTGROUPID=*attribute-map-id*
: specifies an attribute ID for the TEXTGROUP= option.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the axis table values.

Axis options

X2AXIS
: assigns the table to the secondary (top) horizontal axis.

Class options

CLASS=*variable*
: creates a separate axis table for each unique value of the specified variable.

CLASSDISPLAY=STACK | CLUSTER
: specifies how the class values are displayed.

CLASSORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the order in which the class values are displayed.

NOMISSINGCLASS
: specifies that missing values of the class variable are not included in the table.

Label options

LABEL

LABEL="*text-string*"

NOLABEL
: specifies whether the table label is shown or hidden.

LABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the color and font attributes of the axis table label.

LABELPOS=LEFT | RIGHT
: specifies the position of the labels at the left or right side of the axis table.

STATLABEL

NOSTATLABEL
: specifies whether the variable statistic is displayed in the table's label.

Plot options

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
: specifies the statistic for the axis table.

X=*variable*
: Specifies the X variable to use to align the table values to the X or X2 axis.

Plot reference options

NAME="*text-string*"
: assigns a name to a plot statement.

Title options

TITLE="*text-string*"
: specifies a title for the axis table.

TITLEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the title for the axis table.

Required Argument

variable <...variable-n>
: specifies one or more variables for the axis table. When multiple variables are specified, the axis tables are stacked one on the other. Any options that you add to the statement apply to all the variables that are specified in that statement.

Note When the variable specified is a character variable, the first value of each category is displayed in the axis table. When the variable is numeric, the axis table displays the sum statistic. You can modify the statistic using the STAT= option.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CLASS=*variable*
creates a separate axis table for each unique value of the specified variable. This option acts as a classification variable for the axis table. Each axis table is labeled by the class value.

Restriction The CLASS option is ignored when the axis table is used with bar, line, or dot charts. If the GROUP= option is specified in the chart, that group variable is used as the CLASS variable for all axis tables.

Interaction If NOLABEL is also specified, then the class labels are removed.

Tips Use the CLASSDISPLAY= option to control whether the class values are clustered or stacked.

Use the CLASSORDER= option to control the order in which the class values are displayed.

CLASSDISPLAY=STACK | CLUSTER
specifies how the class values are displayed.

STACK
displays the class values vertically at each midpoint value on the X axis.

CLUSTER
displays the class values horizontally at each midpoint value on the X axis.

Restriction The CLASSDISPLAY=CLUSTER option applies only when the axis table is on a discrete axis.

Default STACK

Interaction For this option to have any effect, the CLASS= option must be specified.

CLASSORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the order in which the class values are displayed.

DATA
displays the class values in the order in which they occur in the data.

REVERSEDATA
displays the class values in the reverse order from which they occur in the data.

Tip This option is useful when the plot axis is reversed.

ASCENDING
displays the class values in ascending order.

DESCENDING
displays the class values in descending order.

Default DATA

Restriction This option is ignored when the axis table is used with DOT, HBAR, VBAR, HLINE, and VLINE statements. If the GROUPORDER= option is specified in the chart, then that group order variable is used as the class order variable for all axis tables.

Interaction For this option to have any effect, the CLASS= option must be specified.

COLORGROUP=*variable*
specifies a variable that is used to determine the color of the table values. Once the variable values are found, the value colors are taken from the GraphData1–GraphData*n* style elements in the current style. The CONTRASTCOLOR attribute is used for the value text.

Restriction When used with DOT, HBAR, VBAR, HLINE, and VLINE statements, this option has no effect unless *variable* is used in the accompanying chart.

Note This option is used only to color the table values. If you want to set additional text attributes, used the TEXTGROUP= option instead.

DROPONMISSING
specifies that the entire axis table is dropped when all of the values are missing. Consider using this option if the SAS log indicates that the specified data column used for the axis table is missing all values.

LABEL
LABEL="*text-string*"
NOLABEL
specifies whether the table label is shown or hidden. If you specify LABEL=, then you can also specify a text string for the label.

Defaults LABEL

If you do not specify a text string, then the variable name is used for the label. Or, if CLASS= is also specified, then the unique values of the specified class variable are used for the table labels.

Tip Use the LABELATTRS= option to modify the label text attributes. Use the LABELPOS= option to move the label.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the color and font attributes of the axis table label. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

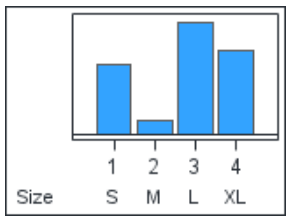
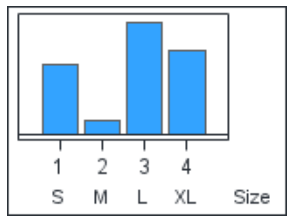
For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Restriction Group behavior occurs only when the CLASS= and COLORGROUP= option values are the same.

Interactions This option has no effect if NOLABEL is also specified.

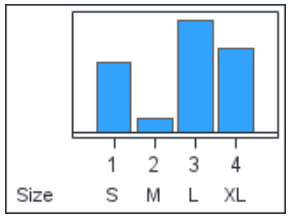
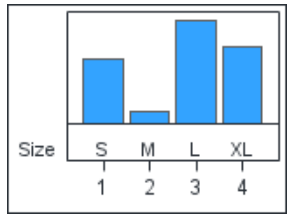
If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphValueText style element.

LABELPOS=LEFT | RIGHT
specifies the position of the labels at the left or right side of the axis table.

LEFT	RIGHT
	

Default LEFT

LOCATION=OUTSIDE | INSIDE
specifies whether the axis table is placed outside or inside the axis area.

OUTSIDE	INSIDE
	

Default OUTSIDE

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGCHAR
suppresses the display of the MISSING character (.) for missing numeric values. Missing numeric values are displayed as blanks.

NOMISSINGCLASS
specifies that missing values of the class variable are not included in the table.

Interaction For this option to have any effect, the CLASS= option must be specified.

PAD=*numeric-value<units>* | *(pad-options)*
specifies the amount of extra space that is added inside the table border. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

numeric-value<units>
specifies a dimension to use for the extra space at the table border.

(pad-options)
specifies a space-separated list of pad options, enclosed in parentheses.

Specify one or more of the following name-value-pair options:

TOP=*dimension*
specifies the amount of extra space added to the top.

Default 0 px

BOTTOM=*dimension*
specifies the amount of extra space added to the bottom.

Default 0 px

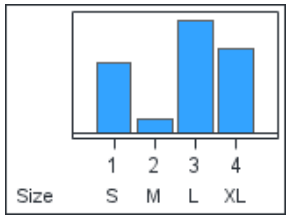
Notes Sides that are not assigned padding are padded with the default amount of space.

LEFT and RIGHT are not valid for an X-axis table.

Tip Use *pad-options* to create non-uniform padding.

Note The default units for *dimension* are pixels. If you want to specify values in other units, then you must specify the desired units with the value. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

POSITION=BOTTOM | TOP
specifies the position of the axis table at the bottom or top of the graph.

BOTTOM	TOP
	

Defaults BOTTOM when the primary horizontal axis is used for the plot.

TOP when the secondary (X2AXIS) horizontal axis is specified for the plot.

SEPARATOR
creates a separating line between the axis table or axis tables and the plot.

Interaction This option has no effect unless LOCATION=INSIDE is also specified.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
specifies the statistic for the axis table. The statistic is applied only to numeric variables. For non-categorical plots, only the SUM and MEAN statistics are available.

Specify one of the following:

FREQ
the frequency of the axis table variable.

Interactions For this value to take effect, the graph must use a categorical plot (bar chart, line chart, or dot plot).

The FREQ value has a different effect when the axis table variable is the same as the categorical variable of the primary plot. In that case, the axis table variable is treated as a category (class) variable instead of a response (analysis) variable when

computing the frequency. The FREQ value is the frequency of the categorical variable.

MEAN
the mean of the axis table variable.

MEDIAN
the median of the axis table variable.

Interactions For this value to take effect, the graph must use a category plot (bar chart, line chart, or dot plot).

This value is ignored when the axis table variable is the same as the categorical variable of the primary plot. In that case, the FREQ statistic is used and a message is written to the SAS log.

PERCENT
the percentage of the sum of the axis table variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions For this value to take effect, the graph must use a categorical plot (bar chart, line chart, or dot plot).

The PERCENT value has a different effect when the axis table variable is the same as the categorical variable of the primary plot. In that case, the axis table variable is treated as a category (class) variable instead of as a response (analysis) variable when computing the percentage. Therefore, the PERCENT value is calculated as a percentage of frequency instead of a percentage of the sum.

The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the PCTLEVEL= option in the PROC SGPLOT statement.

You can use the PCTNDEC= option in the SGPLOT procedure statement to control the number of decimals to be used when calculating the percent values. The default value is 1.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level are zero.

SUM
the sum of the axis table variable.

Defaults SUM

When used with a categorical plot and the axis table variable is the same as the categorical variable, FREQ is the default.

Restrictions The STAT= option is applied only to numeric variables.

Only SUM and MEAN are supported for non-categorical plots.

When used with a categorical plot and the axis table variable is the same as the categorical variable, only FREQ and PERCENT are supported.

Interaction Any STAT= value specified in the primary plot has no effect on the axis table statistic.

STATLABEL | NOSTATLABEL
specifies whether the variable statistic is displayed in the table's label.

STATLABEL
forces the statistic to be displayed in the label.

NOSTATLABEL
removes the statistic from the label.

Defaults The statistic is displayed for the variable.

When a custom label is assigned to the variable, the statistic is not displayed.

Interaction This option has no effect unless the STAT= option is also specified in the axis table statement.

TEXTGROUP=*attribute-map-group-variable*
specifies the group variable that is used in a discrete attribute map data set to map text attributes to values for each observation. You specify this option only if you are using an attribute map to control visual attributes of the graph. The variable's values must correspond to the values in the VALUE variable in the attribute map data set. For more information, see Chapter 18, "Using Discrete Attribute Maps," on page 1901.

Interactions The TEXTGROUPID= option specifies the attribute ID to use for the attribute mapping. If TEXTGROUPID= is not specified, then the ATTRID= option is used. If the ATTRID= option is also not specified, then the TEXTGROUP= option is ignored.

When this option is specified, the COLORGROUP= option is ignored.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

TEXTGROUPID=*attribute-map-id*
specifies an attribute ID for the TEXTGROUP= option.

Default If neither TEXTGROUPID= nor ATTRID= is specified, the TEXTGROUP= option is ignored.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

TITLE="*text-string*"
specifies a title for the axis table. It the axis table statement specifies more than one variable, the title is displayed for each variable.

Tip Use the TITLEATTRS= option to modify the title text attributes.

TITLEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the title for the axis table. You can specify the appearance by using a style element or by specifying text options. If you specify a style element, you can also specify text options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interactions This option has no effect unless TITLE= is also specified.

If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphDataText style element.

Examples
```
TITLEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TITLEATTRS=GraphTitleText
```

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis table values. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeigh attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeigh attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interaction If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then

the properties that are not specified are derived from the GraphDataText style element.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

X=*variable*
Specifies the X variable to use to align the table values to the X or X2 axis.

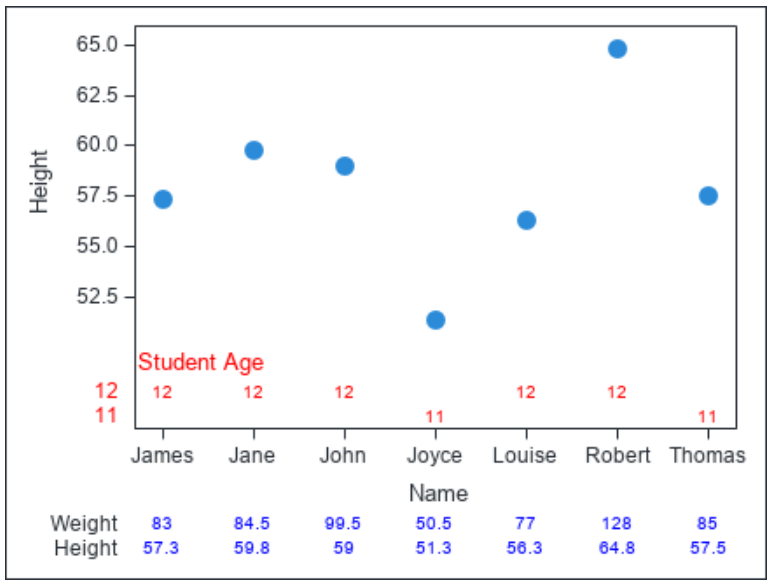
Default When this option is not specified, the X variable of the primary plot is used by default.

Interaction This option is ignored when the axis table is used with a categorical chart (bar, line, or dot). The category variable from the chart is used instead.

X2AXIS
assigns the table to the secondary (top) horizontal axis.

Example: XAXISTABLE Statements

This example shows a table of values along the X axis of a scatter plot that plots student weight.



Here are the noteworthy features:

- The first axis table statement displays the age (in red) of each student above the X axis. The LOCATION= on page 1642INSIDE option locates the axis table within the plot axes. The ages appear on different rows because the CLASS= on page 1638AGE option is used.

- The second axis table statement specifies two variables, height and weight, and is located below the X axis.

Here is the SAS code for this example.

```
ods graphics / width=4.5in;

proc sgplot data=sashelp.class (where=(age < 13));
scatter x=name y=height;
xaxistable age / class=age title="Student Age" location=inside
      valueattrs=(color=red)
      labelattrs=(color=red)
      titleattrs=(color=red);
xaxistable weight height / valueattrs=(color=blue);
run;
```

For an example that shows vertical axis tables, see "Example: YAXISTABLE Statements" on page 1649.

YAXIS Statement

Specifies the axis options for the Y axis.

Notes:

All YAXIS options should be included in a single YAXIS statement. Otherwise, some axis options might not be applied.

You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

YAXIS *options*;

Summary of Optional Arguments

Appearance options

COLORBANDS=NONE | EVEN | ODD
: specifies the display of alternating wall-color bands corresponding to the discrete axis bins.

COLORBANDSATTRS=*style-element*<(***options***)> | (*options*)
: specifies the fill appearance of the color band.

DISPLAY=ALL | NONE | (*options*)
: specifies which features of the axis are displayed.

GRID
: creates grid lines at each tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the grid lines.

MINORGRID
creates grid lines at each minor tick on the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the minor grid lines.

TICKSTYLE=OUTSIDE | INSIDE | ACROSS | INBETWEEN
specifies the placement of the major tick marks in relation to the axis line.

Axis options

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED
specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis.

INTEGER
specifies that only integers are used for tick mark values.

INTERVAL=*time-interval*
specifies the tick interval for a time axis.

INTERVALMULT=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the axis.

LOGBASE=2 | 10 | e
specifies the base value for the logarithmic scale.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
specifies how to scale and format the values for the major tick marks for logarithmic axes.

LOGVTYPE=EXPANDED | EXPONENT
specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options.

MAX=*numeric-value*
specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

MIN=*numeric-value*
specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

MINOR
adds minor tick marks to a linear, log, or time axis.

MINORCOUNT=*numeric-value*
specifies the number of minor tick marks for the axis.

MINORINTERVAL=*time-interval*
specifies the time interval between minor ticks.

NOTIMESPLIT
prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX=*numeric-value*
specifies an offset that follows the highest data value on the axis.

OFFSETMIN=*numeric-value*
specifies an offset that precedes the lowest data value on the axis.

RANGES=(*start–end <start2–end2 startN–endN ...>*)

specifies the ranges for a broken axis.

REFTICKS

REFTICKS=(*options*)

adds tick marks to the axis that is opposite from the specified axis.

REVERSE

specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX=*numeric-value*

Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN=*numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

TYPE=DISCRETE | LINEAR | LOG | TIME

specifies the type of axis.

Text options

FITPOLICY=NONE | SPLIT | SPLITALWAYS | THIN

specifies the method that is used to fit tick mark values on a vertical axis when there is not enough room to draw them normally.

LABEL="*text-string*"

specifies a label for the axis.

LABELATTRS=*style-element*<(***options***)> | (*options*)

specifies the appearance of the axis labels.

LABELPOS=BOTTOM | CENTER | DATACENTER | TOP

specifies the position of the axis label.

SPLITCHAR="*character-list*"

splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect.

SPLITCHARNODROP

specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT

specifies the horizontal alignment of the value text that is being split.

VALUEATTRS=*style-element*<(***options***)> | (*options*)

specifies the appearance of the axis tick value labels.

VALUES=(*values-list*) | ("*string-list*")

specifies the values for the ticks on an axis.

VALUESDISPLAY=

specifies the text that is to be displayed for the tick values that are defined in the VALUES= option.

VALUESFORMAT=DATA | *SAS-format*

specifies how to format the values for major tick marks.

VALUESHALIGN=LEFT | CENTER | RIGHT

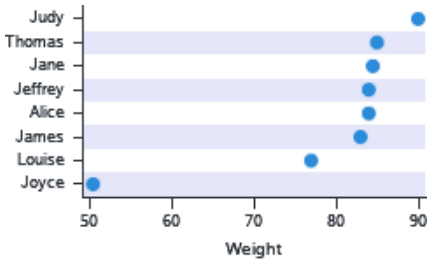
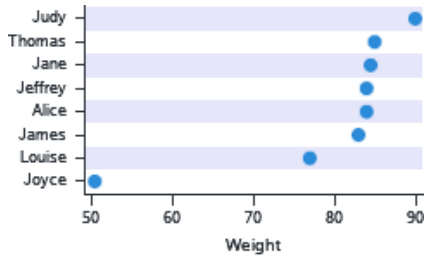
specifies the horizontal alignment for all of the tick values that are displayed on the axis.

VALUESHINT

specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option.

Optional Arguments

COLORBANDS=NONE | EVEN | ODD
specifies the display of alternating wall-color bands corresponding to the discrete axis bins. The following images show the results of ODD and EVEN settings:

COLORBANDS=ODD	COLORBANDS=EVEN
	

Default NONE

Restriction This option applies to discrete axes only.

Interaction Specifying this option for more than one axis in the graph might have unexpected results. The order in which color bands are drawn might not match the order in which the axis options are specified.

Note The full width of a color band is the distance between midpoints. When no axis offsets are specified, the first band begins at one-half of the midpoint distance, and the last band ends at one-half of the midpoint distance. When axis offsets are specified, the first and last color bands on the axis might extend into their adjacent offsets by as much as half the color-band width.

Tip Use the COLORBANDSATTRS= option to customize the color bands.

COLORBANDSATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill appearance of the color band. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interaction This option has no effect unless COLORBANDS= is also specified.

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED
specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis. Specify one of the following values:

DATA
places the values in the order in which they appear in the data.

Note: This value is not supported with data that is stored in a table.

FORMATTED
sorts the formatted values in ascending character order.

UNFORMATTED
sorts the unformatted values in ascending character order.

Default UNFORMATTED

DISPLAY=ALL | NONE | (*options*)
specifies which features of the axis are displayed.

ALL
displays all of the features of the axis.

NONE
specifies that none of the features of the axis are displayed.

(*options*)
specifies one or more axis features to hide.

Specify one or more of the following:

NOLABEL
hides the axis label

NOLINE
hides the axis line

Tips This value hides the axis line but has no effect on the graph's wall border, part of which occupies the same area as the axis line. To hide the wall border, specify NOBORDER in the PROC SGPLOTstatement.

This value has no effect on baselines. For plots that support a baseline, such as bar charts and needle plots, you might need to suppress the baseline. In the plot statement, use the BASELINEATTRS= option to set the line thickness to 0.

NOTICKS
hides the tick marks on the axis

NOVALUES
hides the tick mark values on the axis

Default ALL

Example DISPLAY=(NOTICKS NOVALUES)

FITPOLICY=NONE | SPLIT | SPLITALWAYS | THIN
specifies the method that is used to fit tick mark values on a vertical axis when there is not enough room to draw them normally. Select one of the following values:

NONE
does not split the values.

SPLIT
splits the values at the character or characters specified in the SPLITCHAR= option.

No split occurs at split characters that occur where a split is not needed. In that case, the split character is displayed with the text value. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Restriction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

SPLITALWAYS
always splits the values at the character or characters specified in the SPLITCHAR= option.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Restriction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

Example "Fit Policies for Axes" on page 1803

THIN
removes some of the values from the axis.

Default THIN

See "Fit Policies for Axes" on page 1803

GRID
creates grid lines at each tick on the axis.

Interaction Grid lines are not displayed when you specify the COLORBANDS= option. The color bands take the place of grid lines.

Tip You can specify the MINORGRID option to create grid lines at each minor tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless GRID is also specified.

Examples
```
GRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:
```
GRIDATTRS=GraphAxisLines
```

INTEGER
specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL=*time-interval*
specifies the tick interval for a time axis. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

Specify one of the following values:

Table 10.64 *Time Intervals*

INTERVAL	Unit	Tick interval	Default tick value format
AUTO	DATE, TIME, or DATETIME	automatically chosen	automatically chosen
SECOND	TIME or DATETIME	second	TIME8.
MINUTE	TIME or DATETIME	minute	TIME8.
HOUR	TIME or DATETIME	hour	TIME8.
DAY	DATE or DATETIME	day	DATE9.
TENDAY	DATE or DATETIME	10 days	DATE9.
WEEK	DATE or DATETIME	7 days	DATE9.
SEMIMONTH	DATE or DATETIME	1st and 16th of each month	DATE9.

INTERVAL	Unit	Tick interval	Default tick value format
MONTH	DATE or DATETIME	month	MONYY7.
QUARTER	DATE or DATETIME	3 months	YYQC6.
SEMIYEAR	DATE or DATETIME	6 months	MONYY7.
YEAR	DATE or DATETIME	year	YEAR4.

Default AUTO

Tip If unwanted thinning of tick marks occurs when you specify INTERVAL=, try using minor intervals instead. For example, you might specify MINOR MINORINTERVAL=MONTH instead of INTERVAL=MONTH. As a result, in the example monthly tick marks are displayed even though the major values are thinned.

INTERVALMULT=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the axis.

Alias INTERVALMULTIPLIER

Default 1

Restriction This option applies to time axes only.

Interaction This option is ignored if VALUES=*values-list* is used.

Tip Use the INTERVAL= option to specify a different time interval.

Examples To specify 3-month intervals:

```
INTERVAL=MONTH INTERVALMULT=3
```

To specify 10-year intervals:

```
INTERVAL=YEAR INTERVALMULT=10
```

LABEL="*text-string*"
specifies a label for the axis.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LABELPOS=BOTTOM | CENTER | DATACENTER | TOP
specifies the position of the axis label.

BOTTOM
positions the label at the bottom of the axis area.

The label is oriented horizontally (unrotated).

The label is right-justified in the axis area for the Y axis and left-justified for the Y2 axis. If there is not sufficient room in the axis area to display the label, the label grows to the right for the Y axis and to the left for the Y2 axis.

CENTER
centers the axis label in the axis area (including any offsets).

The label is oriented vertically (rotated).

The label is positioned to the left of the tick values for the Y axis or to the right of the axis values for the Y2 axis.

DATACENTER
centers the axis label in the axis tick display area (excluding any offsets).

The label is oriented vertically (rotated).

The label is positioned to the left of the tick values for the Y axis or to the right of the axis values for the Y2 axis.

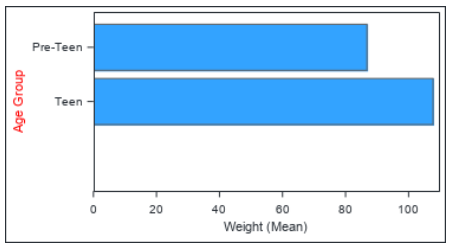
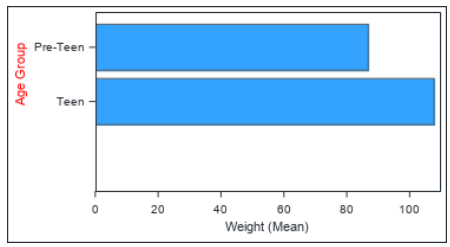
TOP
positions the label at the top of the axis area.

The label is oriented horizontally (unrotated).

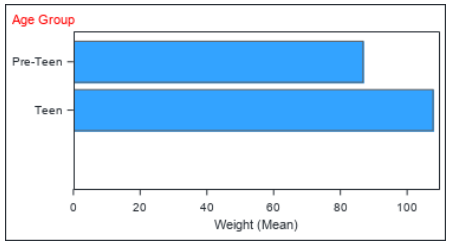
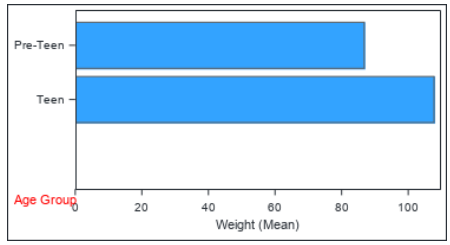
The label is right-justified in the axis area for the Y axis and left-justified for the Y2 axis. If there is not sufficient room in the axis area to display the label, the label grows to the right for the Y axis and to the left for the Y2 axis.

The following figure shows the CENTER and DATACENTER positions for the red Y-axis label “Age Group.” An axis offset is applied to the maximum end of the axis in order to demonstrate the difference between CENTER and DATACENTER. (For HBAR plot statements, the Y axis is reversed by default. The offset is created at the bottom end of the Y axis.)

In the example, CENTER centers the labels on the entire axis area, including the offset. DATACENTER centers the labels on the tick display areas, which does not include the offset.

CENTER	DATACENTER
	

The next figure shows the TOP and BOTTOM positions for the same axis labels.

TOP	BOTTOM
	

Default CENTER

LOGBASE=2 | 10 | e
specifies the base value for the logarithmic scale.

Default 10

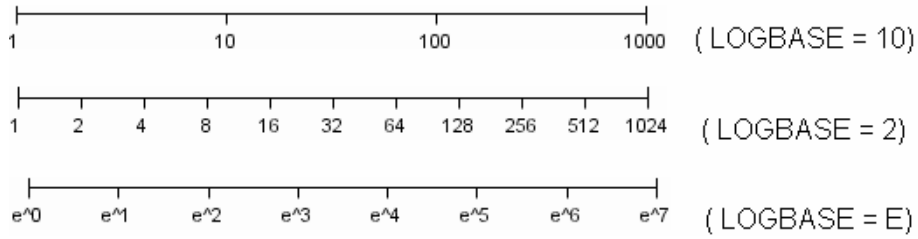
Interaction This option has no effect unless you also specify TYPE=LOG.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND
places the tick marks at integer powers of the base.

For example, if you specified **LOGBASE=2**, the tick marks might be at 1, 2, 4, 8, 16. See Figure 10.38 on page 1597 .

Figure 10.16 *Graph Axes with LOGEXPAND*



LOGEXPONENT
places the tick marks at integer powers of the base, but identifies the values by the exponent.

For example, if you specified **LOGBASE=10**, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See Figure 10.39 on page 1597 .

Figure 10.17 *An Axis with LOGEXPONENT*

LINEAR
places the tick marks at uniform linear intervals, but spaces them logarithmically.

In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See Figure 10.40 on page 1597 .

Figure 10.18 *An Axis with LINEAR*

Default LOGEXPAND

Interaction This option has no effect unless you also specify TYPE=LOG.

LOGVTYPE=EXPANDED | EXPONENT
specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options. This option enables you to choose your preferred way of specifying log-axis values regardless of the LOGSTYLE= option value.

Specify one of the following values:

EXPANDED
the values are interpreted as integer powers of the base (decimal numbers).

EXPONENT
the values are interpreted as integer exponents of the base.

Default EXPANDED

Interaction This option has no effect unless you also specify TYPE=LOG. You must also specify values for the VALUES= option or the MIN= and MAX= options or all of them.

Tip This option is particularly useful when the log axis is an odd base (such as base E) or the axis log style is EXPONENT.

Examples The following example specifies MIN= and MAX= as exponent values instead of expanded values on an expanded Base 10 log axis. This results in Y-axis tick values of 10, 100, 1000, 10000, and 100000.

```
yaxis type=log logbase=10 logstyle=logexpand
   logvtype=exponent
   min=1 max=5;
```

The following example specifies VALUES= as a list of expanded values instead of exponent values on an exponent Base 10 log axis. This results in X-axis tick values of 1, 2, 3, 4, and 5.

```
xaxis type=log logbase=10 logstyle=logexponent
   logvtype=expanded
   values=(10 100 1000 10000 100000);
```

MAX=*numeric-value*
specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES=option and you do not also specify the VALUESHINT option.

This option does not determine the maximum axis tick value displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the maximum value is expanded or interpreted as an exponent.

Tip The maximum axis tick value might differ from the MAX= value. The MAX= and MIN= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values. To display the MAX= value as the maximum tick value, use the VALUES= option.

MIN=*numeric-value*
specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

This option does not determine the minimum axis tick value displayed. The THRESHOLDMIN= value is used to determine the minimum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the minimum value is expanded or interpreted as an exponent.

Tip The minimum axis tick value might differ from the MIN= value. The MIN= and MAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values. To display the MIN= value as the minimum tick value, use the VALUES= option.

MINOR
adds minor tick marks to a linear, log, or time axis.

Restriction This option has no effect on discrete axes.

Interaction This option has no effect if you specify the VALUES= option.

Tip Use MINORCOUNT= to specify the number of tick marks.

MINORCOUNT=*numeric-value*
specifies the number of minor tick marks for the axis. This value determines the number of minor tick marks for each interval on the axis.

Restriction This option applies to linear and log axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

MINORGRID
creates grid lines at each minor tick on the axis.

Interaction This option has no effect unless GRID is also specified for the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the minor grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphMinorGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless MINORGRID is also specified.

Tip You can use GRIDATTRS= to change the appearance of the major grid lines.

Examples
```
MINORGRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:
```
MINORGRIDATTRS=GraphAxisLines
```

MINORINTERVAL=*time-interval*
specifies the time interval between minor ticks. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

For information about the intervals that you can select, see Table 10.143 on page 1593.

Default AUTO

Restriction This option applies to time axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

NOTIMESPLIT
prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction This option applies to time axes only.

OFFSETMAX=*numeric-value*
specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the maximum end of the axis to the highest data value or highest tick value, whichever is greater.

For a discrete axis, the offset is measured from the maximum end of the axis to the midpoint of the last category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the maximum end of the axis. The sum of the minimum and maximum offsets does not exceed 50% of the axis height.

Interaction For HBOX, HBAR, HLINE, and DOT plot statements, the Y axis is reversed by default, so the axis origin is at the top.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See “Adjusting Axis Offsets” on page 1788

OFFSETMIN=*numeric-value*
specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the beginning of the axis to the lowest data value or lowest tick value, whichever is less.

For a discrete axis, the offset is measured from the beginning of the axis to the midpoint of the first category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the maximum end of the axis. The sum of the minimum and maximum offsets does not exceed 50% of the axis height.

Interaction For HBOX, HBAR, HLINE, and DOT plot statements, the Y axis is reversed by default, so the axis origin is at the top.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See "Adjusting Axis Offsets" on page 1788

RANGES=(*start–end <start2–end2 startN–endN ...>*)
specifies the ranges for a broken axis.

start
specifies the start of a range.

Start can be one of the following:

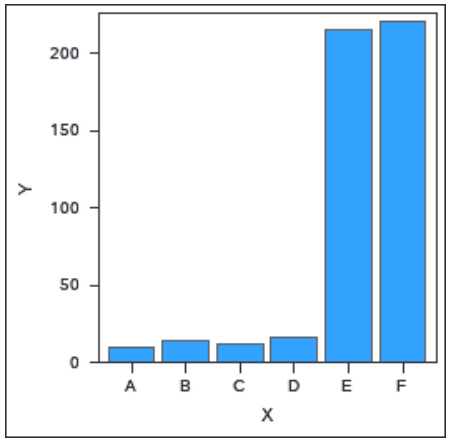
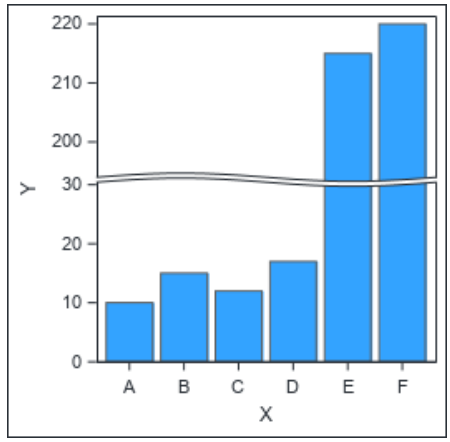
- a number (linear axis only).
- the keyword MIN specifies the minimum data value.
- a SAS time, date, or date-time constant (time axis only).

end
specifies the end of a range.

End can be one of the following:

- a number (linear axis only).
- the keyword MAX specifies the maximum data value.
- a SAS time, date, or date-time constant (time axis only).

The following figure shows a linear axis, broken into ranges 0–30 and 195–220. Although the figure shows the split range on a vertical axis, the same concept applies to a horizontal axis.

RANGES Not Specified	RANGES Specified
	

As shown in the figure, break lines are drawn to indicate the break in the axis.

Restrictions

This option is valid for linear and time axes only.

Only one axis can be broken. If this option is specified for both axes, then it is honored for the vertical axis and ignored for the horizontal axis.

If this option is specified for both axes in the same direction (X and X2 or Y and Y2), then only the primary axis is broken.

When plots are associated with both the X and X2 axes or with both the Y and Y2 axes, neither axis can be broken.

Requirements

All of the ranges must be enclosed in parentheses.

You must specify each range as a starting value, a hyphen, and an ending value. You must separate adjacent ranges with a space.

Interactions

When this option is specified, axis options THRESHOLDMIN=, THRESHOLDMAX=, MIN=, and MAX= are ignored.

When this option is specified, data tips are not displayed, and the plot URL= option is ignored.

When data labels are used in the graph, the data label font size might be reduced in order to avoid overlapping labels and markers. When a range is specified, the data label font size is not scaled during label placement.

Notes

When this option is specified, data-clipping might occur for the following graphics elements: plot markers and marker characters, box plot outlier markers, fixed-position data labels, needle fringe plots, reference lines and drop lines on the broken axis, axis tables, and relative bubble plots.

Curve label positions are based on the non-broken axis data range. When curve labels are specified with a broken axis, the curve label positions might not be ideal.

Tip To control the axis break symbol, use the AXISBREAK= option in the STYLEATTR statement.

Examples
```
ranges=(10-500 1000-5000 10000-50000)

ranges=('01Jan2001'd-'01May2003'd '01Jan2005'd-'01Oct2005'd)
```

REFTICKS
REFTICKS=(*options*)
adds tick marks to the axis that is opposite from the specified axis. You can also specify *options*:

LABEL
in addition to the tick marks, displays the axis label.

NOTICKS
suppresses the tick marks.

VALUES
in addition to the tick marks, displays the values that are represented by the tick marks.

Note This option has no effect if the target axis already contains data.

REVERSE
specifies that the tick values are displayed in reverse (descending) order.

SPLITCHAR="*character-list*"
splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect. When FITPOLICY=SPLIT is in effect, the text is split on occurrences of the specified split character or characters where it is required to fit the tick marks. When FITPOLICY=SPLITALWAYS is in effect, the text is split on all occurrences of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Restriction This option has no effect unless the axis is discrete.

Interactions This option has no effect unless FITPOLICY= is specified as either SPLIT or SPLITALWAYS.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Default RIGHT

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THRESHOLDMAX=*numeric-value*
Specifies a threshold for displaying one more tick mark at the high end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

THRESHOLDMIN=*numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

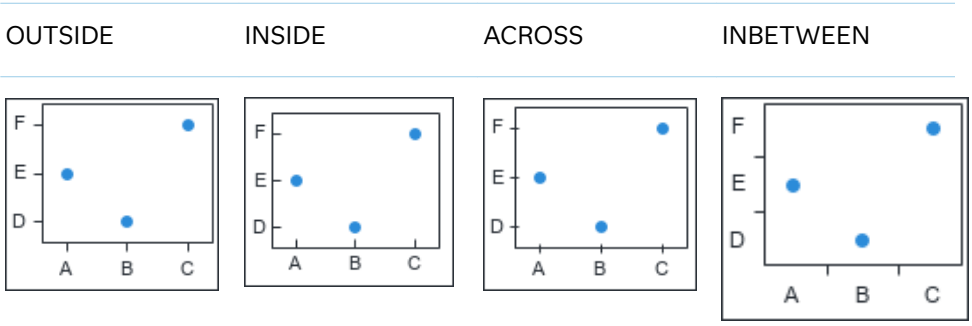
Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

TICKSTYLE=OUTSIDE | INSIDE | ACROSS | INBETWEEN

specifies the placement of the major tick marks in relation to the axis line. The figure shows the tick display for each value.



OUTSIDE

displays the major tick marks outside of the axis frame.

INSIDE

displays the major tick marks inside the axis frame.

ACROSS

displays the major tick marks across the axis line.

INBETWEEN

displays the major tick marks between the discrete data values and outside of the axis frame.

Interaction This value is available only with discrete axes. The option is ignored if specified with other axis types.

Default OUTSIDE

Restriction This option applies to the major tick marks only. It does not affect the minor tick marks.

Interaction This option has effect only if the DISPLAY= option setting includes tick marks, which it does by default.

Note This option has no effect on the placement of the tick values, which are always outside the axis frame.

Tip You can use the SG annotation facility to draw inside minor tick marks in your graph. See Part 7, "SG Annotation," on page 1933. The post Creating an axis in GTL with both major and minor tick marks shown inside on the Graphics Programming community page provides an example for the SGPLOT procedure.

TYPE=DISCRETE | LINEAR | LOG | TIME
specifies the type of axis. Specify one of the following values:

DISCRETE
specifies an axis with discrete values.

If a character variable is assigned to an axis, then the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable. Bar charts support a linear category axis.

LINEAR
specifies a linear scale for the axis.

This is the default axis type for numeric variables, except when the data is discrete, or when the numeric variable has a date or time format.

LOG
specifies a logarithmic scale for the axis.

This axis type is never a default.

Restriction A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

Interactions Use the LOGSTYLE= option to specify the scale and format for the tick values.

Use the LOGBASE= option to specify the base value.

Use the LOGVTYPE= option to specify how the values that are provided in the VALUES= option and the MIN= and MAX= options are interpreted.

TIME
specifies a time scale for the axis.

If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by specifying specific options. If you

specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

For ungrouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

VALUES=(*values-list*) | ("*string-list*")

specifies the values for the ticks on an axis. The syntax for this option varies depending on the type of axis.

- VALUES=(*values-list*) specifies tick values for linear, time, and logarithmic axes.
- VALUES=(*"string-list"*) specifies tick values for discrete axes. The values can be character or numeric.

values-list

specifies tick values for linear, time, and logarithmic axes.

Values in *values-list* can be specified as a simple list of values in the following form:

value <...*value-n*>

The values must be listed in either ascending or descending order. The values can also be specified as a range of values of a specific increment in the following form:

value-1 TO *value-2* BY *increment-value*

By default the increment value is 1. You can use a negative integer for increment to specify a value list in descending order.

You can use both forms when specifying values:

<*value* ... *value-n* >*value-1* TO *value-2* BY *increment-value* <*value* ... *value-n*>

In all forms, multiple *n* values can be separated by blanks or commas. Here are some examples:

- Place tick marks at 2, 4, and 6 on a linear axis:

```
values=(2 4 6)
```

- Place tick marks at 6, 4, and 2 on a linear axis:

```
values=(6,4,2)
```

- Place tick marks at 25MAY08, 04JUL08, and 23AUG08 on a time axis:

```
values=("25MAY08"d "04JUL08"d "23AUG08"d)
```

- Place tick marks from 2 to 10 at an increment of 2 on a linear axis:

```
values=(2 to 10 by 2)
```

- Place tick marks from 50 to 10 at an increment of -5 on a linear axis:

```
values=(50 to 10 by -5)
```

If the specified range is not evenly divisible by the increment value, the highest value displayed on the axis is the last incremental value below the ending value for the range. For example, this value list produces a maximum axis value of 9:

```
values=(0 to 10 by 3)
```

Restrictions This option has no effect on discrete axes.

If your VALUES= option creates more than 1000 values, then the option has no effect.

Interactions For logarithmic axes, use the LOGVTYPE option to control whether the values are expanded or interpreted as exponents.

If a custom format is applied to the value, the raw value is plotted first and then formatted.

Tips Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

For graphs such as histograms, the final bin might be truncated at the maximum data value. If you need to expand the axis in order to show all of the data, use the VALUESHINT option.

"string-list"

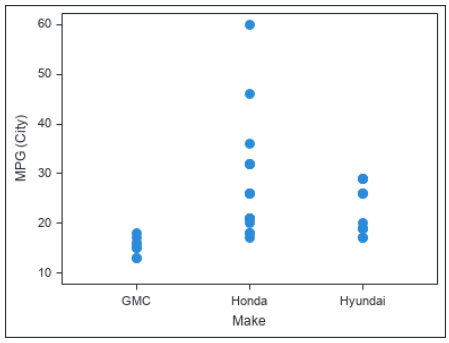
specifies a space-separated list of string values enclosed in parentheses for a discrete axis.

Each value in the list must be enclosed in quotation marks. Numeric values must also be enclosed in quotation marks.

Only the tick values that are included in the string list are displayed on the axis. The values are displayed in the order in which they are listed. The data values that are not in the list are dropped. The list can be a subset of the data values. It can also contain values that are not included in the actual data. A tick value that is not included in the data appears on the axis, but no data is represented at its tick mark.

The following example specifies the values for an X axis. Similar values can be provided for the Y axis.

Table 10.65 *Axis with Specified Character Values*




```
proc sgplot data=sashelp.cars;
xaxis values=("GMC" "Honda" "Hyundai")
   offsetmin=0.2 offsetmax=0.2;
scatter x=make y=mpg_city;
run;
```

Restrictions

This option has no effect on linear, time, or logarithmic axes.

Each value must be enclosed in quotation marks and separated from adjacent values by a blank space.

Notes

If the string list contains duplicate values, the first occurrence of the duplicated value in the list is honored while the remaining instances are ignored.

The axis data can be character or numeric.

If a numeric variable has an associated format, the specified values must use the same format.

Tip

Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

Examples

The following example specifies the axis tick values Sedan, Sports, Wagon, and SUV:

```
values=("Sedan" "Sports" "Wagon" "SUV")
```

The following example specifies the axis tick values 10, 20, 30, and 40:

```
values=("10" "20" "30" "40")
```

The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

TIP Along the axis, numeric tick values are arranged in ascending order while character values are arranged in the order in which they are used in the graph. In some cases, the resulting tick-value order might not be desirable, especially if the graph consists of multiple plots or if the data contains missing values. This option is useful in that case. You can use this option to set the order of the axis tick values.

VALUESDISPLAY=
specifies the text that is to be displayed for the tick values that are defined in the VALUES= option. The list of values must be enclosed in parentheses. Each value must be enclosed in quotation marks and separated from adjacent values by a blank space. Numeric values must also be enclosed in quotation marks.

Restriction This option applies only to discrete and linear axes.

Interaction This option should be used with the VALUES= option. The number of items in the list for this option should equal the number of items in the list for the VALUES= option.

Example The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

VALUESFORMAT=DATA | *SAS-format*
specifies how to format the values for major tick marks. This option supports discrete and logarithmic axes.

DATA
uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

SAS-format
specifies a format to apply to the major tick values.

Restriction This option currently honors most, but not every, SAS format. For more information, see "Format Support in ODS Graphics" in *SAS Graph Template Language: Reference*.

Notes If you specify a format that significantly reduces precision, then, because of tick-value rounding, the plot data elements might not align properly with the axis tick values. In that case, specify a tick-value format with a higher precision.

A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F5.1

Alias TICKVALUEFORMAT

Restriction For discrete axes, only character formats are supported.

Interactions This option is ignored when LOGSTYLE=LOGEXPONENT.

When LOGSTYLE =LOGEXPAND, this option is honored for the base 10 and base 2 logarithmic scales, and is ignored for the base E scale.

When LOGSTYLE =LINEAR, this option is honored for the base 10, base 2, and base E logarithmic scales.

VALUESHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment for all of the tick values that are displayed on the axis.

Default RIGHT

Interaction This option is ignored when FITPOLICY= is specified as either SPLIT or SPLITALWAYS. To align split tick mark values, use the SPLITJUSTIFY= option.

VALUESHINT
specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option. Tick values specified by the VALUES= option are treated as suggestions for the tick values that actually appear, so that the actual data range of the axis is preserved. If the *values-list* contains values that are less than the data minimum, or greater than the data maximum, then those values are ignored.

Interaction This option has no effect unless you also specify the VALUES= option.

Y2AXIS Statement

Specifies the axis options for the Y2 axis, which is displayed on the right side of the graph.

Notes: All Y2AXIS options should be included in a single Y2AXIS statement. Otherwise, some axis options might not be applied.

You can control the features of the axis (for example, the axis label, grid lines, and minor tick marks). You can also control the structure of the axis (for example, the data range, data type, and tick mark values).

Syntax

Y2AXIS *options*;

Summary of Optional Arguments

Appearance options

COLORBANDS=NONE | EVEN | ODD
specifies the display of alternating wall-color bands corresponding to the discrete axis bins.

COLORBANDSATTRS=*style-element*<(***options***)> | (*options*)
specifies the fill appearance of the color band.

DISPLAY=ALL | NONE | (*options*)

specifies which features of the axis are displayed.

GRID
creates grid lines at each tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the grid lines.

MINORGRID
creates grid lines at each minor tick on the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the minor grid lines.

TICKSTYLE=OUTSIDE | INSIDE | ACROSS | INBETWEEN
specifies the placement of the major tick marks in relation to the axis line.

Axis options

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED
specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis.

INTEGER
specifies that only integers are used for tick mark values.

INTERVAL=*time-interval*
specifies the tick interval for a time axis.

INTERVALMULT=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the axis.

LOGBASE=2 | 10 | e
specifies the base value for the logarithmic scale.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
specifies how to scale and format the values for the major tick marks for logarithmic axes.

LOGVTYPE=EXPANDED | EXPONENT
specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options.

MAX=*numeric-value*
specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

MIN=*numeric-value*
specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

MINOR
adds minor tick marks to a linear, log, or time axis.

MINORCOUNT=*numeric-value*
specifies the number of minor tick marks for the axis.

MINORINTERVAL=*time-interval*
specifies the time interval between minor ticks.

NOTIMESPLIT
prevents a time axis from splitting the time, date, or datetime values into two rows.

OFFSETMAX=*numeric-value*

specifies an offset that follows the highest data value on the axis.

OFFSETMIN=*numeric-value*
specifies an offset that precedes the lowest data value on the axis.

RANGES=(*start–end <start2–end2 startN–endN ...>*)
specifies the ranges for a broken axis.

REFTICKS

REFTICKS=(*options*)
adds tick marks to the axis that is opposite from the specified axis.

REVERSE
specifies that the tick values are displayed in reverse (descending) order.

THRESHOLDMAX=*numeric-value*
Specifies a threshold for displaying one more tick mark at the high end of the axis.

THRESHOLDMIN=*numeric-value*
Specifies a threshold for displaying one more tick mark at the low end of the axis.

TYPE=DISCRETE | LINEAR | LOG | TIME
specifies the type of axis.

Text options

FITPOLICY=NONE | SPLIT | SPLITALWAYS | THIN
specifies the method that is used to fit tick mark values on a vertical axis when there is not enough room to draw them normally.

LABEL="*text-string*"
specifies a label for the axis.

LABELATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the axis labels.

LABELPOS=BOTTOM | CENTER | DATACENTER | TOP
specifies the position of the axis label.

SPLITCHAR="*character-list*"
splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the axis tick value labels.

VALUES=(*values-list*) | ("*string-list*")
specifies the values for the ticks on an axis.

VALUESDISPLAY=
specifies the text that is to be displayed for the tick values that are defined in the VALUES= option.

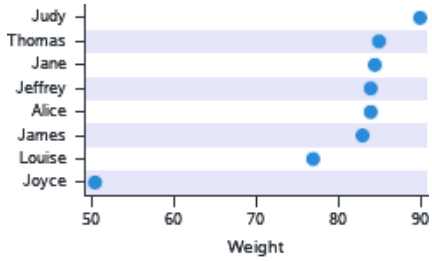
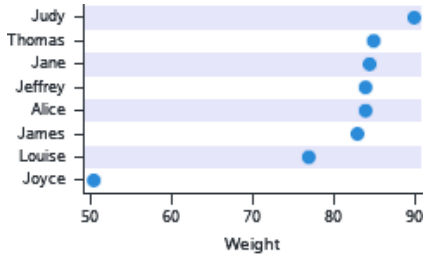
VALUESFORMAT=DATA | *SAS-format*
specifies how to format the values for major tick marks.

VALUESHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment for all of the tick values that are displayed on the axis.

VALUESHINT
specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option.

Optional Arguments

COLORBANDS=NONE | EVEN | ODD
specifies the display of alternating wall-color bands corresponding to the discrete axis bins. The following images show the results of ODD and EVEN settings:

COLORBANDS=ODD	COLORBANDS=EVEN
	

Default NONE

Restriction This option applies to discrete axes only.

Interaction Specifying this option for more than one axis in the graph might have unexpected results. The order in which color bands are drawn might not match the order in which the axis options are specified.

Note The full width of a color band is the distance between midpoints. When no axis offsets are specified, the first band begins at one-half of the midpoint distance, and the last band ends at one-half of the midpoint distance. When axis offsets are specified, the first and last color bands on the axis might extend into their adjacent offsets by as much as half the color-band width.

Tip Use the COLORBANDSATTRS= option to customize the color bands.

COLORBANDSATTRS=*style-element*<(*options*)> | (*options*)
specifies the fill appearance of the color band. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of fill attributes, see "Fill Attributes" on page 1832.

Interaction This option has no effect unless COLORBANDS= is also specified.

DISCRETEORDER=DATA | FORMATTED | UNFORMATTED
specifies the order in which discrete tick values are placed on the axis. This option affects any plot with a discrete axis. Specify one of the following values:

DATA
places the values in the order in which they appear in the data.

Note: This value is not supported with data that is stored in a table.

FORMATTED
sorts the formatted values in ascending character order.

UNFORMATTED
sorts the unformatted values in ascending character order.

Default UNFORMATTED

DISPLAY=ALL | NONE | (*options*)
specifies which features of the axis are displayed.

ALL
displays all of the features of the axis.

NONE
specifies that none of the features of the axis are displayed.

(*options*)
specifies one or more axis features to hide.

Specify one or more of the following:

NOLABEL
hides the axis label

NOLINE
hides the axis line

Tips This value hides the axis line but has no effect on the graph's wall border, part of which occupies the same area as the axis line. To hide the wall border, specify NOBORDER in the PROC SGPLOTstatement.

This value has no effect on baselines. For plots that support a baseline, such as bar charts and needle plots, you might need to suppress the baseline. In the plot statement, use the BASELINEATTRS= option to set the line thickness to 0.

NOTICKS
hides the tick marks on the axis

NOVALUES
hides the tick mark values on the axis

Default ALL

Example DISPLAY=(NOTICKS NOVALUES)

FITPOLICY=NONE | SPLIT | SPLITALWAYS | THIN
specifies the method that is used to fit tick mark values on a vertical axis when there is not enough room to draw them normally. Select one of the following values:

NONE
does not split the values.

SPLIT
splits the values at the character or characters specified in the SPLITCHAR= option.

No split occurs at split characters that occur where a split is not needed. In that case, the split character is displayed with the text value. If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Restriction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

SPLITALWAYS
always splits the values at the character or characters specified in the SPLITCHAR= option.

If the value does not contain any of the specified split characters, a split does not occur.

Default The default split character is a space.

Restriction This option has no effect unless the axis is discrete.

Tip You can specify the split character using the SPLITCHAR= option.

Example "Fit Policies for Axes" on page 1803

THIN
removes some of the values from the axis.

Default THIN

See "Fit Policies for Axes" on page 1803

GRID
creates grid lines at each tick on the axis.

Interaction Grid lines are not displayed when you specify the COLORBANDS= option. The color bands take the place of grid lines.

Tip You can specify the MINORGRID option to create grid lines at each minor tick on the axis.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless GRID is also specified.

Examples
```
GRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:
```
GRIDATTRS=GraphAxisLines
```

INTEGER
specifies that only integers are used for tick mark values. This option affects only linear axes.

INTERVAL=*time-interval*
specifies the tick interval for a time axis. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

Specify one of the following values:

Table 10.66 *Time Intervals*

INTERVAL	Unit	Tick interval	Default tick value format
AUTO	DATE, TIME, or DATETIME	automatically chosen	automatically chosen
SECOND	TIME or DATETIME	second	TIME8.
MINUTE	TIME or DATETIME	minute	TIME8.
HOUR	TIME or DATETIME	hour	TIME8.
DAY	DATE or DATETIME	day	DATE9.
TENDAY	DATE or DATETIME	10 days	DATE9.
WEEK	DATE or DATETIME	7 days	DATE9.

INTERVAL	Unit	Tick interval	Default tick value format
SEMIMONTH	DATE or DATETIME	1st and 16th of each month	DATE9.
MONTH	DATE or DATETIME	month	MONYY7.
QUARTER	DATE or DATETIME	3 months	YYQC6.
SEMIYEAR	DATE or DATETIME	6 months	MONYY7.
YEAR	DATE or DATETIME	year	YEAR4.

Default AUTO

Tip If unwanted thinning of tick marks occurs when you specify INTERVAL=, try using minor intervals instead. For example, you might specify MINOR MINORINTERVAL=MONTH instead of INTERVAL=MONTH. As a result, in the example monthly tick marks are displayed even though the major values are thinned.

INTERVALMULT=*positive-integer*
specifies a multiplier to apply to the time interval that is in effect for the axis.

Alias INTERVALMULTIPLIER

Default 1

Restriction This option applies to time axes only.

Interaction This option is ignored if VALUES=*values-list* is used.

Tip Use the INTERVAL= option to specify a different time interval.

Examples To specify 3-month intervals:

```
INTERVAL=MONTH INTERVALMULT=3
```

To specify 10-year intervals:

```
INTERVAL=YEAR INTERVALMULT=10
```

LABEL="*text-string*"
specifies a label for the axis.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style

element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphLabelText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
LABELATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
LABELATTRS=GraphTitleText
```

LABELPOS=BOTTOM | CENTER | DATACENTER | TOP
specifies the position of the axis label.

BOTTOM
positions the label at the bottom of the axis area.

The label is oriented horizontally (unrotated).

The label is right-justified in the axis area for the Y axis and left-justified for the Y2 axis. If there is not sufficient room in the axis area to display the label, the label grows to the right for the Y axis and to the left for the Y2 axis.

CENTER
centers the axis label in the axis area (including any offsets).

The label is oriented vertically (rotated).

The label is positioned to the left of the tick values for the Y axis or to the right of the axis values for the Y2 axis.

DATACENTER
centers the axis label in the axis tick display area (excluding any offsets).

The label is oriented vertically (rotated).

The label is positioned to the left of the tick values for the Y axis or to the right of the axis values for the Y2 axis.

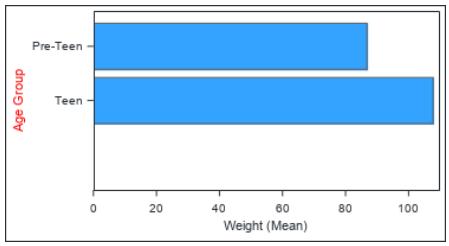
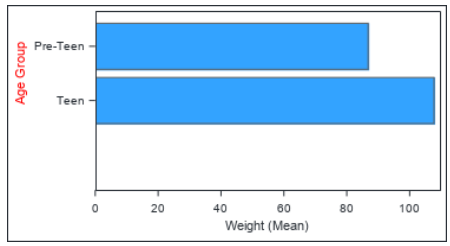
TOP
positions the label at the top of the axis area.

The label is oriented horizontally (unrotated).

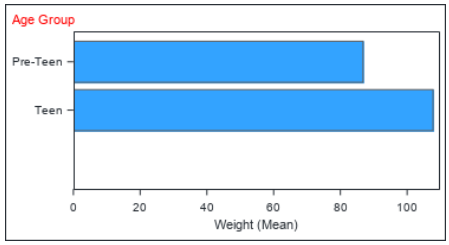
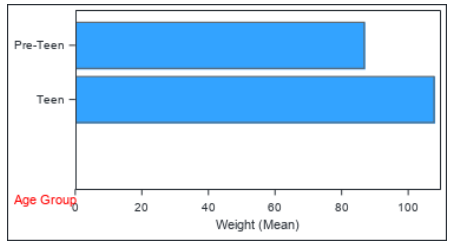
The label is right-justified in the axis area for the Y axis and left-justified for the Y2 axis. If there is not sufficient room in the axis area to display the label, the label grows to the right for the Y axis and to the left for the Y2 axis.

The following figure shows the CENTER and DATACENTER positions for the red Y-axis label “Age Group.” An axis offset is applied to the maximum end of the axis in order to demonstrate the difference between CENTER and DATACENTER. (For HBAR plot statements, the Y axis is reversed by default. The offset is created at the bottom end of the Y axis.)

In the example, CENTER centers the labels on the entire axis area, including the offset. DATACENTER centers the labels on the tick display areas, which does not include the offset.

CENTER	DATACENTER
	

The next figure shows the TOP and BOTTOM positions for the same axis labels.

TOP	BOTTOM
	

Default CENTER

LOGBASE=2 | 10 | e
specifies the base value for the logarithmic scale.

Default 10

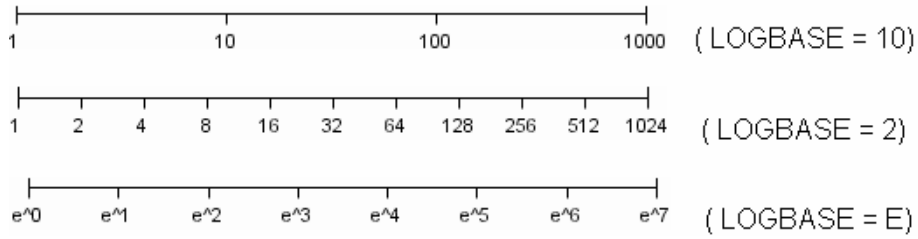
Interaction This option has no effect unless you also specify TYPE=LOG.

LOGSTYLE=LINEAR | LOGEXPAND | LOGEXPONENT
specifies how to scale and format the values for the major tick marks for logarithmic axes. Specify one of the following values:

LOGEXPAND
places the tick marks at integer powers of the base.

For example, if you specified `LOGBASE=2`, the tick marks might be at 1, 2, 4, 8, 16. See Figure 10.41 on page 1621 .

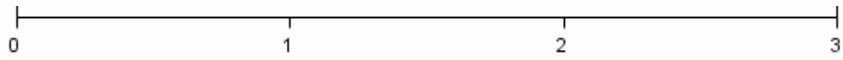
Figure 10.19 *Graph Axes with LOGEXPAND*



LOGEXPONENT
places the tick marks at integer powers of the base, but identifies the values by the exponent.

For example, if you specified **LOGBASE=10**, the tick marks might be at 1, 10, 100, 1000, but the tick values would read 0, 1, 2, 3. See Figure 10.42 on page 1621 .

Figure 10.20 *An Axis with LOGEXPONENT*



LINEAR
places the tick marks at uniform linear intervals, but spaces them logarithmically.

In some cases an intermediate tick mark is placed between the first and second marks.

For example, if the data on this axis range from 14 to 1154, and you specify **LOGBASE=10**, then the tick marks might be at 10, 40, 200, 400, 600, 800, 1000, 1200. See Figure 10.43 on page 1621 .

Figure 10.21 *An Axis with LINEAR*

Default LOGEXPAND

Interaction This option has no effect unless you also specify TYPE=LOG.

LOGVTYPE=EXPANDED | EXPONENT
specifies the scale that is used when interpreting the values in the VALUES option and the MIN and MAX options. This option enables you to choose your preferred way of specifying log-axis values regardless of the LOGSTYLE= option value.

Specify one of the following values:

EXPANDED
the values are interpreted as integer powers of the base (decimal numbers).

EXPONENT
the values are interpreted as integer exponents of the base.

Default EXPANDED

Interaction This option has no effect unless you also specify TYPE=LOG. You must also specify values for the VALUES= option or the MIN= and MAX= options or all of them.

Tip This option is particularly useful when the log axis is an odd base (such as base E) or the axis log style is EXPONENT.

Examples The following example specifies MIN= and MAX= as exponent values instead of expanded values on an expanded Base 10 log axis. This results in Y-axis tick values of 10, 100, 1000, 10000, and 100000.

```
yaxis type=log logbase=10 logstyle=logexpand
   logvtype=exponent
   min=1 max=5;
```

The following example specifies VALUES= as a list of expanded values instead of exponent values on an exponent Base 10 log axis. This results in X-axis tick values of 1, 2, 3, 4, and 5.

```
xaxis type=log logbase=10 logstyle=logexponent
   logvtype=expanded
   values=(10 100 1000 10000 100000);
```

MAX=*numeric-value*
specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES=option and you do not also specify the VALUESHINT option.

This option does not determine the maximum axis tick value displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the maximum value is expanded or interpreted as an exponent.

Tip The maximum axis tick value might differ from the MAX= value. The MAX= and MIN= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values. To display the MAX= value as the maximum tick value, use the VALUES= option.

MIN=*numeric-value*
specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).

Restriction This option affects linear, log, and time axes only.

Interactions This option has no effect if you specify the VALUES= option and you do not also specify the VALUESHINT option.

This option does not determine the minimum axis tick value displayed. The THRESHOLDMIN= value is used to determine the minimum tick value.

For logarithmic axes, use the LOGVTYPE option to control whether the minimum value is expanded or interpreted as an exponent.

Tip The minimum axis tick value might differ from the MIN= value. The MIN= and MAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values. To display the MIN= value as the minimum tick value, use the VALUES= option.

MINOR
adds minor tick marks to a linear, log, or time axis.

Restriction This option has no effect on discrete axes.

Interaction This option has no effect if you specify the VALUES= option.

Tip Use MINORCOUNT= to specify the number of tick marks.

MINORCOUNT=*numeric-value*
specifies the number of minor tick marks for the axis. This value determines the number of minor tick marks for each interval on the axis.

Restriction This option applies to linear and log axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

MINORGRID
creates grid lines at each minor tick on the axis.

Interaction This option has no effect unless GRID is also specified for the axis.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the minor grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphMinorGridLines style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style.

Interaction This option has no effect unless MINORGRID is also specified.

Tip You can use GRIDATTRS= to change the appearance of the major grid lines.

Examples
```
MINORGRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:
```
MINORGRIDATTRS=GraphAxisLines
```

MINORINTERVAL=*time-interval*
specifies the time interval between minor ticks. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, you must select *AUTO, SECOND, MINUTE*, or *HOUR*.

For information about the intervals that you can select, see Table 10.145 on page 1617.

Default AUTO

Restriction This option applies to time axes only.

Note This option does not automatically add minor tick marks to the axis. Use the MINOR option to add tick marks.

NOTIMESPLIT
prevents a time axis from splitting the time, date, or datetime values into two rows.

Restriction This option applies to time axes only.

OFFSETMAX=*numeric-value*
specifies an offset that follows the highest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the maximum end of the axis to the highest data value or highest tick value, whichever is greater.

For a discrete axis, the offset is measured from the maximum end of the axis to the midpoint of the last category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the maximum end of the axis. The sum of the minimum and maximum offsets does not exceed 50% of the axis height.

Interaction For HBOX, HBAR, HLINE, and DOT plot statements, the Y axis is reversed by default, so the axis origin is at the top.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See “Adjusting Axis Offsets” on page 1788

OFFSETMIN=*numeric-value*
specifies an offset that precedes the lowest data value on the axis. Specify a value between 0 and 1.

The value represents the offset as a proportion to the total length of the axis. For a continuous axis, the offset is measured from the beginning of the axis to the lowest data value or lowest tick value, whichever is less.

For a discrete axis, the offset is measured from the beginning of the axis to the midpoint of the first category.

Default Just enough area is reserved automatically to fully display markers and other graphical features near the maximum end of the axis. The sum of the minimum and maximum offsets does not exceed 50% of the axis height.

Interaction For HBOX, HBAR, HLINE, and DOT plot statements, the Y axis is reversed by default, so the axis origin is at the top.

Note The sum of OFFSETMIN= and OFFSETMAX= must not exceed 1. If the sum exceeds 1, the OFFSETMIN= and OFFSETMAX= options are ignored, and the defaults are used instead.

See "Adjusting Axis Offsets" on page 1788

RANGES=(*start–end <start2–end2 startN–endN ...>*)
specifies the ranges for a broken axis.

start
specifies the start of a range.

Start can be one of the following:

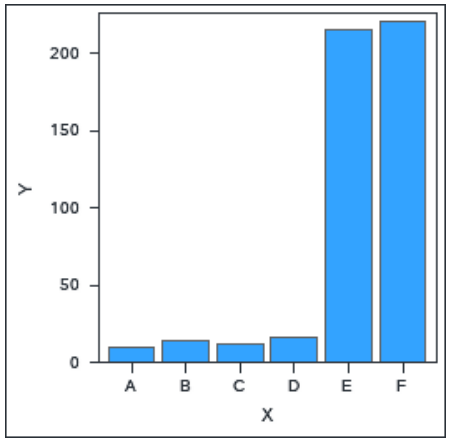
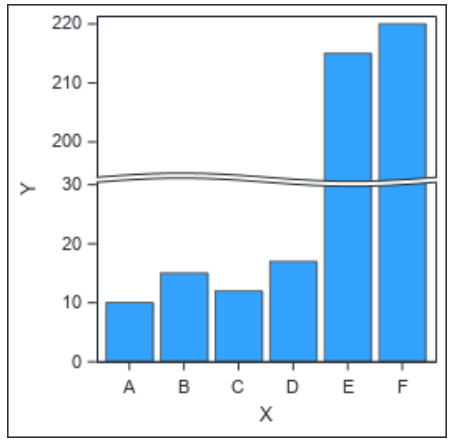
- a number (linear axis only).
- the keyword MIN specifies the minimum data value.
- a SAS time, date, or date-time constant (time axis only).

end
specifies the end of a range.

End can be one of the following:

- a number (linear axis only).
- the keyword MAX specifies the maximum data value.
- a SAS time, date, or date-time constant (time axis only).

The following figure shows a linear axis, broken into ranges 0–30 and 195–220. Although the figure shows the split range on a vertical axis, the same concept applies to a horizontal axis.

RANGES Not Specified	RANGES Specified
	

As shown in the figure, break lines are drawn to indicate the break in the axis.

Restrictions

This option is valid for linear and time axes only.

Only one axis can be broken. If this option is specified for both axes, then it is honored for the vertical axis and ignored for the horizontal axis.

If this option is specified for both axes in the same direction (X and X2 or Y and Y2), then only the primary axis is broken.

When plots are associated with both the X and X2 axes or with both the Y and Y2 axes, neither axis can be broken.

Requirements

All of the ranges must be enclosed in parentheses.

You must specify each range as a starting value, a hyphen, and an ending value. You must separate adjacent ranges with a space.

Interactions

When this option is specified, axis options THRESHOLDMIN=, THRESHOLDMAX=, MIN=, and MAX= are ignored.

When this option is specified, data tips are not displayed, and the plot URL= option is ignored.

When data labels are used in the graph, the data label font size might be reduced in order to avoid overlapping labels and markers. When a range is specified, the data label font size is not scaled during label placement.

Notes

When this option is specified, data-clipping might occur for the following graphics elements: plot markers and marker characters, box plot outlier markers, fixed-position data labels, needle fringe plots, reference lines and drop lines on the broken axis, axis tables, and relative bubble plots.

Curve label positions are based on the non-broken axis data range. When curve labels are specified with a broken axis, the curve label positions might not be ideal.

Tip To control the axis break symbol, use the AXISBREAK= option in the STYLEATTR statement.

Examples

```
ranges=(10-500 1000-5000 10000-50000)
```

```
ranges=('01Jan2001'd-'01May2003'd '01Jan2005'd-'01Oct2005'd)
```

REFTICKS
REFTICKS=(*options*)
adds tick marks to the axis that is opposite from the specified axis. You can also specify *options*:

LABEL
in addition to the tick marks, displays the axis label.

NOTICKS
suppresses the tick marks.

VALUES
in addition to the tick marks, displays the values that are represented by the tick marks.

Note This option has no effect if the target axis already contains data.

REVERSE
specifies that the tick values are displayed in reverse (descending) order.

SPLITCHAR="*character-list*"
splits the text for tick mark values at the specified character or characters according to the fit policy that is in effect. When FITPOLICY=SPLIT is in effect, the text is split on occurrences of the specified split character or characters where it is required to fit the tick marks. When FITPOLICY=SPLITALWAYS is in effect, the text is split on all occurrences of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Restriction This option has no effect unless the axis is discrete.

Interactions This option has no effect unless FITPOLICY= is specified as either SPLIT or SPLITALWAYS.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Default LEFT

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

THRESHOLDMAX=*numeric-value*
Specifies a threshold for displaying one more tick mark at the high end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

THRESHOLDMIN=*numeric-value*

Specifies a threshold for displaying one more tick mark at the low end of the axis.

Default 0.30

Range 0 to 1

Restriction This option applies to linear axes only.

Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

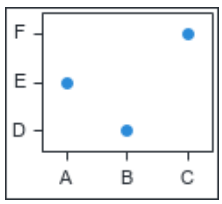
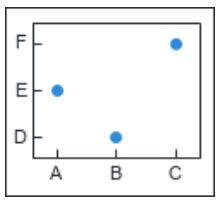
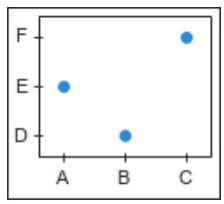
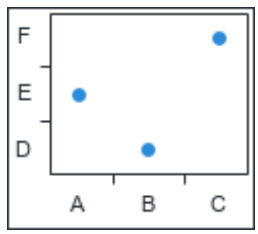
Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= option and the THRESHOLDMAX= option to 0.

TICKSTYLE=OUTSIDE | INSIDE | ACROSS | INBETWEEN

specifies the placement of the major tick marks in relation to the axis line. The figure shows the tick display for each value.

Note: Although the figure shows tick displays for the primary axes, the same placement in relation to the axis applies when secondary axes are used.

OUTSIDE	INSIDE	ACROSS	INBETWEEN
			

OUTSIDE

displays the major tick marks outside of the axis frame.

INSIDE

displays the major tick marks inside the axis frame.

ACROSS

displays the major tick marks across the axis line.

INBETWEEN

displays the major tick marks between the discrete data values and outside of the axis frame.

Interaction This value is available only with discrete axes. The option is ignored if specified with other axis types.

Default OUTSIDE

Restriction This option applies to the major tick marks only. It does not affect the minor tick marks.

Interaction This option has effect only if the DISPLAY= option setting includes tick marks, which it does by default.

Note This option has no effect on the placement of the tick values, which are always outside the axis frame.

Tip You can use the SG annotation facility to draw inside minor tick marks in your graph. See Part 7, "SG Annotation," on page 1933. The post Creating an axis in GTL with both major and minor tick marks shown inside on the Graphics Programming community page provides an example for the SGPLOT procedure.

TYPE=DISCRETE | LINEAR | LOG | TIME

specifies the type of axis. Specify one of the following values:

DISCRETE

specifies an axis with discrete values.

If a character variable is assigned to an axis, then the default type for that axis is discrete. In addition, all categorization plots use a discrete axis for the category variable. Bar charts support a linear category axis.

LINEAR

specifies a linear scale for the axis.

This is the default axis type for numeric variables, except when the data is discrete, or when the numeric variable has a date or time format.

LOG

specifies a logarithmic scale for the axis.

This axis type is never a default.

Restriction A logarithmic scale cannot be used with linear regression plots (REG statement where DEGREE=1).

Interactions Use the LOGSTYLE= option to specify the scale and format for the tick values.

Use the LOGBASE= option to specify the base value.

Use the LOGVTYPE= option to specify how the values that are provided in the VALUES= option and the MIN= and MAX= options are interpreted.

TIME

specifies a time scale for the axis.

If the variable assigned to an axis has a time, date, or datetime format associated with it, then time is the default axis type.

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis tick value labels. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

For ungrouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

VALUES=(*values-list*) | ("*string-list*")
specifies the values for the ticks on an axis. The syntax for this option varies depending on the type of axis.

- VALUES=(*values-list*) specifies tick values for linear, time, and logarithmic axes.
- VALUES=(*"string-list"*) specifies tick values for discrete axes. The values can be character or numeric.

values-list
specifies tick values for linear, time, and logarithmic axes.

Values in *values-list* can be specified as a simple list of values in the following form:

value <...*value-n*>

The values must be listed in either ascending or descending order. The values can also be specified as a range of values of a specific increment in the following form:

value-1 TO *value-2* BY *increment-value*

By default the increment value is 1. You can use a negative integer for increment to specify a value list in descending order.

You can use both forms when specifying values:

<*value* ... *value-n* >*value-1* TO *value-2* BY *increment-value* <*value* ... *value-n*>

In all forms, multiple *n* values can be separated by blanks or commas. Here are some examples:

- Place tick marks at 2, 4, and 6 on a linear axis:

```
values=(2 4 6)
```

- Place tick marks at 6, 4, and 2 on a linear axis:

```
values=(6,4,2)
```

- Place tick marks at 25MAY08, 04JUL08, and 23AUG08 on a time axis:

```
values=("25MAY08"d "04JUL08"d "23AUG08"d)
```

- Place tick marks from 2 to 10 at an increment of 2 on a linear axis:

```
values=(2 to 10 by 2)
```

- Place tick marks from 50 to 10 at an increment of -5 on a linear axis:

```
values=(50 to 10 by -5)
```

If the specified range is not evenly divisible by the increment value, the highest value displayed on the axis is the last incremental value below the ending value for the range. For example, this value list produces a maximum axis value of 9:

```
values=(0 to 10 by 3)
```

Restrictions This option has no effect on discrete axes.

If your VALUES= option creates more than 1000 values, then the option has no effect.

Interactions For logarithmic axes, use the LOGVTYPE option to control whether the values are expanded or interpreted as exponents.

If a custom format is applied to the value, the raw value is plotted first and then formatted.

Tips Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

For graphs such as histograms, the final bin might be truncated at the maximum data value. If you need to expand the axis in order to show all of the data, use the VALUESHINT option.

"string-list"

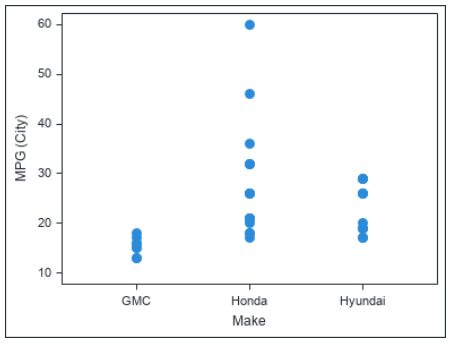
specifies a space-separated list of string values enclosed in parentheses for a discrete axis.

Each value in the list must be enclosed in quotation marks. Numeric values must also be enclosed in quotation marks.

Only the tick values that are included in the string list are displayed on the axis. The values are displayed in the order in which they are listed. The data values that are not in the list are dropped. The list can be a subset of the data values. It can also contain values that are not included in the actual data. A tick value that is not included in the data appears on the axis, but no data is represented at its tick mark.

The following example specifies the values for an X axis. Similar values can be provided for the Y axis.

Table 10.67 *Axis with Specified Character Values*




```
proc sgplot data=sashelp.cars;
xaxis values=("GMC" "Honda" "Hyundai")
   offsetmin=0.2 offsetmax=0.2;
scatter x=make y=mpg_city;
run;
```

Restrictions This option has no effect on linear, time, or logarithmic axes.

Each value must be enclosed in quotation marks and separated from adjacent values by a blank space.

Notes If the string list contains duplicate values, the first occurrence of the duplicated value in the list is honored while the remaining instances are ignored.

The axis data can be character or numeric.

If a numeric variable has an associated format, the specified values must use the same format.

Tip Use the VALUESDISPLAY= option to specify string replacement text for the specified values.

Examples The following example specifies the axis tick values Sedan, Sports, Wagon, and SUV:

```
values=("Sedan" "Sports" "Wagon" "SUV")
```

The following example specifies the axis tick values 10, 20, 30, and 40:

```
values=("10" "20" "30" "40")
```

The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

TIP Along the axis, numeric tick values are arranged in ascending order while character values are arranged in the order in which they are used in the graph. In some cases, the resulting tick-value order might not be desirable, especially if the graph consists of multiple plots or if the data contains missing values. This option is useful in that case. You can use this option to set the order of the axis tick values.

VALUESDISPLAY=
specifies the text that is to be displayed for the tick values that are defined in the VALUES= option. The list of values must be enclosed in parentheses. Each value must be enclosed in quotation marks and separated from adjacent values by a blank space. Numeric values must also be enclosed in quotation marks.

Restriction This option applies only to discrete and linear axes.

Interaction This option should be used with the VALUES= option. The number of items in the list for this option should equal the number of items in the list for the VALUES= option.

Example The following example specifies numeric values of 14 and 15, and then displays them as fourteen and fifteen:

```
values=("14" "15") valuesdisplay=("fourteen" "fifteen")
```

VALUESFORMAT=DATA | *SAS-format*
specifies how to format the values for major tick marks. This option supports discrete and logarithmic axes.

DATA
uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

SAS-format
specifies a format to apply to the major tick values.

Restriction This option currently honors most, but not every, SAS format. For more information, see "Format Support in ODS Graphics" in *SAS Graph Template Language: Reference*.

Notes If you specify a format that significantly reduces precision, then, because of tick-value rounding, the plot data elements might not align properly with the axis tick values. In that case, specify a tick-value format with a higher precision.

A syntax error might be generated if you specify the format using the *w.d* SAS format. To avoid this error condition, insert an F in front of the format name. Example: F5.1

Alias TICKVALUEFORMAT

Restriction For discrete axes, only character formats are supported.

Interactions This option is ignored when LOGSTYLE=LOGEXPONENT.

When LOGSTYLE =LOGEXPAND, this option is honored for the base 10 and base 2 logarithmic scales, and is ignored for the base E scale.

When LOGSTYLE =LINEAR, this option is honored for the base 10, base 2, and base E logarithmic scales.

VALUESHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment for all of the tick values that are displayed on the axis.

Default LEFT

Interaction This option is ignored when FITPOLICY= is specified as either SPLIT or SPLITALWAYS. To align split tick mark values, use the SPLITJUSTIFY= option.

VALUESHINT
specifies that the minimum and maximum axis values are determined independently of the values that you specify in the VALUES= option. Tick values specified by the VALUES= option are treated as suggestions for the tick values that actually appear, so that the actual data range of the axis is preserved. If the *values-list* contains values that are less than the data minimum, or greater than the data maximum, then those values are ignored.

Interaction This option has no effect unless you also specify the VALUES= option.

YAXISTABLE Statement

Creates a basic plot that displays data values at specific locations along the vertical axis.

Requirements: At least one other plot statement must be used in the procedure. This other plot becomes the primary plot. The primary plot provides the default placement of values along the axis table's vertical axis.

For the axis table, you must specify at least one variable. This variable determines which values are displayed in the axis table.

Interactions: You can specify a variable for the Y= option to identify the locations along the axis where the values are displayed. If you do not specify Y=, the locations are determined by the Y variable in the primary plot. When the primary plot is a categorical chart (bar, line, or dot), the category variable from the chart is used. With the exception of that Y (or category) variable and any group variable that might be specified, axis tables are separate plots and are unaware of the options specified in the accompanying plots.

When used with bar charts, line charts, and dot plots, all axis tables must align with the category axis of the chart. If a statement uses the wrong orientation, the statement is rejected with a message in the SAS log. For example, if your procedure has a VBAR statement along with a YAXISTABLE statement, the YAXISTABLE statement is rejected with a message.

Axis tables cannot be used when the primary plot is one of the following plot types: BAND, BLOCK, FRINGE, REG, LOESS, and PBSPLINE. The axis table requires at least one other plot statement that is not one of these types. Otherwise, the axis table is not created and an error is written to the log.

Notes: The data values can be placed inside or outside of the axis.

You can use multiple axis table statements.

Syntax

YAXISTABLE *variable* <...*variable-n*> </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*character-value*
: specifies the value of the ID variable in a discrete attribute map data set.

COLORGROUP=*variable*
: specifies a variable that is used to determine the color of the table values.

DROPONMISSING
: specifies that the entire axis table is dropped when all of the values are missing.

INDENT=*dimension*<*unit*>
: specifies a value to be used with the INDENTWEIGHT= option to determine the indention for each text value.

INDENTWEIGHT=*numeric-variable*
: specifies the indention weight (multiplier) for each observation.

LOCATION=OUTSIDE | INSIDE
: specifies whether the axis table is placed outside or inside the axis area.

NOMISSINGCHAR
: suppresses the display of the MISSING character (.) for missing numeric values.

PAD=*numeric-value*<*units*> | *(pad-options)*
: specifies the amount of extra space that is added inside the table border.

POSITION=LEFT | RIGHT
: specifies the position of the axis table at the left or right side of the graph.

SEPARATOR
: creates a separating line between the axis table or axis tables and the plot.

TEXTGROUP=*attribute-map-group-variable*
: specifies the group variable that is used in a discrete attribute map data set to map text attributes to values for each observation.

TEXTGROUPID=*attribute-map-id*
: specifies an attribute ID for the TEXTGROUP= option.

VALUEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the axis table values.

VALUEHALIGN=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the axis table values relative to the column width in the table.

VALUEJUSTIFY=LEFT | CENTER | RIGHT
: specifies the justification of the axis table values relative to the column width in the table.

Axis options

Y2AXIS
: assigns the table to the secondary vertical axis.

Class options

CLASS=*variable*
: creates a separate axis table for each unique value of the specified variable.

CLASSDISPLAY=STACK | CLUSTER
: specifies how the class values are displayed.

CLASSORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
: specifies the order in which the class values are displayed.

NOMISSINGCLASS
: specifies that missing values of the class variable are not included in the table.

Label options

LABEL

LABEL="*text-string*"

NOLABEL
: specifies whether the table label is shown or hidden.

LABELATTRS=*style-element*<(***options***)> | (*options*)
: specifies the color and font attributes of the axis table label.

LABELHALIGN=LEFT | CENTER | RIGHT
: specifies the horizontal alignment of the column labels, when displayed.

LABELJUSTIFY=LEFT | CENTER | RIGHT
: specifies the justification of the labels.

LABELPOS=BOTTOM | TOP
: specifies the position of the labels at the bottom or top of the axis table.

STATLABEL

NOSTATLABEL
: specifies whether the variable statistic is displayed in the table's label.

Plot options

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM
: specifies the statistic for the axis table.

Y=*variable*
: Specifies the variable to use to align the table values to the Y or Y2 axis.

Plot reference options

NAME="*text-string*"
: assigns a name to a plot statement.

Title options

TITLE="*text-string*"
: specifies a title for the axis table.

TITLEATTRS=*style-element*<(***options***)> | (*options*)
: specifies the appearance of the title for the axis table.

TITLEHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the axis table title bounding box relative to the axis table width.

TITLEJUSTIFY=LEFT | CENTER | RIGHT
specifies the justification of the lines of a multiline title in the title bounding box. The justification is relative to the axis table width.

Required Argument

variable <...variable-n>
specifies one or more variables for the axis table. When multiple variables are specified, the axis tables are placed in columns. Any options that you add to the statement apply to all the variables that are specified in that statement.

Note When the variable specified is a character variable, the first value of each category is displayed in the axis table. When the variable is numeric, the axis table displays the sum statistic. You can modify the statistic using the STAT= option.

Optional Arguments

ATTRID=*character-value*
specifies the value of the ID variable in a discrete attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph.

See Chapter 18, "Using Discrete Attribute Maps," on page 1901

"Overview of Attribute Maps" on page 1897

CLASS=*variable*
creates a separate axis table for each unique value of the specified variable. This option acts as a classification variable for the axis table. Each axis table is labeled by the class value.

Restriction The CLASS option is ignored when the axis table is used with bar, line, or dot charts. If the GROUP= option is specified in the chart, that group variable is used as the CLASS variable for all axis tables.

Interaction If NOLABEL is also specified, then the class labels are removed.

Tips Use the CLASSDISPLAY= option to control whether the class values are clustered or stacked.

Use the CLASSORDER= option to control the order in which the class values are displayed.

CLASSDISPLAY=STACK | CLUSTER
specifies how the class values are displayed.

STACK
displays the class values horizontally at each midpoint value on the Y axis.

CLUSTER
displays the class values vertically at each midpoint value on the Y axis.

Restriction The CLASSDISPLAY=CLUSTER option applies only when the axis table is on a discrete axis.

Default STACK

Interaction For this option to have any effect, the CLASS= option must be specified.

CLASSORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the order in which the class values are displayed.

DATA
displays the class values in the order in which they occur in the data.

REVERSEDATA
displays the class values in the reverse order from which they occur in the data.

Tip This option is useful when the plot axis is reversed.

ASCENDING
displays the class values in ascending order.

DESCENDING
displays the class values in descending order.

Default DATA

Restriction This option is ignored when the axis table is used with DOT, HBAR, VBAR, HLINE, and VLINE statements. If the GROUPORDER= option is specified in the chart, then that group order variable is used as the class order variable for all axis tables.

Interaction For this option to have any effect, the CLASS= option must be specified.

COLORGROUP=*variable*
specifies a variable that is used to determine the color of the table values. Once the variable values are found, the value colors are taken from the GraphData1–GraphData*n* style elements in the current style. The CONTRASTCOLOR attribute is used for the value text.

Restriction When used with DOT, HBAR, VBAR, HLINE, and VLINE statements, this option has no effect unless *variable* is used in the accompanying chart.

Note This option is used only to color the table values. If you want to set additional text attributes, used the TEXTGROUP= option instead.

DROPONMISSING
specifies that the entire axis table is dropped when all of the values are missing. Consider using this option if the SAS log indicates that the specified data column used for the axis table is missing all values.

INDENT=*dimension<unit>*
specifies a value to be used with the INDENTWEIGHT= option to determine the indention for each text value. The default units for *dimension* are inches. If you want to specify values in other units, then you must specify the desired units with the value. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Default 1/8 inch

Restriction Axis table indention is not supported when DOT, HBAR, VBAR, HLINE, or VLINE statements are used. As an alternative, you can use one of the following: HBARBASIC, VBARBASIC, HBARPARM, VBARPARM, SERIES, or SCATTER.

Interaction The INDENTWEIGHT= option must be specified for this option to have any effect.

INDENTWEIGHT=*numeric-variable*
specifies the indention weight (multiplier) for each observation.

Restriction Axis table indention is not supported when DOT, HBAR, VBAR, HLINE, or VLINE statements are used. As an alternative, you can use one of the following: HBARBASIC, VBARBASIC, HBARPARM, VBARPARM, SERIES, or SCATTER.

Interaction For each observation, the INDENT= option value is multiplied by the value of the column specified by this option to determine the indention for that observation's value.

Note If the axis-table values are numeric and a user-defined format is applied to the values, then indention might not work well, especially if the format results in long strings.

Tip When using the INDENTWEIGHT= option, specify VALUEHALIGN=LEFT to ensure correct indention. Otherwise, the indention might not be as expected.

LABEL
LABEL="*text-string*"
NOLABEL
specifies whether the table label is shown or hidden. If you specify LABEL=, then you can also specify a text string for the label.

Defaults LABEL

If you do not specify a text string, then the variable name is used for the label. Or, if CLASS= is also specified, then the unique values of the specified class variable are used for the table labels.

Tip Use the LABELATTRS= option to modify the label text attributes. Use the LABELPOS= option to move the label.

LABELATTRS=*style-element*<(*options*)> | (*options*)
specifies the color and font attributes of the axis table label. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element in the current style.

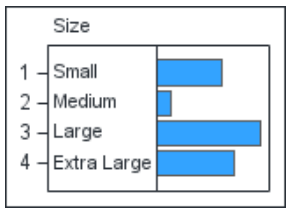
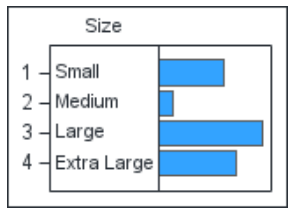
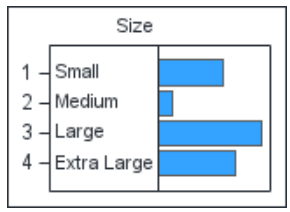
For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphValueText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Restriction Group behavior occurs only when the CLASS= and COLORGROUP= option values are the same.

Interactions This option has no effect if NOLABEL is also specified.

If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphValueText style element.

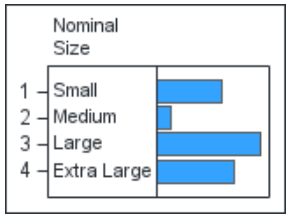
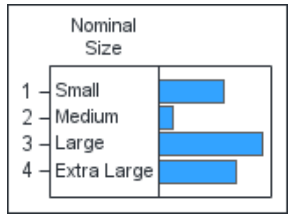
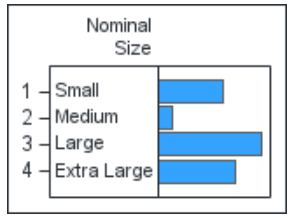
LABELHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the column labels, when displayed.

LEFT	CENTER	RIGHT
		

Default Uses the effective value of the LABELJUSTIFY= option.

Interaction This option has no effect if NOLABEL is also specified.

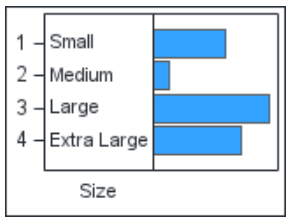
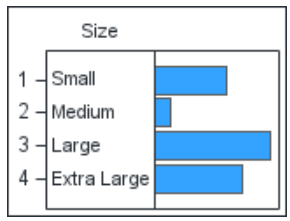
LABELJUSTIFY=LEFT | CENTER | RIGHT
specifies the justification of the labels.

LEFT	CENTER	RIGHT
		

Default Uses LEFT for text values or RIGHT for numeric values.

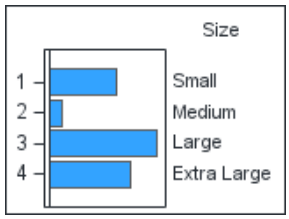
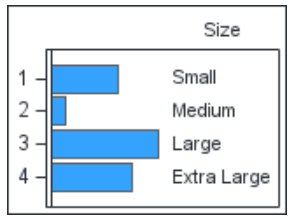
Interaction This option has no effect if NOLABEL is also specified.

LABELPOS=BOTTOM | TOP
specifies the position of the labels at the bottom or top of the axis table.

BOTTOM	TOP
	

Default TOP

LOCATION=OUTSIDE | INSIDE
specifies whether the axis table is placed outside or inside the axis area.

OUTSIDE	INSIDE
	

Default OUTSIDE

NAME="*text-string*"
assigns a name to a plot statement. You can use the name to refer to this plot in other statements.

Note The *text-string* is case-sensitive and must define a unique name within the procedure.

Tip This option is often used with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

NOMISSINGCHAR
suppresses the display of the MISSING character (.) for missing numeric values. Missing numeric values are displayed as blanks.

NOMISSINGCLASS
specifies that missing values of the class variable are not included in the table.

Interaction For this option to have any effect, the CLASS= option must be specified.

PAD=*numeric-value<units>* | *(pad-options)*
specifies the amount of extra space that is added inside the table border. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

numeric-value<units>
specifies a dimension to use for the extra space at the table border.

(pad-options)
specifies a space-separated list of pad options, enclosed in parentheses.

Specify one or more of the following name-value-pair options:

LEFT=*dimension*
specifies the amount of extra space added to the left side.

Default 4 px

RIGHT=*dimension*
specifies the amount of extra space added to the right side.

Default 4 px

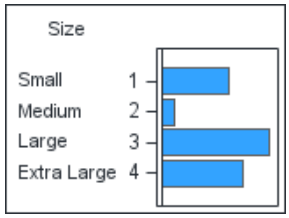
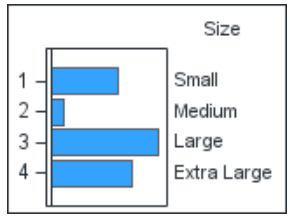
Notes Sides that are not assigned padding are padded with the default amount of space.

TOP and BOTTOM are not valid for a Y-axis table.

Tip Use *pad-options* to create non-uniform padding.

Note The default units for *dimension* are pixels. If you want to specify values in other units, then you must specify the desired units with the value. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

POSITION=LEFT | RIGHT
specifies the position of the axis table at the left or right side of the graph.

LEFT	RIGHT
	

Defaults RIGHT when the primary vertical axis is used for the plot.

LEFT when the secondary (Y2AXIS) vertical axis is specified for the plot.

SEPARATOR

creates a separating line between the axis table or axis tables and the plot.

Interaction This option has no effect unless LOCATION=INSIDE is also specified.

STAT=FREQ | MEAN | MEDIAN | PERCENT | SUM

specifies the statistic for the axis table. The statistic is applied only to numeric variables. For non-categorical plots, only the SUM and MEAN statistics are available.

Specify one of the following:

FREQ

the frequency of the axis table variable.

Interactions For this value to take effect, the graph must use a categorical plot (bar chart, line chart, or dot plot).

The FREQ value has a different effect when the axis table variable is the same as the categorical variable of the primary plot. In that case, the axis table variable is treated as a category (class) variable instead of a response (analysis) variable when computing the frequency. The FREQ value is the frequency of the categorical variable.

MEAN

the mean of the axis table variable.

MEDIAN

the median of the axis table variable.

Interactions For this value to take effect, the graph must use a category plot (bar chart, line chart, or dot plot).

This value is ignored when the axis table variable is the same as the categorical variable of the primary plot. In that case, the FREQ statistic is used and a message is written to the SAS log.

PERCENT
the percentage of the sum of the axis table variable.

When calculating the percentage of the sum, it is possible to have negative percentage values. However, the procedure calculates the absolute value of these percentages. Therefore, the percentages add up to 100% at the requested level.

Alias PCT

Interactions For this value to take effect, the graph must use a categorical plot (bar chart, line chart, or dot plot).

The PERCENT value has a different effect when the axis table variable is the same as the categorical variable of the primary plot. In that case, the axis table variable is treated as a category (class) variable instead of as a response (analysis) variable when computing the percentage. Therefore, the PERCENT value is calculated as a percentage of frequency instead of a percentage of the sum.

The PERCENT calculation can be performed at different levels in the graph. The level can be specified with the PCTLEVEL= option in the PROC SGPLOT statement.

You can use the PCTNDEC= option in the SGPLOT procedure statement to control the number of decimals to be used when calculating the percent values. The default value is 1.

Note If all of the frequencies or sums for a specified level are zero, all of the percentages for that level are zero.

SUM
the sum of the axis table variable.

Defaults SUM

When used with a categorical plot and the axis table variable is the same as the categorical variable, FREQ is the default.

Restrictions The STAT= option is applied only to numeric variables.

Only SUM and MEAN are supported for non-categorical plots.

When used with a categorical plot and the axis table variable is the same as the categorical variable, only FREQ and PERCENT are supported.

Interaction Any STAT= value specified in the primary plot has no effect on the axis table statistic.

STATLABEL | NOSTATLABEL
specifies whether the variable statistic is displayed in the table's label.

STATLABEL
forces the statistic to be displayed in the label.

NOSTATLABEL
removes the statistic from the label.

Defaults The statistic is displayed for the variable.

When a custom label is assigned to the variable, the statistic is not displayed.

Interaction This option has no effect unless the STAT= option is also specified in the axis table statement.

TEXTGROUP=*attribute-map-group-variable*
specifies the group variable that is used in a discrete attribute map data set to map text attributes to values for each observation. You specify this option only if you are using an attribute map to control visual attributes of the graph. The variable's values must correspond to the values in the VALUE variable in the attribute map data set. For more information, see Chapter 18, "Using Discrete Attribute Maps," on page 1901.

Interactions The TEXTGROUPID= option specifies the attribute ID to use for the attribute mapping. If TEXTGROUPID= is not specified, then the ATTRID= option is used. If the ATTRID= option is also not specified, then the TEXTGROUP= option is ignored.

When this option is specified, the COLORGROUP= option is ignored.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

TEXTGROUPID=*attribute-map-id*
specifies an attribute ID for the TEXTGROUP= option.

Default If neither TEXTGROUPID= nor ATTRID= is specified, the TEXTGROUP= option is ignored.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

TITLE="*text-string*"
specifies a title for the axis table. It the axis table statement specifies more than one variable, the title is displayed for each variable.

Tip Use the TITLEATTRS= option to modify the title text attributes.

TITLEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the title for the axis table. You can specify the appearance by using a style element or by specifying text options. If you specify a style element, you can also specify text options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeight attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

Interactions This option has no effect unless TITLE= is also specified.

If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphDataText style element.

Examples
```
TITLEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
TITLEATTRS=GraphTitleText
```

TITLEHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the axis table title bounding box relative to the axis table width. This option applies only to Y-axis tables.

Default LEFT

Interaction This option takes effect only when TITLE= is also specified.

Tip For a multiline title, use TITLEJUSTIFY= to justify the title text lines in the title bounding box.

TITLEJUSTIFY=LEFT | CENTER | RIGHT
specifies the justification of the lines of a multiline title in the title bounding box. The justification is relative to the axis table width. This option applies only to Y-axis tables.

Default LEFT

Restriction The title must wrap to multiple lines for this option to have any effect.

Interaction This option takes effect only when TITLE= is also specified.

Tip Use TITLEHALIGN= to align the title bounding box.

VALUEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the axis table values. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the text options, see "Text Attributes" on page 1835.

Defaults For ungrouped data, the Color, FontFamily, FontSize, FontStyle, and FontWeigh attributes of the GraphDataText style element in the current style.

For grouped data, the FontFamily, FontSize, FontStyle, and FontWeigh attributes of the GraphDataText style element, and the ContrastColor attribute of a GraphData1–GraphData*n* style element in the current style.

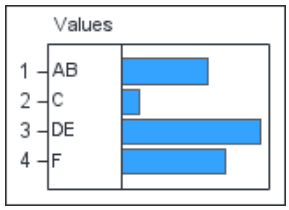
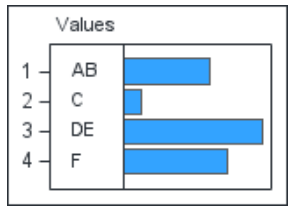
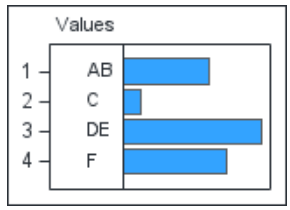
Interaction If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphDataText style element.

Examples
```
VALUEATTRS=(Color=Green Family=Arial Size=8
        Style=Italic Weight=Bold)
```

Here is an example that specifies a style element:
```
VALUEATTRS=GraphTitleText
```

VALUEHALIGN=LEFT | CENTER | RIGHT

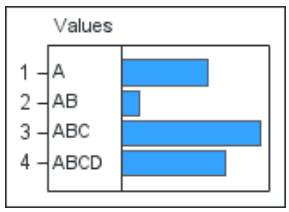
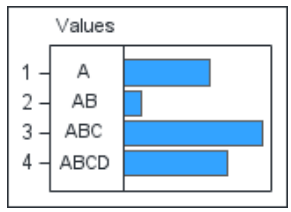
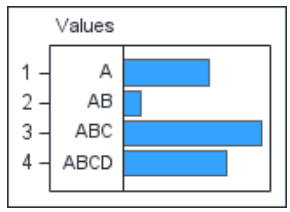
specifies the horizontal alignment of the axis table values relative to the column width in the table. This option aligns the text within the column based on the longest string.

LEFT	CENTER	RIGHT
		

Default Uses the effective value of the VALUEJUSTIFY= option.

VALUEJUSTIFY=LEFT | CENTER | RIGHT

specifies the justification of the axis table values relative to the column width in the table. This option horizontally aligns the table values center, left, or right relative to the column width.

LEFT	CENTER	RIGHT
		

Default LEFT for text values or RIGHT for numeric values.

Y=*variable*
Specifies the variable to use to align the table values to the Y or Y2 axis.

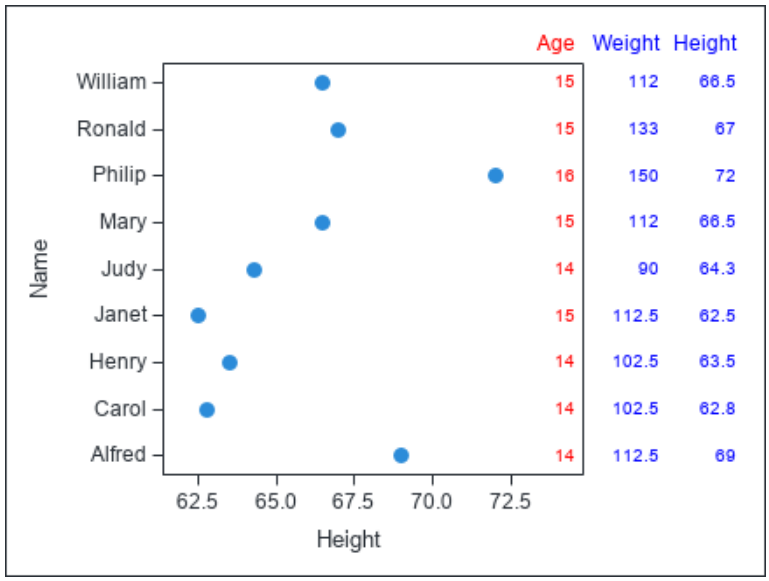
Default When this option is not specified, the Y variable of the primary plot is used by default.

Interaction This option is ignored when the axis table is used with a categorical chart (bar, line, or dot). The category variable from the chart is used instead.

Y2AXIS
assigns the table to the secondary vertical axis.

Example: YAXISTABLE Statements

This example shows a table of values along the Y axis of a scatter plot that plots student weight.



Here are the noteworthy features:

- The first axis table statement displays the age (in red) of each student. The LOCATION= on page 1642INSIDE option locates the axis table within the plot axes.
- The second axis table statement specifies two variables, height and weight, and is located to the right of the Y axis.

Here is the SAS code for this example.

```
ods graphics / width=4.5in;

proc sgplot data=sashelp.class (where=(age > 13));
scatter y=name x=height;
yaxistable age / location=inside position=right
```

```
        valueattrs=(color=red)
        labelattrs=(color=red);
yaxistable weight height / valueattrs=(color=blue)
        labelattrs=(color=blue);
run;
```

For an example that shows horizontal axis tables, see "Example: XAXISTABLE Statements" on page 1586.

ATTRIB Statement

Associates a format, informat, label, and length with one or more variables.

Type: Procedure statement

Syntax

ATTRIB *variable-list(s) attribute-list(s)*;

Arguments

variable-list(s)
names the variables that you want to associate with the attributes.

> **TIP** List the variables in any form that SAS allows.

attribute-list(s)
specifies one or more attributes to assign to *variable-list*. Specify one or more of these attributes in the ATTRIB statement:

FORMAT=*format*
associates a format with variables in *variable-list*.

Tip The format can be either a standard SAS format or a format that is defined with the FORMAT procedure.

INFORMAT=*informat*
associates an informat with variables in *variable-list*.

Tip The informat can be either a standard SAS informat or an informat that is defined with the FORMAT procedure.

LABEL='*label*'
associates a label with variables in *variable-list*.

See Also

"ATTRIB Statement" on page 142 for more information about the ATTRIB statement.

BY Statement

Orders the output according to the BY groups.

Type: Procedure statement

Syntax

BY <DESCENDING> *variable-1*
<-<DESCENDING> *variable-n*>
<NOTSORTED>;

Required Argument

variable
specifies the variable that the procedure uses to form BY groups. You can specify more than one variable. If you do not use the NOTSORTED option in the BY statement, then either the observations in the data set must be sorted by all the variables that you specify, or they must be indexed appropriately. Variables in a BY statement are called *BY variables*.

Optional Arguments

DESCENDING
specifies that the observations are sorted in descending order by the variable that immediately follows the word DESCENDING in the BY statement.

NOTSORTED
specifies that observations are not necessarily sorted in alphabetic or numeric order. The observations are grouped in another way (for example, chronological order).

The requirement for ordering or indexing observations according to the values of BY variables is suspended for BY-group processing when you use the NOTSORTED option. In fact, the procedure does not use an index if you specify NOTSORTED. The procedure defines a BY group as a set of contiguous observations that have the same values for all BY variables. If observations with the same values for the BY variables are not contiguous, then the procedure treats each contiguous set as a separate BY group.

You cannot use the NOTSORTED option in a PROC SORT step.

Note: You cannot use the GROUPFORMAT option, which is available in the BY statement in a DATA step, in a BY statement in any PROC step.

See Also

"BY Statement" on page 143 for more information about the BY statement.

FORMAT Statement

Associates formats with variables.

Type: Procedure statement

Syntax

FORMAT *variable-1* <-*variable-n*> <*format-1*> <*variable-1* <-*variable-n*> <*format-2*> > - <DEFAULT=*default-format*>;

Required Argument

variable
names one or more variables for SAS to associate with a format. You must specify at least one *variable*.

Tip To disassociate a format from a variable, use the variable in a FORMAT statement without specifying a format in a DATA step or in PROC DATASETS. In a DATA step, place this FORMAT statement after the SET statement. See "Removing a Format" in *SAS DATA Step Statements: Reference*. You can also use PROC DATASETS.

Optional Arguments

format
specifies the format that is listed for writing the values of the variables.

Tip Formats that are associated with variables by using a FORMAT statement behave like formats that are used with a colon modifier in a subsequent PUT statement. For more information about using a colon modifier, see "PUT Statement: List" in *SAS DATA Step Statements: Reference*.

See *SAS Formats and Informats: Reference*

DEFAULT=*default-format*
specifies a temporary default format for displaying the values of variables that are not listed in the FORMAT statement. These default formats apply only to the current DATA step; they are not permanently associated with variables in the output data set.

A DEFAULT= format specification applies to

- variables that are not named in a FORMAT or ATTRIB statement
- variables that are not permanently associated with a format within a SAS data set
- variables that are not written with the explicit use of a format.

Default If you omit DEFAULT=, SAS uses BEST*w*. as the default numeric format and $*w*. as the default character format.

Restriction Use this option only in a DATA step.

Tip A DEFAULT= specification can occur anywhere in a FORMAT statement. It can specify either a numeric default, a character default, or both.

Example "Assigning Formats and Defaults" in *SAS DATA Step Statements: Reference*

See Also

"FORMAT Statement" on page 146 for more information about the FORMAT statement.

LABEL Statement

Assigns descriptive labels to variables.

Type: Procedure statement

Syntax

Form 1: Creates a descriptive label for one or more variables.

LABEL *variable-1*='*text-string*' <...*variable-n*='*text-string*>;

Form 2: Removes the label from one or more variables.

LABEL *variable-1* = ' ' < ...*variable-n*=' '>; | *variable-1* = <...*variable-n*= >

Arguments

variable

specifies the variable that you want to label. You can specify labels for multiple variables in a single LABEL statement:

```
data test;
   L = 5;
   W = 10;
   label L = 'Length' W = 'Width';
run;
```

text-string

specifies a label of up to 256 bytes.

```
data test;
   L = 5;
   label L = 'Length';
run;
```

Restrictions

If the label includes a semicolon (;) or an equal sign (=), you must enclose the label in either single or double quotation marks.

```
label N = 'Number = ';
```

If the label includes single quotation marks (’), you must enclose the label in double quotation marks.

```
label Name = "Owner's Last Name";
```

There is partial support for variable name substitution with the JAVAIMG device. When the label text for the variable exceeds the space allowed, the variable name is used instead by procedures that are labeling an axis or a legend title. With the exception of the SAS/GRAPH GPLOT procedure, the variable name is not used when the JAVAIMG device is specified.

Notes

Quotation marks are not required unless the label includes a semicolon, quotation marks, or an equal sign.

The LABEL statement expects an equal sign (=) after the first variable. If any other character is found, an error occurs. The LABEL statement considers everything after the equal sign, following the first variable, as part of the label until the statement comes to another variable followed by an equal sign, regardless of quotation marks. In this example, the label for x is **`this is x y 4 = this is y`** because the next variable that is followed by an equal sign is z.

```
data test;
   x = 1;
   y = 1;
   z = 1;
   label x = 'this is x' y 4 = 'this is y' z = 'This is z';
run;
```

In this LABEL statement, the x variable has the same label:

```
label x = this is x y 4 = this is y  z = This is z;
```

See “Character Constants and Character Variables” in *SAS Programmer’s Guide: Essentials*.

’’

removes a label from a variable. Enclose a single blank space in quotation marks to remove an existing label. You can remove labels from multiple variables in a single LABEL statement:

```
data test;
   L=5; W=10;
   label L = ' '  W = ' ';
run;
```

Alternatively, you can remove a label by specifying nothing after the equal sign:

```
data test;
   L=5; W=10;
   label L =  W = ;
run;
```

See Also

“LABEL Statement” on page 150 for more information about the LABEL statement.

WHERE Statement

Subsets the input data set by specifying certain conditions that each observation must meet before it is available for processing.

Type: Procedure statement

Syntax

WHERE *where-expression*;

Required Argument

where-expression
is an arithmetic or logical expression that generally consists of a sequence of operands and operators. For more information about where processing, see *SAS DATA Step Statements: Reference*.

Tip You can specify multiple where-expressions.

See Also

"WHERE Statement" on page 151 for more information about the WHERE statement.

Examples: SGPLOT Procedure

About the SGPLOT Procedure Examples

Examples in This Section

This section provides basic examples of how to use the SGPLOT procedure. The following examples are provided:

Example	Description
"Example 1: Grouping a Scatter Plot" (p. 1660)	Shows how to use the SCATTER statement to create a simple scatter plot with grouped data.
"Example 2: Clustering a Grouped Scatter Plot" (p. 1661)	Shows how to use the SCATTER statement with the GROUPDISPLAY= and CLUSTERWIDTH= options to create a scatter plot with grouped data that is clustered.
"Example 3: Plotting Three Series" (p. 1662)	Shows how to use multiple SERIES statements to create an overlay series plot.
"Example 4: Adding Prediction and Confidence Bands to a Regression Plot" (p. 1663)	Shows how to use the REG statement to create a regression plot with prediction and confidence limits.
"Example 5: Adding a Prediction Ellipse to a Scatter Plot" (p. 1664)	Shows how to use the SCATTER and ELLIPSE statements to create a scatter plot with a prediction ellipse.
"Example 6: Creating Lines and Bands from Pre-Computed Data" (p. 1666)	Shows how to use the BAND, SCATTER, and SERIES statements to create a scatter plot, fit line, and confidence bands from pre-computed data.

Example	Description
"Example 7: Adding Statistical Limits to a Dot Plot" (p. 1668)	Shows how to use the DOT statement to create a dot plot with a response variable and statistical limits.
"Example 8: Combining Histograms with Density Plots" (p. 1669)	Shows how to use the HISTOGRAM and DENSITY statements to create a histogram combined with two density plots.
"Example 9: Creating a Horizontal Box Plot" (p. 1670)	Shows how to use the HBOX statement to create a horizontal box plot.
"Example 10: Creating a Bar-Line Chart" (p. 1671)	Shows how to use the VBAR and VLINE statements to create a bar-line chart.
"Example 11: Creating a High-Low Chart" (p. 1672)	Shows how to use the HIGHLOW statement to create a high-low plot of stock prices.
"Example 12: Series Plot with Line-Thickness Response and Arrowheads" (p. 1673)	Shows how to use the SERIES statement to create a grouped series plot with line-thickness response and arrowheads.

Additional Examples in This Book

Additional examples exist elsewhere in this book for most of the SGPLOT procedure statements. The following table lists these examples by statement.

Statement	Reference
BAND	"About Band Plots" (p. 17)
BLOCK	"About Block Plots" (p. 18)
BUBBLE	"Example 1: Creating a Basic Bubble Plot" on page 945 "Example 2: Enhancing a Bubble Plot" on page 945
DENSITY	"About Density Plots" (p. 49)
DOT	"About Dot Plots" (p. 60)
DROPLINE	"About Drop Lines" on page 26 "Example" on page 978
ELLIPSE	"About Ellipse Plots" (p. 42)

Statement	Reference
ELLIPSEPARM	“Example” (p. 996)
FRINGE	“About Fringe Plots” (p. 20)
GRADLEGEND	“Example” (p. 1007)
HBAR HBARBASIC HBARPARM	“About Bar Charts” (p. 53)
HBOX	“About Box Plots” (p. 47)
HEATMAP HEATMAPPARM	“About Heat Maps” (p. 21)
HIGHLOW	“About High-Low Plots” (p. 23)
HISTOGRAM	“About Histograms” (p. 51)
HLINE	“About Line Charts” (p. 61)
INSET	“About Text Insets” (p. 38)
KEYLEGEND	“Example 1: Legend for Two Ellipses” on page 1179 “Example 2: Legend with a Position Specified inside the Axis Area” on page 1180
LEGENDITEM	“Example” (p. 1186)
LINEPARM	“About Parameterized Lines” (p. 27)
LOESS	“LOESS Statement” (p. 451)
NEEDLE	“About Needle Plots” (p. 29)
PBSPLINE	“About Penalized B-Spline Plots” (p. 44)
POLYGON	“Example 1: Create a Single Filled Hexagon” on page 1258 “Example 2: Use a Polygon to Highlight Data in a Scatter Plot” on page 1258
REFLINE	“About Reference Lines” (p. 25)
REG	“About Regression Plots” (p. 45)

Statement	Reference
SCATTER	"About Scatter Plots" (p. 31)
SERIES	"About Series Plots" on page 33 "Example 1: Using Group Options without an Attribute Map" on page 1323 "Example 3: Using Curve Label Options with a Series Plot" on page 1327
SPLINE	"About Spline Plots" (p. 34)
STEP	"About Step Plots" (p. 35)
STYLEATTRS	"Example" (p. 906)
SYMBOLCHAR	"Example 1: Simple SYMBOLCHAR Example" on page 1365 "Example 2: SYMBOLCHAR Used with an SG Attribute Map" on page 1366
SYMBOLIMAGE	"Example 1: Simple SYMBOLIMAGE Example" on page 1370 "Example 2: SYMBOLIMAGE Used with an SG Attribute Map" on page 1371
TEXT	"About Text Plots" (p. 36)
VBAR VBARBASIC VBARPARM	"About Bar Charts" (p. 53)
VBOX	"About Box Plots" (p. 47)
VECTOR	"About Vector Plots" (p. 39)
VLINE	"About Line Charts" (p. 61)
WATERFALL	"About Waterfall Charts" (p. 64)
XAXISTABLE	"Example" (p. 1586)
YAXISTABLE	"Example" (p. 1649)

Examples Available Online

You can find more examples on the Graphically Speaking blog. On the blog page, use **Search Graphically Speaking** to search for SGPLOT to find the SGPLOT procedure examples.

You can also find examples on support.sas.com at the following URL: https://support.sas.com/sassamples/graphgallery/PROC_SGPLOT.html.

Example 1: Grouping a Scatter Plot

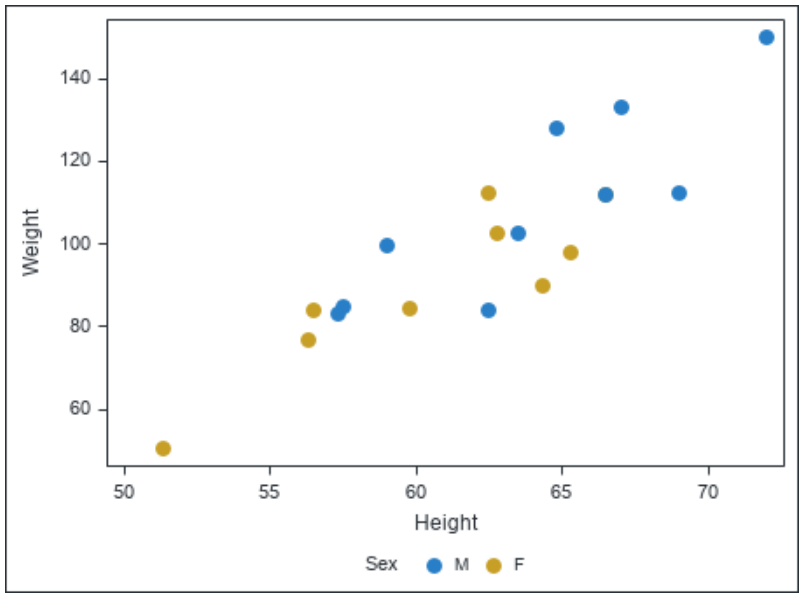
Features: SCATTER statement

Sample library member: SGPLSCT

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a simple scatter plot with grouped data.

Output



Program

Create the scatter plot. In the SCATTER statement, the GROUP= on page 1311 option groups the data by the SEX variable.

```
proc sgplot data=sashelp.class;
  scatter x=height y=weight / group=sex;
run;
```

Example 2: Clustering a Grouped Scatter Plot

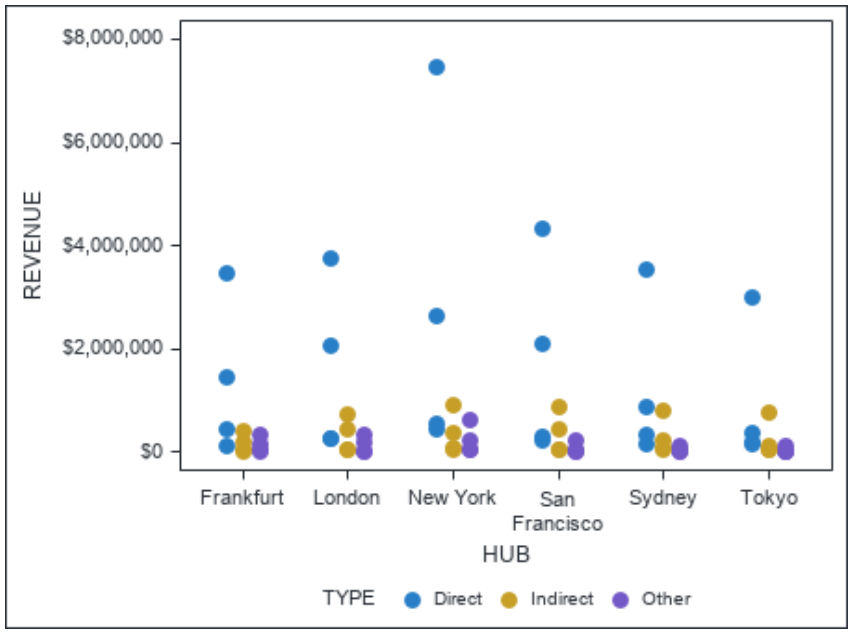
Features: SCATTER statement
GROUPDISPLAY option
CLUSTERWIDTH option

Sample library member: SGPLCLU

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a simple scatter plot with grouped data that is clustered.

Output



Program

Create the scatter plot. In the SCATTER statement, the GROUP= on page 1311 option groups the data by the variable specified in the XAXIS statement TYPE= on page 1570 option.The GROUPDISPLAY= on page 1291 option specifies that the grouped markers are clustered.The CLUSTERWIDTH= on page 1306 option specifies the width of the group clusters.

```
proc sgplot data=sashelp.revhub2;
  scatter x=hub y=revenue /
    group=type groupdisplay=cluster clusterwidth=0.5;
  xaxis type=discrete;
run;
```

Example 3: Plotting Three Series

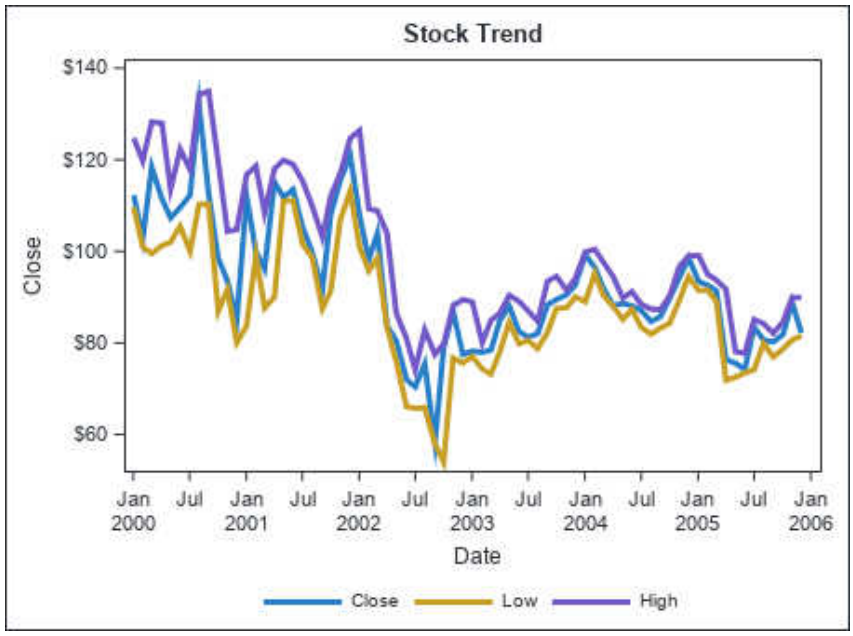
Features: SERIES statement

Sample library member: SGPLSER

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a series plot with three series on the Y axis.

Output



Program

```
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2000"d and stock = "IBM"));
  title "Stock Trend";

  series x=date y=close;
  series x=date y=low;
  series x=date y=high;
run;

title;
```

Program Description

Specify the data set and the title.

```
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2000"d and stock = "IBM"));
  title "Stock Trend";
```

Create the series plots.

```
  series x=date y=close;
  series x=date y=low;
  series x=date y=high;
run;
```

Cancel the title.

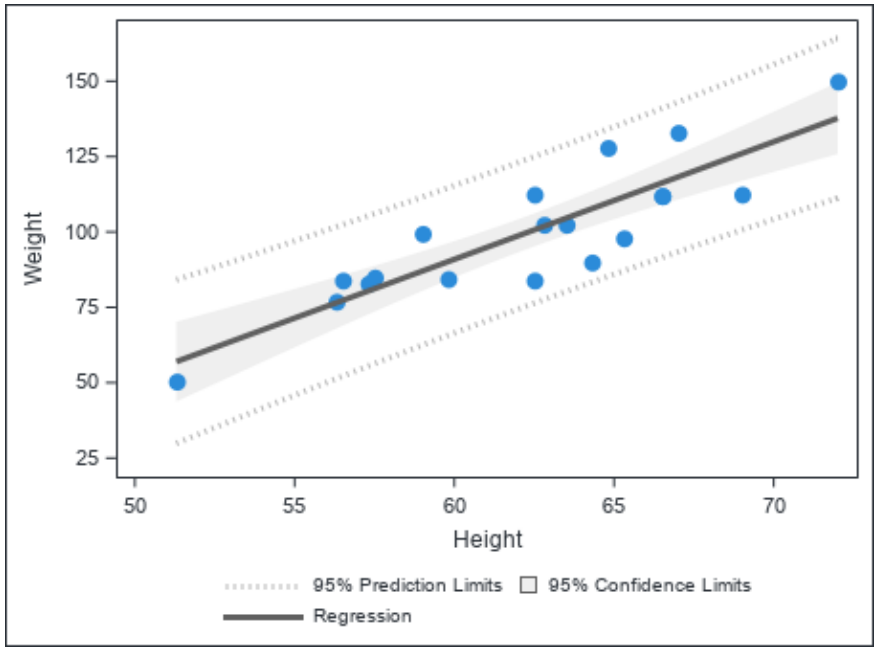
```
title;
```

Example 4: Adding Prediction and Confidence Bands to a Regression Plot

Features:	REG statement
Sample library member:	SGPLREG
Note:	For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a regression plot with prediction and confidence limits.

Output



Program

Create the regression plot. The CLM on page 1272 option adds confidence limits for the mean predicted values.The CLI on page 1270 option adds confidence limits for the individual predicted values.

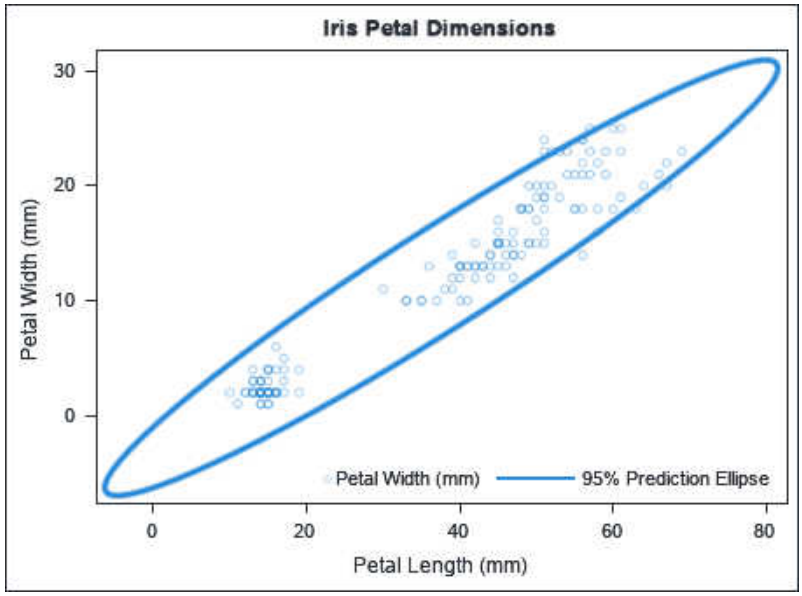
```
proc sgplot data=sashelp.class;
  reg x=height y=weight / CLM CLI;
run;
```

Example 5: Adding a Prediction Ellipse to a Scatter Plot

Features:	SCATTER statement ELLIPSE statement KEYLEGEND statement
Sample library member:	SGPLELI
Note:	For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a scatter plot with a prediction ellipse.

Output



Program

```
proc sgplot data=sashelp.iris;
  title "Iris Petal Dimensions";
  scatter x=petallength y=petalwidth /
     markerattrs=(symbol=circle size=4pt);

  ellipse x=petallength y=petalwidth;

  keylegend / location=inside position=bottomright;
run;

title;
```

Program Description

Set the title and create the scatter plot.

```
proc sgplot data=sashelp.iris;
  title "Iris Petal Dimensions";
  scatter x=petallength y=petalwidth /
     markerattrs=(symbol=circle size=4pt);
```

Create the ellipse.

```
  ellipse x=petallength y=petalwidth;
```

Position the Legend. The LOCATION= on page 1174 option places the legend inside the plot area.The POSITION= on page 1175 option places the legend at the bottom right.

```
  keylegend / location=inside position=bottomright;
```

```
run;
```

Cancel the title.

```
title;
```

Example 6: Creating Lines and Bands from Pre-Computed Data

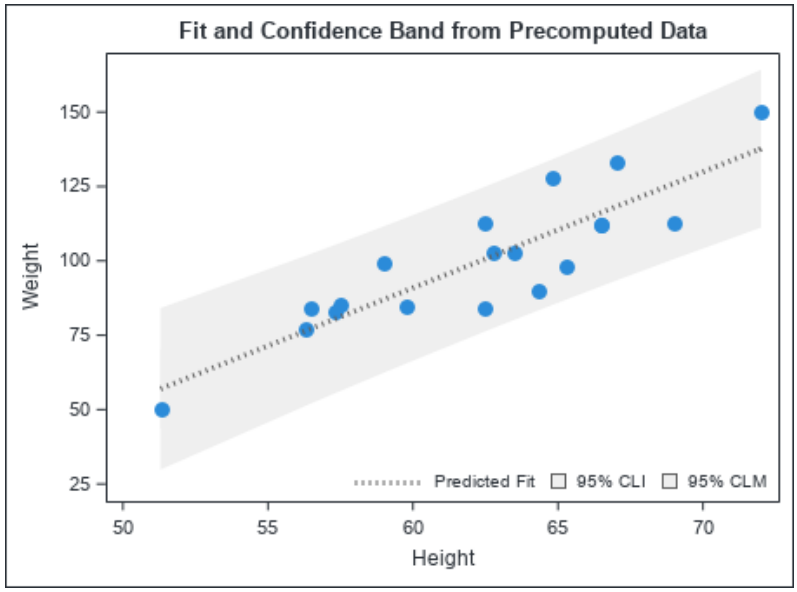
Features: BAND statement
KEYLEGEND statement
SCATTER statement
SERIES statement

Sample library member: SGPLBND

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows how to use pre-computed data to create a scatter plot, fit line, and confidence bands. The data set was created by the REG procedure. This technique is useful for more complex fit models.

Output



Program

```
proc sgplot data=sashelp.classfit;
  title "Fit and Confidence Band from Precomputed Data";
  band x=height lower=lower upper=upper /
       legendlabel="95% CLI" name="band1";


  band x=height lower=lowermean upper=uppermean /
       fillattrs=GraphConfidence2
       legendlabel="95% CLM" name="band2";


  scatter x=height y=weight;
  series x=height y=predict / lineattrs=GraphPrediction
         legendlabel="Predicted Fit" name="series";

  keylegend "series" "band1" "band2" / location=inside
position=bottomright;
run;

title;
```

Program Description

Set the title and create the first band plot. The LEGENDLABEL= on page 913 option in the BAND statement specifies the label for the band plot in the legend.

```
proc sgplot data=sashelp.classfit;
  title "Fit and Confidence Band from Precomputed Data";
  band x=height lower=lower upper=upper /
       legendlabel="95% CLI" name="band1";
```

Create the second band plot. The LEGENDLABEL= option specifies the label for the band plot in the legend. The FILLATTRS= on page 937 option specifies the style element for the fill.

```
  band x=height lower=lowermean upper=uppermean /
       fillattrs=GraphConfidence2
       legendlabel="95% CLM" name="band2";
```

Create the scatter and series plots. The LINEATTRS= on page 1315 option in the SERIES statement specifies the style attribute for the series plot.The LEGENDLABEL= on page 1315 option in the SERIES statement specifies the legend label for the series plot.

```
  scatter x=height y=weight;
  series x=height y=predict / lineattrs=GraphPrediction
         legendlabel="Predicted Fit" name="series";
```

Create a legend for the graph. The quoted strings specify the names of the plots that you want to include in the legend. The LOCATION= on page 1174 option places the legend inside the plot area.The POSITION= on page 1175 option places the legend in the bottom right corner of the graph.

```
  keylegend "series" "band1" "band2" / location=inside
position=bottomright;
run;
```

Cancel the title.

```
title;
```

Example 7: Adding Statistical Limits to a Dot Plot

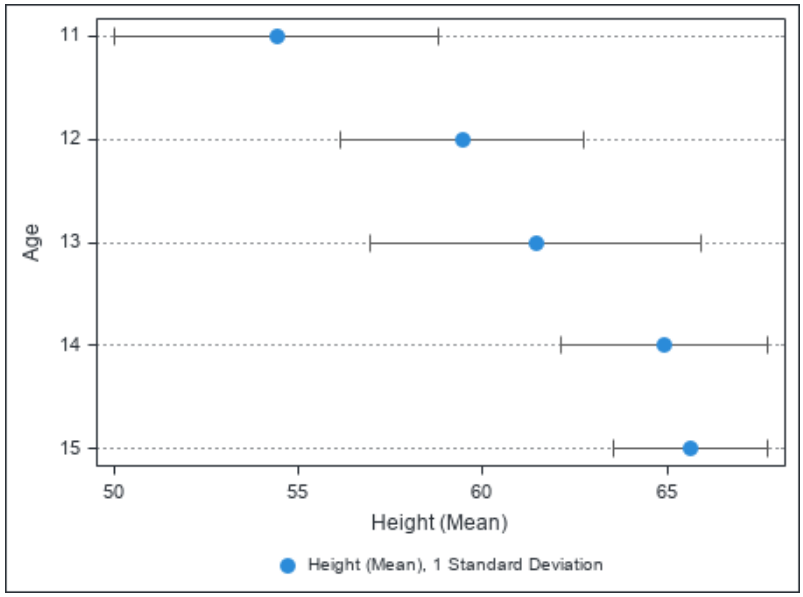
Features: DOT statement

Sample library member: SGPLDOT

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a dot plot with a response variable and statistical limits. Each dot represents the mean for each value of the category variable, and bands represent the standard deviation.

Output



Program

Create the dot plot. The RESPONSE= on page 968 option specifies the response variable.The STAT= on page 1030 option specifies that the mean statistic is used to analyze the graph.The LIMITSTAT= on page 965 option specifies that the limit

statistic is the standard deviation.The NUMSTD= on page 968 option specifies that one standard deviation is used.

```
proc sgplot data=sashelp.class(where=(age<16));
  dot age / response=height stat=mean
            limitstat=stddev numstd=1;
run;
```

Example 8: Combining Histograms with Density Plots

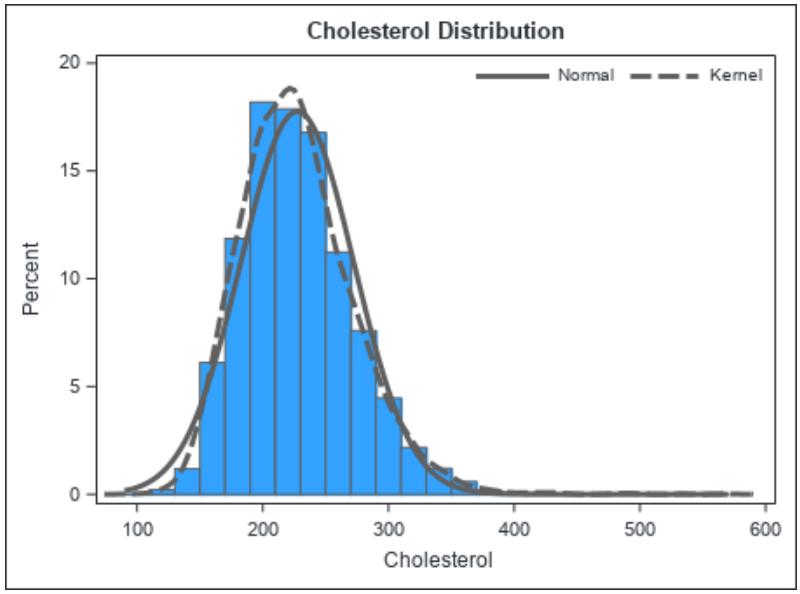
Features: HISTOGRAM statement
DENSITY statement
KEYLEGEND statement

Sample library member: SGPLHST

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a histogram combined with two density plots. One density plot uses a normal density estimate and the other density plot uses a kernel density estimate.

Output



Program

```
proc sgplot data=sashelp.heart;
  title "Cholesterol Distribution";
  histogram cholesterol;

  density cholesterol;
  density cholesterol / type=kernel;

  keylegend / location=inside position=topright;
run;

title;
```

Program Description

Set the title, set a label for the X axis, and create the histogram.

```
proc sgplot data=sashelp.heart;
  title "Cholesterol Distribution";
  histogram cholesterol;
```

Create the density plots. The TYPE= on page 952 option specifies which density equation is used.

```
  density cholesterol;
  density cholesterol / type=kernel;
```

Position the Legend The LOCATION= on page 1174 option places the legend inside the plot area.The POSITION= on page 1175 option places the legend at the top right.

```
  keylegend / location=inside position=topright;
run;
```

Cancel the title.

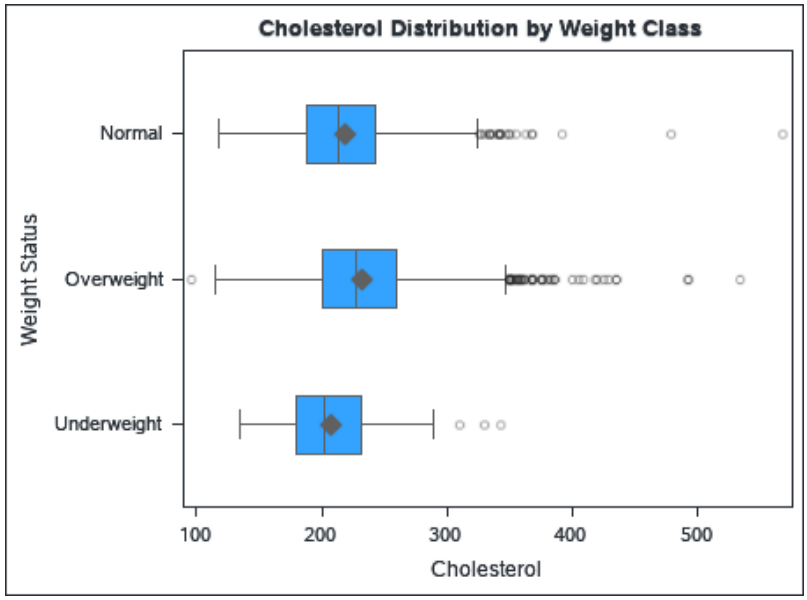
```
title;
```

Example 9: Creating a Horizontal Box Plot

Features:	HBOX statement
Sample library member:	SGPLBOX
Note:	For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a horizontal box plot.

Output



Program

Create the box plot. The CATEGORY= on page 1081 option specifies the category variable.

```
proc sgplot data=sashelp.heart;
  title "Cholesterol Distribution by Weight Class";
  hbox cholesterol / category=weight_status
     outlierattrs=(symbol=circle size=4pt);
run;
title;
```

Example 10: Creating a Bar-Line Chart

Features: VBAR statement
VLINE statement

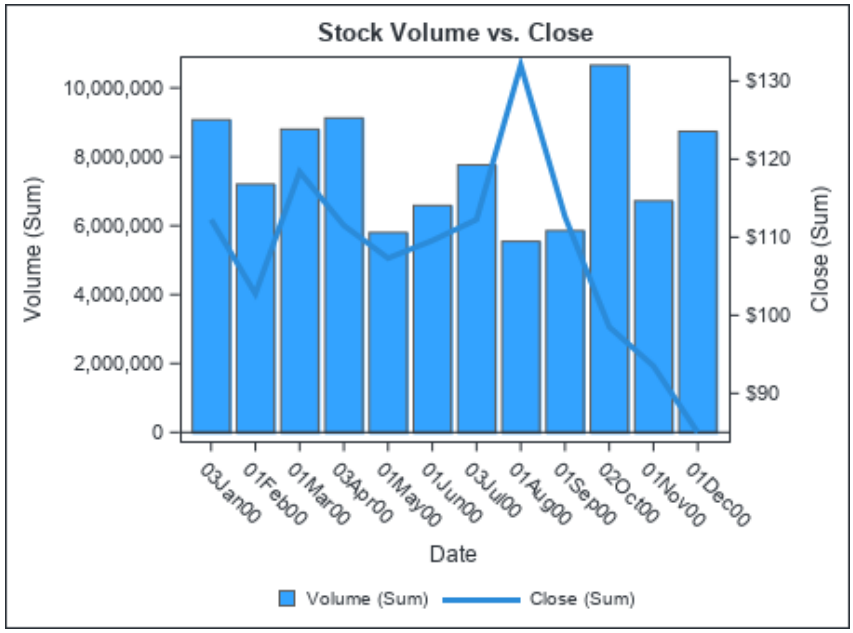
Sample library member: SGPLBAR

Notes: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

You can also combine the HBAR on page 1008 and HLINE on page 1145 statements to create a horizontal bar-line chart.

This example creates a bar-line chart.

Output



Program

Create the Bar-line Chart. The Y2AXIS on page 1514 option in the VLINE statement assigns the line plot to the Y2 axis.

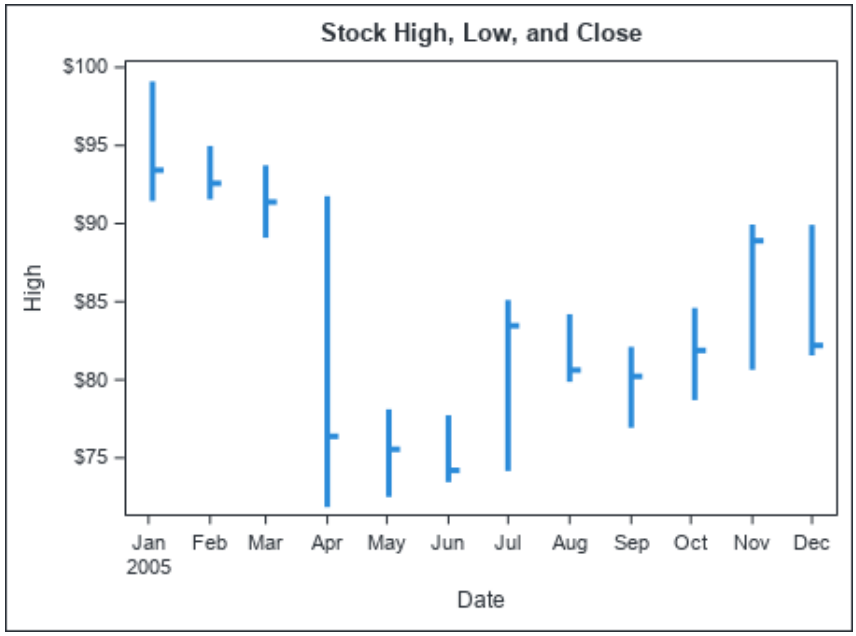
```
proc sgplot data=sashelp.stocks (where=(date >= "01jan2000"d
                                 and date <= "01jan2001"d
                                 and stock = "IBM"));
   title "Stock Volume vs. Close";
   vbar date / response=volume;
   vline date / response=close y2axis;
run;
title;
```

Example 11: Creating a High-Low Chart

Features:	HIGHLOW statement
Sample library member:	SGPLHILO
Note:	For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows the high, low, and closing stock prices for a company during the year 2005.

Output



Program

Create the high-low chart. The HIGH, LOW, and CLOSE variables are used in the HIGHLOW statement. In addition, the plot subsets the data by year and by company.

```
title "Stock High, Low, and Close";
proc sgplot data=sashelp.stocks;
  where Date >= '01JAN2005'd and stock='IBM';
  highlow x=date high=high low=low
    / close=close;
run;
title;
```

Example 12: Series Plot with Line-Thickness Response and Arrowheads

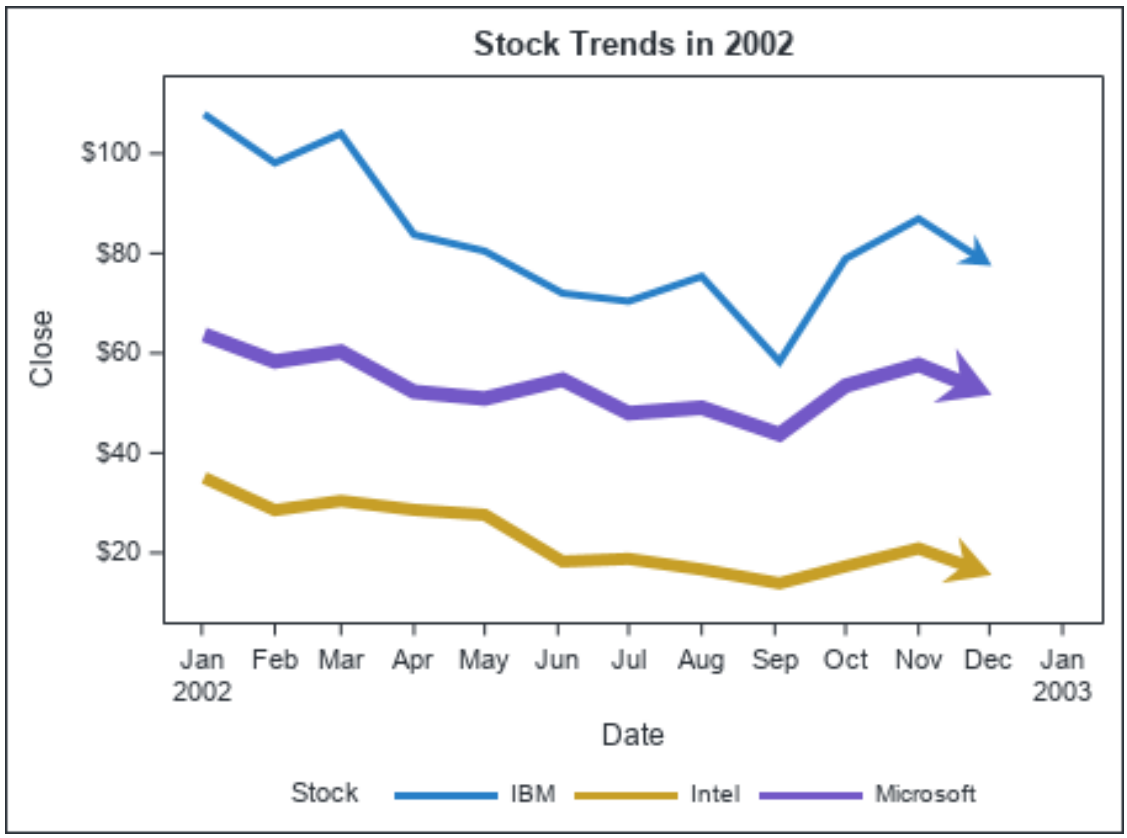
Features: SERIES statement

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

You can specify a response variable to control the thickness of the lines in your grouped series plot. You can also add arrowheads to one or both ends of your grouped or ungrouped series lines.

This example creates a grouped series plot that shows the monthly closing price for IBM, Intel, and Microsoft stock in 2002. The mean volume is computed for each stock and is used to control the thickness of the series line for each stock. The maximum line width is set to 7px. A barbed arrowhead, scaled to 1.25, is positioned at the end of each series line. In order to position the arrowheads properly, the data must be sorted in ascending order by date.

Output



Program

```
data stocks;
  set sashelp.stocks(where=
    (date between "1jan02"d and "31dec02"d));
  volume = volume / 1000000;
  format date MONNAME3. volume 6.2;
run;

proc means data=stocks noprint;
  by stock notsorted;
  var volume;
  output out=meanvolume(keep=stock meanvolume) mean=meanvolume;
run;

data stocks;
```

```
  merge stocks meanvolume;
  by stock;
run;

proc sort data=stocks;
  by date;
run;

title "Stock Trends in 2002";
proc sgplot data=stocks;
 series x=date y=close / group=stock
   thickresp=meanvolume thickmax=7px
   arrowheadpos=end arrowheadshape=barbed
   arrowheadscale=1.25;
run;
title;
```

Program Description

Extract the 2002 data from SASHELP.STOCKS and convert volume to millions.

```
data stocks;
  set sashelp.stocks(where=
    (date between "1jan02"d and "31dec02"d));
  volume = volume / 1000000;
  format date MONNAME3. volume 6.2;
run;
```

Compute the average volume for each stock.

```
proc means data=stocks noprint;
  by stock notsorted;
  var volume;
  output out=meanvolume(keep=stock meanvolume) mean=meanvolume;
run;
```

Merge the average volume data with the stock data.

```
data stocks;
  merge stocks meanvolume;
  by stock;
run;
```

Sort the data by date.

```
proc sort data=stocks;
  by date;
run;
```

Create the series plot.

```
title "Stock Trends in 2002";
proc sgplot data=stocks;
 series x=date y=close / group=stock
   thickresp=meanvolume thickmax=7px
   arrowheadpos=end arrowheadshape=barbed
   arrowheadscale=1.25;
run;
title;
```

11

SGRENDER Procedure

Overview: SGRENDER Procedure ***1677***
What Does the SGRENDER Procedure Do? 1677

Syntax: SGRENDER Procedure ***1678***
PROC SGRENDER Statement 1678
DATTRVAR Statement 1680
DYNAMIC Statement 1683
ATTRIB Statement 1684
BY Statement 1685
FORMAT Statement 1686
LABEL Statement 1688
WHERE Statement 1690

Examples: SGRENDER Procedure ***1691***
About the SGRENDER Procedure Examples 1691
Example 1: Generating a Graph from a Simple GTL Template 1692
Example 2: Defining Dynamic Variables 1693

Overview: SGRENDER Procedure

What Does the SGRENDER Procedure Do?

The SGRENDER procedure produces graphical output from templates that are created with the Graph Template Language (GTL). The templates are referred to as StatGraph templates. The GTL is a comprehensive language for creating statistical graphics, which can be used to create customized layouts and graphs that are beyond the scope of the ODS Graphics procedures. For more information about the GTL, see *SAS Graph Template Language: User's Guide* and *SAS Graph Template Language: Reference*.

Syntax: SGRENDER Procedure

PROC SGRENDER <option(s)>;
DATTRVAR *variable-assignment(s)*;
DYNAMIC *variable-assignment(s)*;
ATTRIB *variable-list(s) attribute-list(s)*;
BY *variable-1*<...*variable-N*><*options*>;
FORMAT *variable-1*<...*variable-N*<*format-1*>><*options*>;
LABEL *variable-1*=*label-1*<...*variable-n*=*label-n*>;
WHERE *where-expression*;

PROC SGRENDER Statement

Identifies the data set that contains the plot variables and the StatGraph template. The statement also gives you the option to specify the name of the output object and the label for the output object.

Requirement: Requires an input data set and a template.

Syntax

PROC SGRENDER TEMPLATE=*statgraph-template* <option(s)>

Summary of Optional Arguments

Render options

DATA=*input-data-set*
: specifies the SAS data set that contains the variables to process.

DATTRMAP=*discrete-attr-map-data-set*
: specifies a map data set that contains discrete attribute map descriptions.

OBJECT=*object-name*
: specifies a name for the ODS output object.

OBJECTLABEL="*text-string*"
: specifies a description for the output image.

SGANNO=*annotation-data-set*

specifies the name of a SAS data set that contains annotation instructions.

Required Argument

TEMPLATE=*statgraph-template*
specifies a StatGraph template that defines one or more graphs. The SGRENDER procedure applies the StatGraph template to your data to create the output graphs.

Interaction This argument is required except when you use the SGE= option.

Note Specifying a template that is not a StatGraph template is not supported. If you specify a non-StatGraph template, then the SGRENDER procedure might produce unpredictable results.

Optional Arguments

DATA=*input-data-set*
specifies the SAS data set that contains the variables to process. By default, the procedure uses the most recently created SAS data set.

DATTRMAP=*discrete-attr-map-data-set*
specifies a map data set that contains discrete attribute map descriptions. The attribute map descriptions are read from the data set and are rendered in the graph output. Use this option when the discrete attribute map is defined in a SAS data set rather than in a DISCRETEATTRMAP block in the StatGraph template to be rendered.

Restriction Discrete attribute maps that are stored in tables are not supported.

Requirements Do either one of the following:

- specify the attribute map ID variable by using the DATTRVAR statement in the SGRENDER procedure.
- specify the attribute map ID variable using the DISCRETEATTRVAR variable in the StatGraph template.

If neither of these requirements is met, the descriptions in the attribute map data set are ignored.

The values in the DATTRMAP data set must be sorted by ID. If they are not, only the first value is found.

Interactions If both the DATTRVAR and DISCRETEATTRVAR statements are used, then the DISCRETEATTRVAR statement takes precedence.

If the ID value in the data set has the same identifier as an existing DISCRETEATTRMAP statement's NAME= option in the template, the definition in the template is used.

See "Defining a Discrete Attribute Map" in *SAS Graph Template Language: Reference*

"Key Concepts for Using Attribute Maps" in *SAS Graph Template Language: Reference*

OBJECT=*object-name*
specifies a name for the ODS output object.

Alias NAME=

Default SGRender

Note To specify the filename of the output image, use the IMAGENAME= option in the ODS GRAPHICS statement.

OBJECTLABEL="*text-string*"
specifies a description for the output image. The description identifies the image in the following locations:

- the alternate text for the image in HTML output
- the table of contents that is created by the CONTENTS option in an ODS statement

The default description is "The SGRENDER Procedure".

Aliases DES=

DESCRIPTION=

SGANNO=*annotation-data-set*
specifies the name of a SAS data set that contains annotation instructions. You specify this option only if you are rendering a template that contains one or more ANNOTATE statements. During rendering, when the ANNOTATE statement is encountered in the template, the annotation instructions are read from the annotation data set and are rendered in the current context in the graph. If the template does not include the ANNOTATE statement, the annotations in the annotation data set are ignored.

Restriction Annotation instructions that are stored in tables are not supported.

See "ANNOTATE Statement" in *SAS Graph Template Language: Reference*

"About the GTL Annotation Facility" in *SAS Graph Template Language: Reference*

DATTRVAR Statement

Specifies one or more input variables to be associated with an attribute map at run time.

Interactions: This statement is used in conjunction with the DATTRMAP= option in the PROC SGRENDER statement to associate a discrete attribute map with the StatGraph template.

If the template that is specified in the SGRENDER statement's TEMPLATE= option uses a DISCRETEATTRVAR statement, the DISCRETEATTRVAR statement takes precedence over the DATTRVAR statement.

Note: You can specify multiple instances of this statement in the SGRENDER procedure. The *variable-assignment(s)* pairs from all of the DATTRVAR statements are accumulated.

Syntax

DATTRVAR *variable-assignment(s)*;

Required Argument

variable-assignment(s)
specifies the values for one or more variables. When multiple variable assignments are specified, separate each with a space.

Each variable assignment takes the following form:

group-variable-name="attrid"

group-variable-name
a group variable to be associated with an attribute map at run time.

The variable must exist in the data set that you specify with the DATA= option in the PROC SGRENDER statement.

The variable can be character or numeric, but the values must match the type of values that are specified in the attribute map. For numeric variables, all values are treated as discrete values.

"attrid"
an identifier for the attribute map.

The *attrid* value must exist in the attribute map data set. This value must be a quoted string.

Note If a *group-variable-name* is not found or is of the wrong type for the attribute map, the *group-variable-name="attrid"* pair is ignored.

Details

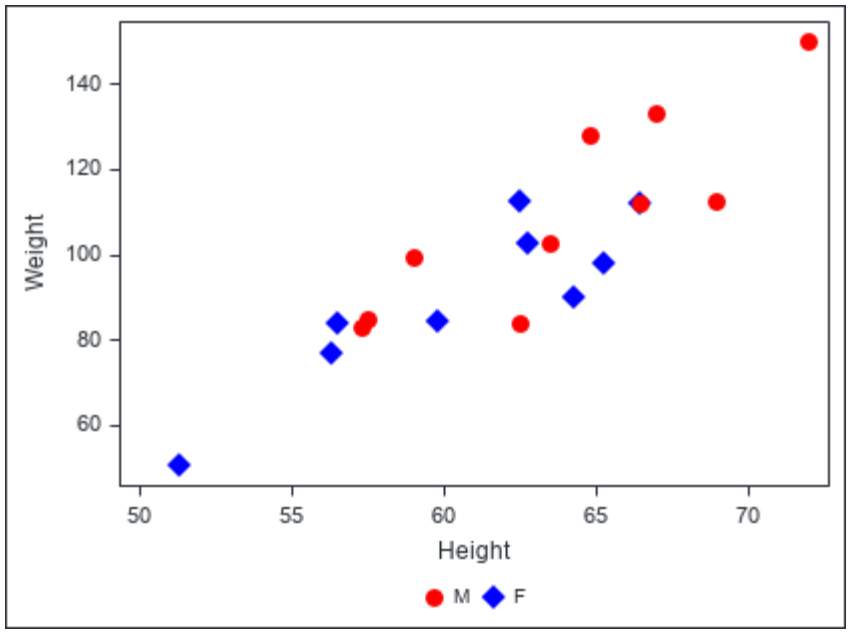
The StatGraph template does not need to reference the discrete attribute map. Instead, the SGRENDER procedure statement associates the template with the attribute map.

The SGRENDER procedure does the following:

- specifies the name of the attribute map data set using the DATTRMAP= option
- in the DATTRVAR statement, associates the group variable with the attribute map ID

Example

This example shows how to associate a StatGraph template with a discrete attribute map that is specified in a SAS data set.



In this example, the SGRENDER procedure does the following:

- specifies the name of the attribute map data set using DATTRMAP= on page 1679MYATTRMAP
- associates the group variable (SEX) with the attribute map ID (MYID) in the DATTRVAR on page 1680 statement

```
/* Create the attribute map data set */
data myAttrMap;
  length markersymbol $13;
  retain ID "myid";
  input value $ markersymbol $ markercolor $;
datalines;
F DiamondFilled blue
M CircleFilled red
;
run;

 /* Create a StatGraph template, and
     specify a group variable for the plot */
  proc template;
  define statgraph noattrmap;
    begingraph;
      layout overlay;
```

```
        scatterplot x=height y=weight / name="scatter"
         group=sex;
        discretelegend "scatter";
      endlayout;
    endgraph;
  end;

/* Render the graph */
proc sgrender data=sashelp.class template=noattrmap dattrmap=myAttrMap;
dattrvar sex="myid";
run;
```

DYNAMIC Statement

Defines the values for dynamic variables in your StatGraph template code.

See: "Example 2: Defining Dynamic Variables" on page 1693

Syntax

DYNAMIC *variable-assignment(s)*;

Required Argument

variable-assignment(s)
specifies the values for one or more dynamic variables. The variables that you assign must also be declared in a DYNAMIC statement within the StatGraph template.

You can specify either a numeric value or a quoted text string. For example, the following statement defines one string variable and one numeric variable:

```
dynamic region="Europe" limit=3000;
```

Details

Description

The DYNAMIC statement defines values for dynamic variables that are used in your StatGraph template.

You can either specify variables on multiple DYNAMIC statements, or specify all of your variables on a single DYNAMIC statement.

Note: If your template uses a dynamic variable to specify a required attribute, such as a variable name, and the name is misspelled or not provided in the SGRENDER procedure, then a warning is issued and the respective plot statement drops out of

the final graph. A graph is produced, but it might be a blank graph, or it might show the results of all statements except those that are in error.

Special Dynamic Variables

Some special dynamic variables can be declared and referenced in a StatGraph template and do not need to be assigned in the DYNAMIC statement of the SGRENDER procedure. The following table describes those dynamic variables.

Table 11.1 *Special Dynamic Variables*

LIBNAME	Represents the name of the library that contains the data set.
MEMNAME	Represents the name of the library member that contains the data set.
BYLINE	Represents the complete BY line, when you specify a BY statement.
BYFOOTNOTE	When the BYLINE=FOOTNOTE option is specified in the ODS Graphics statement, the BY line information is set in this dynamic. Typically, you use this dynamic with the ENTRYFOOTNOTE statement in your template.
BYTITLE	When the BYLINE=TITLE option is specified in the ODS Graphics statement, the BY line information is set in this dynamic. Typically, you use this dynamic with the ENTRYTITLE statement in your template.
BYVAR	Represents the name of the first BY variable, when you specify a BY statement.
BYVAR*n*	Represents the name of the *n*th BY variable, when you specify a BY statement with multiple variables.
BYVAL	Represents the first BY value, when you specify a BY statement.
BYVAL*n*	Represents the value of the *n*th BY variable, when you specify a BY statement with multiple variables.

ATTRIB Statement

Associates a format, informat, label, and length with one or more variables.

Type: Procedure statement

Syntax

ATTRIB *variable-list(s) attribute-list(s)*;

Arguments

variable-list(s)
names the variables that you want to associate with the attributes.

> **TIP** List the variables in any form that SAS allows.

attribute-list(s)
specifies one or more attributes to assign to *variable-list*. Specify one or more of these attributes in the ATTRIB statement:

FORMAT=*format*
associates a format with variables in *variable-list*.

Tip The format can be either a standard SAS format or a format that is defined with the FORMAT procedure.

INFORMAT=*informat*
associates an informat with variables in *variable-list*.

Tip The informat can be either a standard SAS informat or an informat that is defined with the FORMAT procedure.

LABEL='*label*'
associates a label with variables in *variable-list*.

See Also

"ATTRIB Statement" on page 142 for more information about the ATTRIB statement.

BY Statement

Orders the output according to the BY groups.

Type: Procedure statement

Syntax

BY <DESCENDING> *variable-1*

<-<DESCENDING> *variable-n*>
<NOTSORTED>;

Required Argument

variable
specifies the variable that the procedure uses to form BY groups. You can specify more than one variable. If you do not use the NOTSORTED option in the BY statement, then either the observations in the data set must be sorted by all the variables that you specify, or they must be indexed appropriately. Variables in a BY statement are called *BY variables*.

Optional Arguments

DESCENDING
specifies that the observations are sorted in descending order by the variable that immediately follows the word DESCENDING in the BY statement.

NOTSORTED
specifies that observations are not necessarily sorted in alphabetic or numeric order. The observations are grouped in another way (for example, chronological order).

The requirement for ordering or indexing observations according to the values of BY variables is suspended for BY-group processing when you use the NOTSORTED option. In fact, the procedure does not use an index if you specify NOTSORTED. The procedure defines a BY group as a set of contiguous observations that have the same values for all BY variables. If observations with the same values for the BY variables are not contiguous, then the procedure treats each contiguous set as a separate BY group.

You cannot use the NOTSORTED option in a PROC SORT step.

Note: You cannot use the GROUPFORMAT option, which is available in the BY statement in a DATA step, in a BY statement in any PROC step.

See Also

"BY Statement" on page 143 for more information about the BY statement.

FORMAT Statement

Associates formats with variables.

Type: Procedure statement

Syntax

FORMAT *variable-1* <-*variable-n*> <*format-1*> <*variable-1* <-*variable-n*> <*format-2*> > - <DEFAULT=*default-format*>;

Required Argument

variable
names one or more variables for SAS to associate with a format. You must specify at least one *variable*.

Tip To disassociate a format from a variable, use the variable in a FORMAT statement without specifying a format in a DATA step or in PROC DATASETS. In a DATA step, place this FORMAT statement after the SET statement. See "Removing a Format" in *SAS DATA Step Statements: Reference*. You can also use PROC DATASETS.

Optional Arguments

format
specifies the format that is listed for writing the values of the variables.

Tip Formats that are associated with variables by using a FORMAT statement behave like formats that are used with a colon modifier in a subsequent PUT statement. For more information about using a colon modifier, see "PUT Statement: List" in *SAS DATA Step Statements: Reference*.

See *SAS Formats and Informats: Reference*

DEFAULT=*default-format*
specifies a temporary default format for displaying the values of variables that are not listed in the FORMAT statement. These default formats apply only to the current DATA step; they are not permanently associated with variables in the output data set.

A DEFAULT= format specification applies to

- variables that are not named in a FORMAT or ATTRIB statement
- variables that are not permanently associated with a format within a SAS data set
- variables that are not written with the explicit use of a format.

Default If you omit DEFAULT=, SAS uses BEST*w*. as the default numeric format and $*w*. as the default character format.

Restriction Use this option only in a DATA step.

Tip A DEFAULT= specification can occur anywhere in a FORMAT statement. It can specify either a numeric default, a character default, or both.

Example "Assigning Formats and Defaults" in *SAS DATA Step Statements: Reference*

See Also

"FORMAT Statement" on page 146 for more information about the FORMAT statement.

LABEL Statement

Assigns descriptive labels to variables.

Type: Procedure statement

Syntax

Form 1: Creates a descriptive label for one or more variables.

LABEL *variable-1*='*text-string*' <...*variable-n*='*text-string*>;

Form 2: Removes the label from one or more variables.

LABEL *variable-1* = ' ' < ...*variable-n*=' '>; | *variable-1* = <...*variable-n*= >

Arguments

variable

specifies the variable that you want to label. You can specify labels for multiple variables in a single LABEL statement:

```
data test;
   L = 5;
   W = 10;
   label L = 'Length' W = 'Width';
run;
```

text-string

specifies a label of up to 256 bytes.

```
data test;
   L = 5;
   label L = 'Length';
run;
```

Restrictions If the label includes a semicolon (;) or an equal sign (=), you must enclose the label in either single or double quotation marks.

```
label N = 'Number = ';
```

If the label includes single quotation marks ('), you must enclose the label in double quotation marks.

```
label Name = "Owner's Last Name";
```

There is partial support for variable name substitution with the JAVAIMG device. When the label text for the variable exceeds the space allowed, the variable name is used instead by procedures that are labeling an axis or a legend title. With the exception of the SAS/GRAPH GPLOT procedure, the variable name is not used when the JAVAIMG device is specified.

Notes

Quotation marks are not required unless the label includes a semicolon, quotation marks, or an equal sign.

The LABEL statement expects an equal sign (=) after the first variable. If any other character is found, an error occurs. The LABEL statement considers everything after the equal sign, following the first variable, as part of the label until the statement comes to another variable followed by an equal sign, regardless of quotation marks. In this example, the label for x is **`this is x y 4 = this is y`** because the next variable that is followed by an equal sign is z.

```
data test;
   x = 1;
   y = 1;
   z = 1;
   label x = 'this is x' y 4 = 'this is y' z = 'This is z';
run;
```

In this LABEL statement, the x variable has the same label:

```
label x = this is x y 4 = this is y  z = This is z;
```

See

"Character Constants and Character Variables" in *SAS Programmer's Guide: Essentials*.

''

removes a label from a variable. Enclose a single blank space in quotation marks to remove an existing label. You can remove labels from multiple variables in a single LABEL statement:

```
data test;
   L=5; W=10;
   label L = ' '  W = ' ';
run;
```

Alternatively, you can remove a label by specifying nothing after the equal sign:

```
data test;
   L=5; W=10;
   label L =  W = ;
run;
```

See Also

"LABEL Statement" on page 150 for more information about the LABEL statement.

WHERE Statement

Subsets the input data set by specifying certain conditions that each observation must meet before it is available for processing.

Type: Procedure statement

Syntax

WHERE *where-expression*;

Required Argument

where-expression
is an arithmetic or logical expression that generally consists of a sequence of operands and operators. For more information about where processing, see *SAS DATA Step Statements: Reference*.

Tip You can specify multiple where-expressions.

See Also

"WHERE Statement" on page 151 for more information about the WHERE statement.

Examples: SGRENDER Procedure

About the SGRENDER Procedure Examples

Examples in This Section

This section provides basic examples of how to use the SGRENDER procedure. The following examples are provided:

Example	Description
"Example 1: Generating a Graph from a Simple GTL Template" (p. 1692)	Shows how to create and render a template for a surface plot.
"Example 2: Defining Dynamic Variables" (p. 1693)	Shows how to use dynamic variables in a template.

Examples in the SAS Graph Template Language Documentation

You will find many more examples in the Graph Template Language (GTL) documentation:

- *SAS Graph Template Language: User's Guide*
- *SAS Graph Template Language: Reference*

Examples Available Online

You can find more examples on the Graphically Speaking blog. On the blog page, use **Search Graphically Speaking** to search for GTL or SGRENDER to find the GTL and SGRENDER procedure examples.

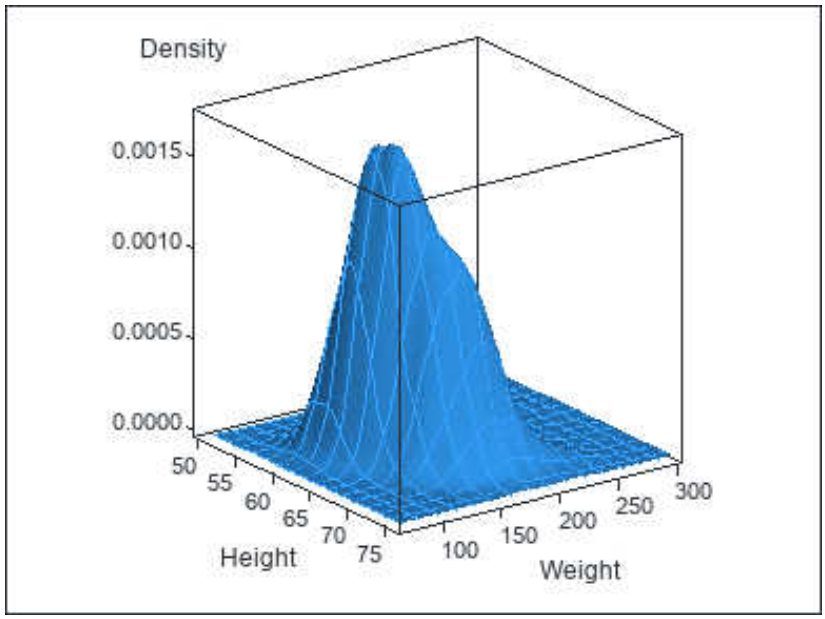
You can also find examples on support.sas.com at the following URL: https://support.sas.com/sassamples/graphgallery/PROC_SGRENDER.html.

Example 1: Generating a Graph from a Simple GTL Template

Features:	GTL template
Sample library member:	SGRENDR
Note:	For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example creates a simple StatGraph template by using Graph Template Language and then generates a graph from the template by using the SGRENDER procedure.

Output



Program

```
proc template;
  define statgraph surface;
  begingraph;
        layout overlay3d;
```

```
      surfaceplotparm x=height y=weight z=density;
        endlayout;
      endgraph;
  end;
run;

proc sgrender data=sashelp.gridded template=surface;
run;
```

Program Description

Create the Statgraph Template.

```
proc template;
  define statgraph surface;
  begingraph;
        layout overlay3d;
      surfaceplotparm x=height y=weight z=density;
        endlayout;
      endgraph;
  end;
run;
```

Generate the graphics output from the template.

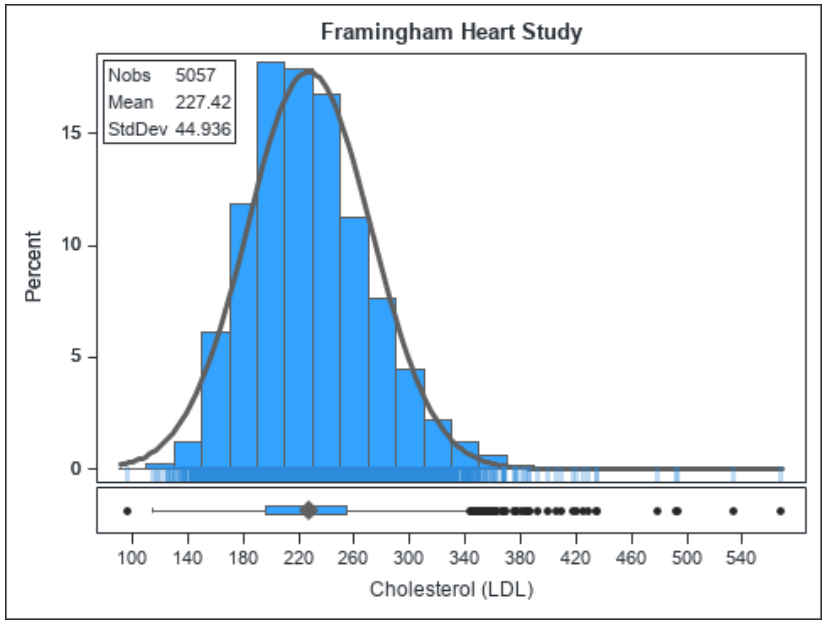
```
proc sgrender data=sashelp.gridded template=surface;
run;
```

Example 2: Defining Dynamic Variables

Features:	DYNAMIC Statement
Sample library member:	SGREND1
Note:	For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example uses dynamic variables to set values within the StatGraph template. By using dynamic variables to set the variable names, variable labels, and other parameters, the StatGraph template can be used with different data sets. The first PROC SGRENDER statement generates a graph for the SASHELP.HEART data set. The second PROC SGRENDER statement generates multiple graph for the CARS data set by using BY grouping.

Output with Heart Data



Program Template and Heart Data

```
proc template;
  define statgraph distribution;
    dynamic VAR VARLABEL TITLE NORMAL _BYLINE_;
     begingraph;
       entrytitle TITLE;
       entrytitle _BYLINE_;
       layout lattice / columns=1 rows=2  rowgutter=2px
                        rowweights=(.9 .1) columndatarange=union;
       columnaxes;
         columnaxis / label=VARLABEL;
       endcolumnaxes;
       layout overlay / yaxisopts=(offsetmin=.035);
          layout gridded / columns=2 border=true autoalign=(topleft
topright);
          entry halign=left "Nobs";
             entry halign=left eval(strip(put(n(VAR),8.)));
          entry halign=left "Mean";
             entry halign=left eval(strip(put(mean(VAR),8.2)));
             entry halign=left "StdDev";
             entry halign=left eval(strip(put(stddev(VAR),8.3)));
              endlayout;
      histogram VAR / scale=percent;
      if (exists(NORMAL))
        densityplot VAR / normal( );
              endif;
      fringeplot VAR / datatransparency=.7;
      endlayout;
      boxplot y=VAR / orient=horizontal;
    endlayout;
```

```
 endgraph;
end;
run;

proc sgrender data=sashelp.heart template=distribution;
  dynamic var="cholesterol" varlabel="Cholesterol (LDL)" normal="yes"
          title="Framingham Heart Study";
run;
title;
```

Program Description

Create the Statgraph template.

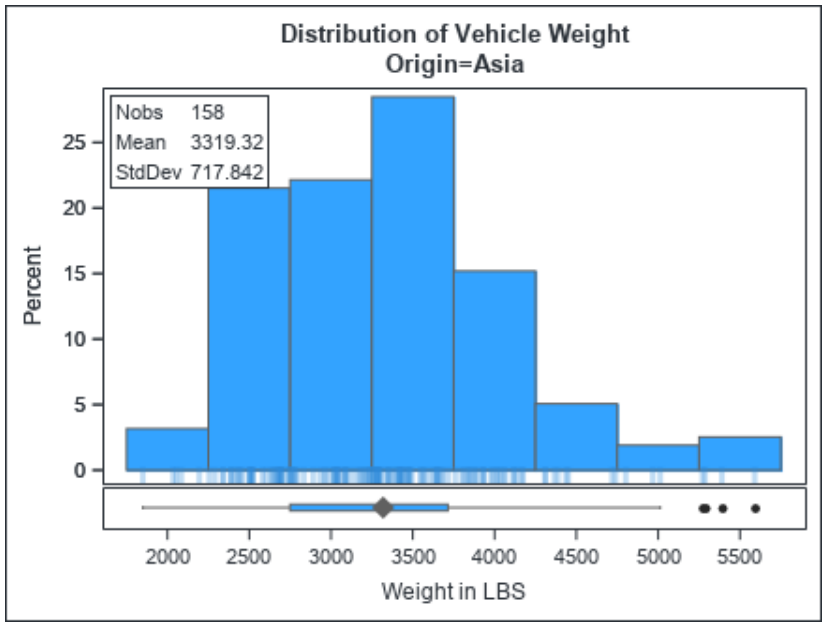
```
proc template;
  define statgraph distribution;
    dynamic VAR VARLABEL TITLE NORMAL _BYLINE_;
     begingraph;
       entrytitle TITLE;
       entrytitle _BYLINE_;
       layout lattice / columns=1 rows=2  rowgutter=2px
                        rowweights=(.9 .1) columndatarange=union;
       columnaxes;
         columnaxis / label=VARLABEL;
       endcolumnaxes;
       layout overlay / yaxisopts=(offsetmin=.035);
          layout gridded / columns=2 border=true autoalign=(topleft
topright);
          entry halign=left "Nobs";
             entry halign=left eval(strip(put(n(VAR),8.)));
          entry halign=left "Mean";
             entry halign=left eval(strip(put(mean(VAR),8.2)));
             entry halign=left "StdDev";
             entry halign=left eval(strip(put(stddev(VAR),8.3)));
              endlayout;
      histogram VAR / scale=percent;
      if (exists(NORMAL))
        densityplot VAR / normal( );
              endif;
      fringeplot VAR / datatransparency=.7;
      endlayout;
      boxplot y=VAR / orient=horizontal;
    endlayout;
 endgraph;
end;
run;
```

Generate the first graphics output from the template using the SASHELP.HEART data set. The DYNAMIC on page 1683 statement defines dynamic variables in the template.

```
proc sgrender data=sashelp.heart template=distribution;
  dynamic var="cholesterol" varlabel="Cholesterol (LDL)" normal="yes"
          title="Framingham Heart Study";
run;
```

```
title;
```

Output with Grouped Cars Data



Program for Grouped Cars Data

```
proc sort data=sashelp.cars out=cars;
  by origin;
run;

proc sgrender data=cars template=distribution;
  by origin;
  dynamic var="weight" varlabel="Weight in LBS"
          title="Distribution of Vehicle Weight";
run;
title;
```

Program Description

Sort the SASHELP.CARS data set. The data set must be sorted by the same variable that the following PROC SGRENDER block uses in its BY statement.

```
proc sort data=sashelp.cars out=cars;
  by origin;
run;
```

Generate the second graphics output from the template using the WORK.CARS data set. The BY on page 1685 statement generates multiple graphs for each value

of the BY variable.The DYNAMIC on page 1683 statement defines dynamic variables in the template.

```
proc sgrender data=cars template=distribution;
  by origin;
  dynamic var="weight" varlabel="Weight in LBS"
          title="Distribution of Vehicle Weight";
run;
title;
```

12

SGSCATTER Procedure

Overview: SGSCATTER Procedure ***1699***
What Does the SGSCATTER Procedure Do? 1700

Concepts: SGSCATTER Procedure ***1702***
Statements for Creating Panels 1702
Axis Features in the Panel Graphs 1705
Legends in the Panel 1708

Syntax: SGSCATTER Procedure ***1708***
PROC SGSCATTER Statement 1709
COMPARE Statement 1714
MATRIX Statement 1733
PLOT Statement 1742
ATTRIB Statement 1763
BY Statement 1764
FORMAT Statement 1765
LABEL Statement 1766
WHERE Statement 1768

Examples: SGSCATTER Procedure ***1769***
About the SGSCATTER Procedure Examples 1769
Example 1: Creating a Scatter Plot Matrix 1770
Example 2: Creating a Graph with Multiple Independent Scatter Plots and Spline Curves 1771
Example 3: Creating a Simple Comparative Panel 1773
Example 4: Creating a Comparative Panel with Regression Fits and Confidence Ellipses 1774

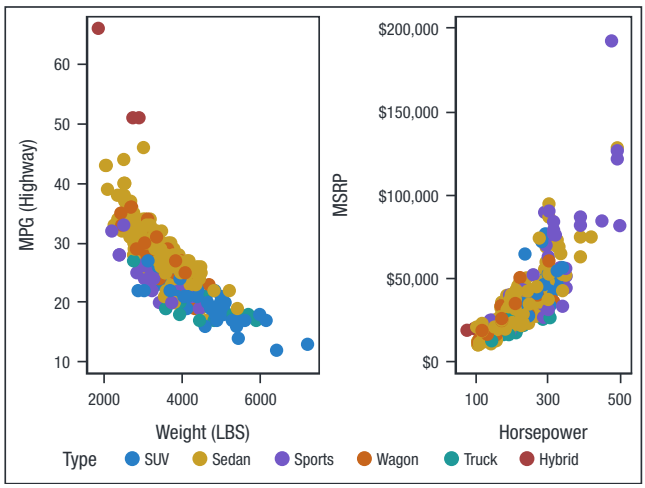
Overview: SGSCATTER Procedure

What Does the SGSCATTER Procedure Do?

The SGSCATTER procedure creates a paneled graph of scatter plots for multiple combinations of variables, depending on the plot statement that you use. You can use options to overlay fit plots and ellipses on your scatter plots.

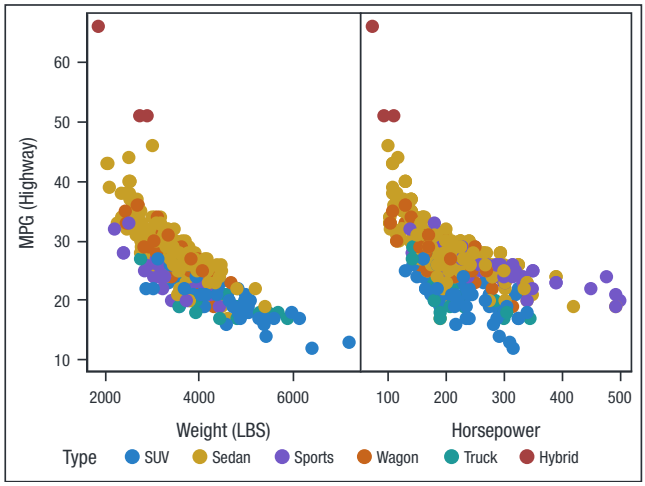
Here are examples of some types of graphs that the SGSCATTER procedure can create.

Table 12.1 *Examples of Graphs That Can Be Generated by the SGSCATTER Procedure*



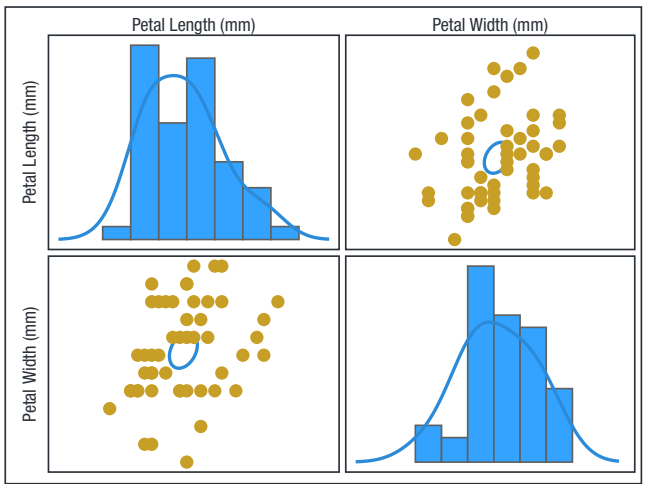
The following code creates a paneled graph of scatter plots:

```
proc sgscatter data=sashelp.cars;
   plot mpg_highway*weight msrp*horsepower /
      group=type;
run;
```



The following code creates a paneled graph of scatter plots with shared axes:

```
proc sgscatter data=sashelp.cars;
   compare y=mpg_highway
      x=(weight horsepower ) / group=type;
run;
```



The following code creates a scatter plot matrix with prediction ellipses and a diagonal with histograms and density plots:

```
proc sgscatter data=sashelp.iris
   (where=(species eq "Virginica"));
   matrix petallength petalwidth /
      markerattrs=GraphData2
      ellipse=(type=mean)
      diagonal=(histogram kernel);
run;
```

Note: The graphs that you create with the SGSCATTER procedure can have many individual graph cells. As the number of cells increases, the overall graph size does not automatically increase. To increase the graph size, use the HEIGHT= and WIDTH= options of the ODS GRAPHICS statement.

Concepts: SGSCATTER Procedure

Statements for Creating Panels

About Creating Panels

The SGSCATTER procedure contains three statements that you can use to create a paneled graph of scatter plots:

- PLOT
- COMPARE
- MATRIX

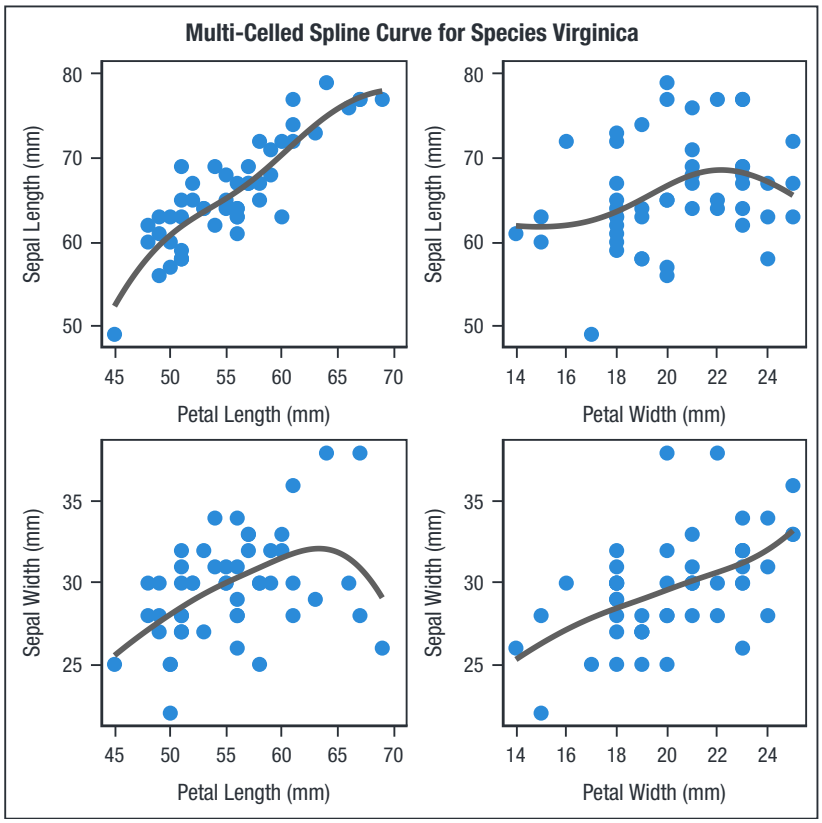
Each of the statements is specialized for creating a different type of paneled graph.

PLOT Statement

The PLOT statement is used to create a paneled graph of scatter plots where each graph cell has its own independent set of axes. Each variable pair that you specify in the PLOT statement creates an independent graph cell. You can also overlay fit plots and ellipses on each cell by using options.

Figure 12.44 shows a graph that contains four independent cells. Each cell contains a scatter plot and a fitted, penalized B-spline curve.

Figure 12.1 *Example Graph from the PLOT Statement*



By default, the axis ranges of each cell are independent from the other cells. However, you can use the UNISCALE= option to specify that all of the cells use the same axis ranges for the X axis, the Y axis, or both axes.

See "Example 2: Creating a Graph with Multiple Independent Scatter Plots and Spline Curves" on page 1771 for the code that generated this graph.

Note: It is possible to create a single scatter cell with the PLOT statement, but the SGPLOT procedure is better suited to creating a single-celled graph.

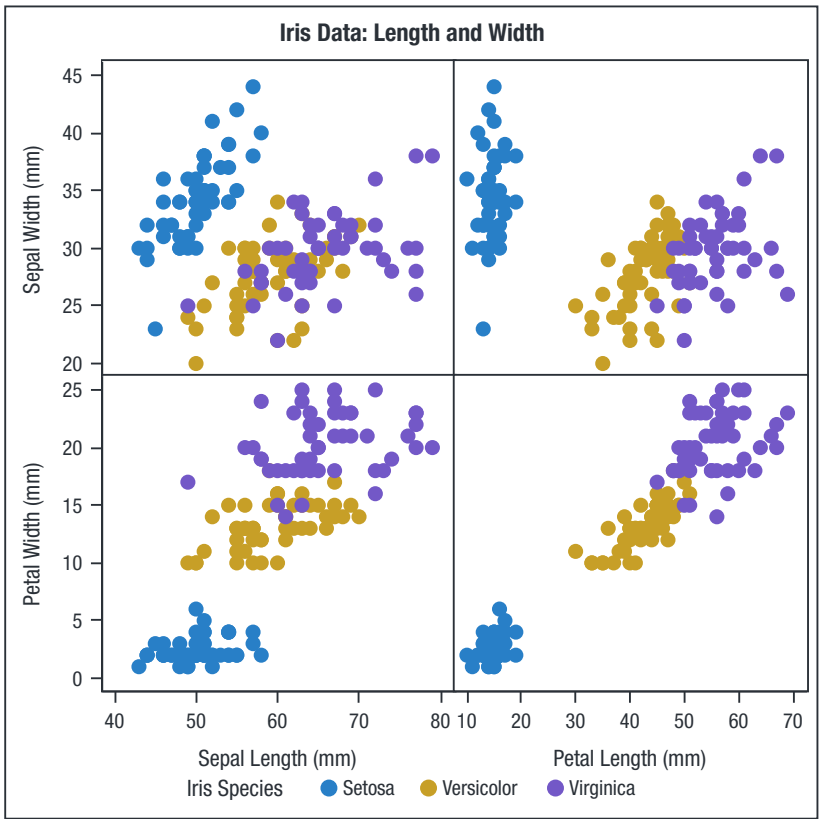
COMPARE Statement

The COMPARE statement is used to create a shared axis panel, also called an MxN matrix. The list of X and Y variables are crossed to create each cell in the graph. All cells in a row share the same row axis range. All cells in a column share the same column axis range.

When using the COMPARE statement, you can add fit plots and confidence ellipses to each cell in the panel by using options. The COMPARE statement can also be used to do simple X- or Y-axis sharing by specifying only one X or Y variable as shown in Table 12.148 on page 1700.

Figure 12.45 shows a comparative scatter plot with grouped data.

Figure 12.2 Example Graph from the COMPARE Statement



See "Example 3: Creating a Simple Comparative Panel" on page 1773 for the code that generated this graph.

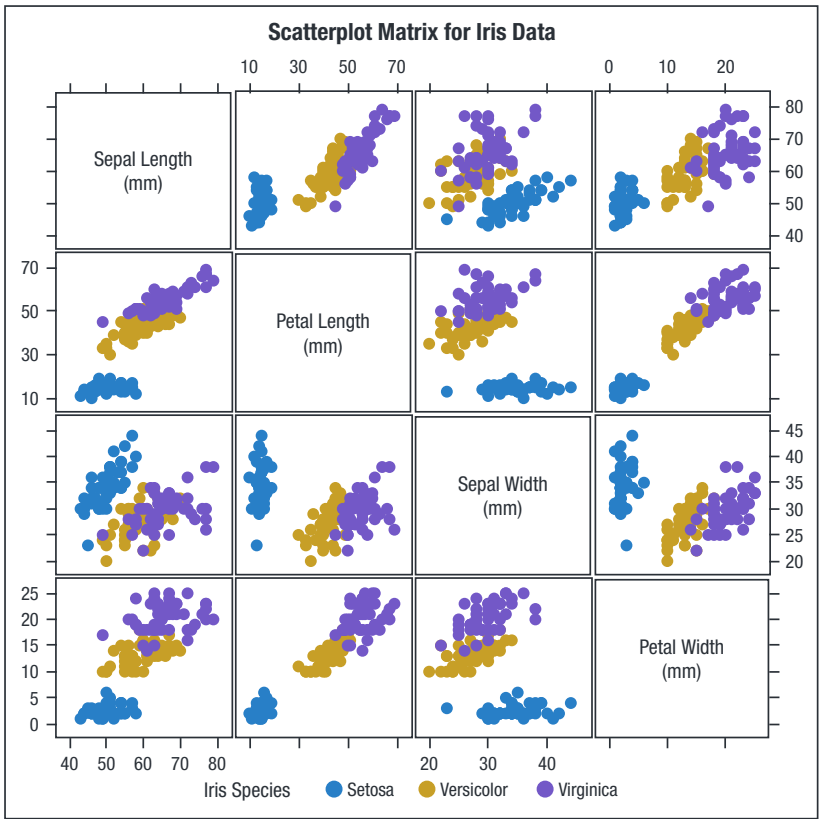
MATRIX Statement

The MATRIX statement is used to create a scatter plot matrix. Each of the variables that you specify are graphed against each other to create the graph. You can use the ELLIPSE option to overlay a confidence ellipse on each cell in the panel.

The MATRIX statement also has an option called DIAGONAL= that enables you to display the distribution of your variables in the diagonal cells of the matrix. You can place histograms in the diagonal cells, and overlay those histograms with normal density curves or kernel density estimates as shown in Table 12.148 on page 1700.

Figure 12.45 shows a scatter plot matrix of four variables with grouped data where the variable names are displayed in the diagonal.

Figure 12.3 *Example Graph from the MATRIX Statement*



See "Example 1: Creating a Scatter Plot Matrix" on page 1770 for the code that generated this graph.

Axis Features in the Panel Graphs

In 2025.05 and earlier releases, for the axes in the panel plots, the axis line, axis label, tick marks, and the tick values are always displayed. By default, an axis grid is not displayed. Starting with 2025.06, for only the SGSCATTER procedure, the display of these axis features, including the axis grid, can be controlled by the following style elements in the ODS style:

- GraphDiscreteAxis
- GraphLinearAxis
- GraphLogAxis
- GraphTimeAxis

Note: These style elements are not honored by the SGPLOT and SGPANEL procedures.

These style elements define the attribute DisplayOpts, which specifies a space-delimited list of axis features to display. The list can include one or more of Grid,

Label, Line, Ticks, and TickValues. For more information about these style elements, see Table 15.18 on page 1879. If the current ODS style does not define these style elements, the default axis features are displayed.

Currently, the following ODS styles define the GraphDiscreteAxis, GraphLinearAxis, GraphLogAxis, and GraphTimeAxis style elements:

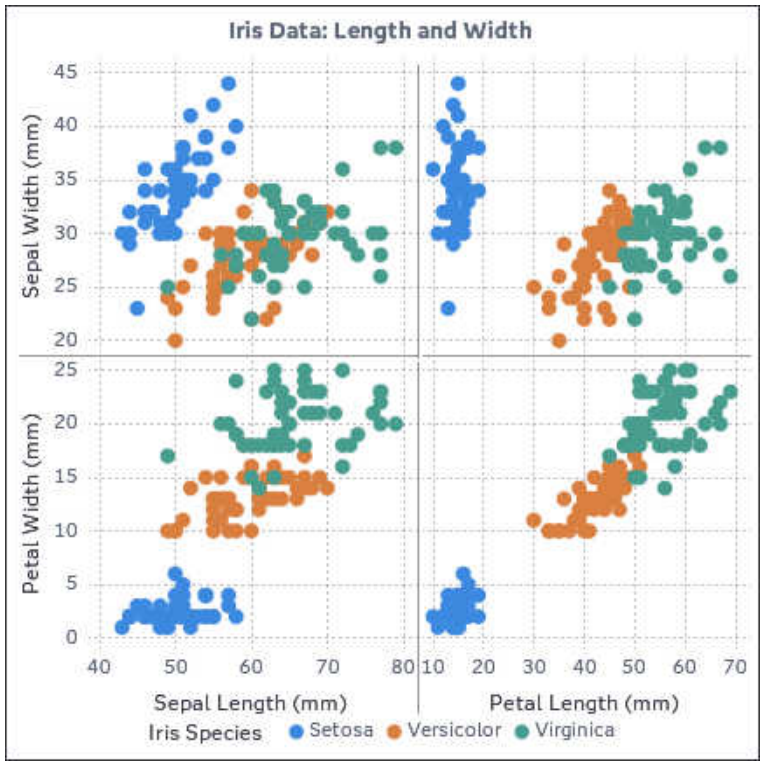
Dark2025	PowerPointDark2025
ExcelDark2025	PowerPointHighContrast2025
ExcelHighContrast2025	PowerPointIgnite
ExcelIgnite	PowerPointIlluminate
ExcelIlluminate	PowerPointLight2025
ExcelLight2025	PowerPointMidnight
ExcelMidnight	PowerPointOpal
ExcelOpal	WordDark2025
HighContrast2025	WordHigContrast2025
Ignite	WordIgnite
Illuminate	WordIlluminate
Light2025	WordLight2025
Midnight	WordMidnight
Opal	WordOpal

The following table shows the axis features that are displayed by these style elements in these ODS styles.

Table 12.2 *Axis Features Displayed*

Style Element	Style Attribute	Axis Features Displayed
GraphDiscreteAxis	DisplayOpts	TickValues Ticks Line Label
GraphLinearAxis	DisplayOpts	TickValues Label Grid
GraphLogAxis	DisplayOpts	TickValues Label Grid
GraphTimeAxis	DisplayOpts	TickValues Label Grid

To demonstrate the effect of these style elements on SGSCATTER plot axes, here is the output of “Example 3: Creating a Simple Comparative Panel” on page 1773 when it is run using ODS style Light2025:

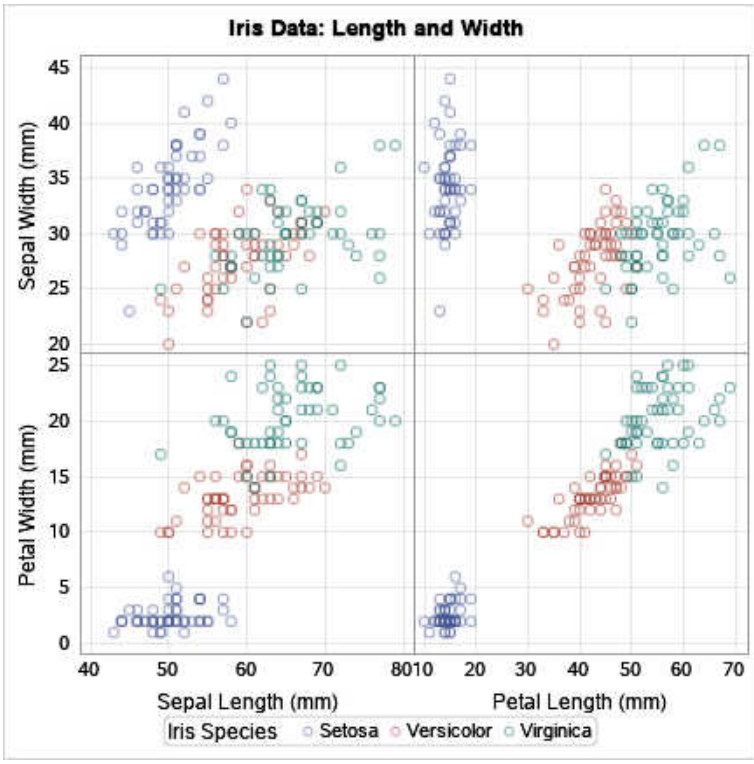


In this example, the axes are linear, so the ODS style element GraphLinearAxis has an effect. Notice that only the axis labels, tick values, and axis grids are displayed as listed in Table 12.149 on page 1706 for linear axes.

For ODS styles that do not define these style elements, you can create a custom style that adds these style elements to any existing style. For example, the following code adds these style elements to style HTMLEncore in order to create the custom style MyHTMLEncore:

```
proc template;
   define style MyHTMLEncore;
      parent=Styles.HTMLEncore;
      class GraphLinearAxis   /
         displayopts = "tickvalues label grid";
      class GraphDiscreteAxis /
         displayopts = "tickvalues ticks line label";
      class GraphTimeAxis     /
         displayopts = "tickvalues label grid";
      class GraphLogAxis      /
         displayopts = "tickvalues label grid";
   end;
run;
```

For the DisplayOpts attributes, you can specify the axis features that you want to display for each axis type. Here is the output of “Example 3: Creating a Simple Comparative Panel” on page 1773 when it is run using the custom style MyHTMLEncore:



Legends in the Panel

The SGSCATTER procedure creates a legend automatically when you specify a GROUP= variable. You can use the NOLEGEND option to disable the legend.

For all of the graph creation statements, you can use the LEGEND= option to specify the attributes of the legend.

You can also generate a continuous color legend, which maps the data range of a response variable to a range of colors. For more information, see "Using Gradient Color Legends" on page 1793.

Syntax: SGSCATTER Procedure

Requirement: One COMPARE, MATRIX, or PLOT statement is required.

PROC SGSCATTER <options>;

COMPARE X=*variable* | (*variable-1*...*variable-n*)
Y=*variable* | (*variable-1*...*variable-n*) </*options*>;

MATRIX *variable-1 variable-2* < ...*variable-n* > </*options*>;

PLOT *plot-request(s)* </*options*>;

ATTRIB *variable-list(s) attribute-list(s)*;

BY *variable-1*<...*variable-N*><*options*>;

FORMAT *variable-1*<...*variable-N*<*format-1*>><*options*>;

LABEL *variable-1=label-1*<...*variable-n=label-n*>;

WHERE *where-expression*;

PROC SGSCATTER Statement

Identifies the data set that contains the plot variables. The statement also gives you the option to specify a description and write template code to a file.

Requirement: An input data set is required.

Syntax

PROC SGSCATTER <*options*>;

Summary of Optional Arguments

BACKCOLOR=*color*
: specifies the background color of the graph area.

DATA=*input-data-set*
: specifies the SAS data set that contains the variables to process.

DATACOLORS=(*color-list*)
: specifies the fill colors for the graphics elements, such as CLM bands and filled markers.

DATACONTRASTCOLORS=(*color-list*)
: specifies the contrast colors for the graphics elements, such as lines and markers.

DATALINEPATTERNS=(*line-pattern-list*)
: specifies the list of line patterns for the graph data lines.

DATASYMBOLS=(*marker-symbol-list*)
: specifies the list of marker symbol for the graph data.

DATTRMAP=*discrete-attribute-map-data-set*
: specifies the discrete attribute map data set that you want to use with the SGSCATTER procedure.

DESCRIPTION="*text-string*"
: specifies a description for the output image.

NOSUBPIXEL

SUBPIXEL
: specifies whether subpixel rendering should be used for rendering curved lines.

OPAQUE

NOOPAQUE
specifies whether the graph background is opaque or transparent.

PAD=*numeric-value<units>* | (*pad-options*)
specifies the amount of extra space that is reserved inside the border of an annotated graph.

RATTRMAP=*range-attribute-map-data-set*
specifies the range attribute map data set that you want to use with the procedure.

SGANNO=*annotation-data-set*
specifies the SG annotation data set that you want to use.

TMPLOUT="*filename*"
specifies a file destination for the template code that is generated by the SGSCATTER procedure.

WALLCOLOR=*color*
specifies the color of the plot wall area.

Optional Arguments

BACKCOLOR=*color*
specifies the background color of the graph area. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The Color attribute of the GraphBackground style element

Examples backcolor=CXFF0000

backcolor=light_blue

DATA=*input-data-set*
specifies the SAS data set that contains the variables to process. By default, the procedure uses the most recently created SAS data set.

DATACOLORS=(*color-list*)
specifies the fill colors for the graphics elements, such as CLM bands and filled markers. Provide a space-separated list of colors enclosed in parentheses. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The colors that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Requirements The input data must be ordered by the group variable for this option to take effect. If otherwise, the results are unpredictable.

The plot statement that is used in the procedure must specify the GROUP= option.

Note When this option is specified, the colors cycle through *color-list* rather than the colors that are defined in the GraphData1–

GraphData*n* style elements. When the colors in *color-list* are exhausted, the colors repeat.

Example datacolors=(CXFF0000 red blue)

DATACONTRASTCOLORS=(*color-list*)
specifies the contrast colors for the graphics elements, such as lines and markers. Provide a space-separated list of colors enclosed in parentheses. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The colors that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Requirements The input data must be ordered by the group variable for this option to take effect. If otherwise, the results are unpredictable.

The plot statement that is used in the procedure must specify the GROUP= option.

Interaction Where applicable, the COLOR= suboption of any plot option related to a marker or line color overrides the DATACONTRASTCOLORS= option.

Note When this option is specified, the colors cycle through *color-list* rather than the colors that are defined in the GraphData1–GraphData*n* style elements. When the colors in *color-list* are exhausted, the colors repeat.

Example datacontrastcolors=(red blue CX00FF00)

DATALINEPATTERNS=(*line-pattern-list*)
specifies the list of line patterns for the graph data lines. Provide a space-separated list of line patterns enclosed in parentheses. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default The line patterns that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Requirements The input data must be ordered by the group variable for this option to take effect. If otherwise, the results are unpredictable.

The plot statement that is used in the procedure must specify the GROUP= option.

Interaction Where applicable, the PATTERN= suboption of any plot option related to line attributes overrides the DATALINEPATTERNS= option.

Note When this option is specified, the line patterns cycle through *line-pattern-list* rather than the line patterns that are defined in the GraphData1–GraphData*n* style elements. When the patterns in *line-pattern-list* are exhausted, the patterns repeat.

Example datalinepatterns=(dot solid longdash 26)

DATASYMBOLS=(*marker-symbol-list*)
specifies the list of marker symbol for the graph data. Provide a space-separated list of symbols enclosed in parentheses. See marker symbols on page 1834.

Default The symbols that are defined in the GraphData1–GraphData*n* style elements in the current style are used.

Requirements The input data must be ordered by the group variable for this option to take effect. If otherwise, the results are unpredictable.

The plot statement that is used in the procedure must specify the GROUP= option.

Interaction Where applicable, the SYMBOL= suboption of the MARKERATTRS= option overrides the DATASYMBOLS= option.

Note When this option is specified, the symbols cycle through *marker-symbol-list* rather than the symbols that are defined in the GraphData1–GraphData*n* style elements. When the symbols in *marker-symbol-list* are exhausted, the symbols repeat.

Example datasymbols=(circle square triangle star)

DATTRMAP=*discrete-attribute-map-data-set*
specifies the discrete attribute map data set that you want to use with the SGSCATTER procedure. You specify this option only if you are using a discrete attribute map to control visual attributes of the graph.

Restriction The procedures do not support an attribute map that is stored in a table. Only standard SAS data sets are supported for attribute map data.

Requirement The values in the DATTRMAP data set must be sorted by ID. If they are not, only the first value is found.

See Chapter 18, “Using Discrete Attribute Maps,” on page 1901

“Overview of Attribute Maps” on page 1897

DESCRIPTION="*text-string*"
specifies a description for the output image. The description identifies the image in the following locations:

- the alternate text for the image in HTML output
- the table of contents that is created by the CONTENTS option in an ODS statement

The default description is “The SGSCATTER Procedure”.

Alias DES=

Notes You can disable the alternate text in HTML output by specifying an empty string. That is, `DESCRIPTION=""`.

The name of the output image is specified by the IMAGENAME= option in the ODS GRAPHICS statement.

NOSUBPIXEL | SUBPIXEL
specifies whether subpixel rendering should be used for rendering curved lines. These options are obsolete and have no effect. Subpixel rendering is always enabled for the SGSCATTER procedure.

OPAQUE | NOOPAQUE
specifies whether the graph background is opaque or transparent.

Default OPAQUE

Restriction The following output formats support transparent background (NOOPAQUE): EMF, PDF, PNG, PS, and SVG, with the following exception. The PS format does not support transparent background when your output format is not vector graphics (that is, your output renders as an image due to some graph feature or you used an OUTPUTFMT= override to an image format, including PNG).

Interaction When NOOPAQUE is specified, the background color is not used.

PAD=*numeric-value<units>* | (*pad-options*)
specifies the amount of extra space that is reserved inside the border of an annotated graph.

You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see Chapter 20, "Annotating ODS Graphics," on page 1935.

This option creates margins around the graph for company logos, annotated notes, and so on. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Use pad options to create non-uniform padding. Edges that are not assigned padding are padded with the default amount.

pad-options can be one or more of the following:

LEFT= *numeric-value<units>*
specifies the amount of extra space to add to the left edge.

RIGHT= *numeric-value<units>*
specifies the amount of extra space to add to the right edge.

TOP= *numeric-value<units>*
specifies the amount of extra space to add to the top edge.

BOTTOM= *numeric-value<units>*
specifies the amount of extra space to add to the bottom edge.

RATTRMAP=*range-attribute-map-data-set*
specifies the range attribute map data set that you want to use with the procedure. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

Restriction The procedures do not support an attribute map that is stored in a table. Only standard SAS data sets are supported for attribute map data.

Requirement The values in the RATTRMAP data set must be sorted by ID. If they are not, only the first value is found.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

SGANNO=*annotation-data-set*
specifies the SG annotation data set that you want to use. You specify this option only if you are using the SG annotation feature to annotate your graph. For more information, see Chapter 20, "Annotating ODS Graphics," on page 1935.

Note: The procedures do not support annotation data that is stored in a table. Only standard SAS data sets are supported for annotation data.

TMPLOUT="*filename*"
specifies a file destination for the template code that is generated by the SGSCATTER procedure.

WALLCOLOR=*color*
specifies the color of the plot wall area. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The Color attribute of the GraphWalls style element

Examples wallcolor=CXFF0000

wallcolor=light_blue

COMPARE Statement

Creates a comparative panel of scatter plots with shared axes.

Examples: "Example 3: Creating a Simple Comparative Panel" on page 1773

"Example 4: Creating a Comparative Panel with Regression Fits and Confidence Ellipses" on page 1774

Syntax

COMPARE X=*variable* | (*variable-1*...*variable-n*)
Y=*variable* | (*variable-1*...*variable-n*) </options>;

Summary of Optional Arguments

Appearance options

ATTRID=*variable*
: specifies the value of the ID variable in an attribute map data set.

GRID
: creates grid lines for each tick on both axes.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the grid lines.

MINORGRID
: creates grid lines at each minor tick on both axes.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the minor grid lines.

NOWALL
: turns off the display of the graph wall's fill and outline.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

SPACING=*n* | (*spacing-options*)
: specifies the amount of spacing (in pixels) that is placed between the cells in the graph.

TRANSPARENCY=*numeric-value*
: specifies the degree of transparency for the plot components.

Axis options

REFTICKS

REFTICKS=(*options*)
: duplicates the tick marks from the X and Y axes on the opposite sides of the graph.

Data tip options

TIP=(*variable-list*)
: displays data tips using the data obtained from the specified variables.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
: specifies a classification variable to divide the values into groups.

Label options

DATALABEL

DATALABEL=*variable*
: displays a label for each data point.

DATALABELPOS=BOTTOM | CENTER | TOP | BOTTOMLEFT | LEFT | TOPLEFT | BOTTOMRIGHT | RIGHT | TOPRIGHT
specifies the location of the data label with respect to the plot.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Legend options

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used with the COLORRESPONSE= option.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color legend.

GRADLEGEND=(*options*)
specifies the appearance of a continuous color legend when the COLORRESPONSE= option is used.

LEGEND=(*options*)
specifies the appearance of the legend for the scatter plot.

NOGRADLEGEND
turns off the display of the continuous color legend that appears when the COLORRESPONSE= option is used.

NOLEGEND
removes the legend from the plot.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

JITTER
specifies that data markers are offset when multiple observations have the same response value.

MARKERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element*<(COLOR=*color*)> | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

Plot options

ELLIPSE

ELLIPSE=(*options*)
Adds a confidence or prediction ellipse to the scatter plot.

JOIN

JOIN=(*options*)

specifies that line segments join all of the scatter points.

LOESS

LOESS=(*options*)

Adds a loess fit to the scatter plot.

PBSPLINE

PBSPLINE=(*options*)

adds a fitted, penalized B-spline curve to the scatter plot.

REG

REG=(*options*)

adds a regression fit to the scatter plot.

Required Arguments

X=*variable* | (*variable-1*) ... (*variable-n*)

specifies one or more variables for the X axis. To specify more than one variable, enclose the list of variables in parentheses.

Y=*variable* | (*variable-1*) ... (*variable-n*)

specifies the one or more variables for the Y axis. To specify more than one variable, enclose the list of variables in parentheses.

Optional Arguments

ATTRID=*variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see Chapter 18, "Using Discrete Attribute Maps," on page 1901.

COLORMODEL=*style-element* | (*color-list*)

specifies a color ramp that is to be used with the COLORRESPONSE= option.

style-element

specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)

specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Requirement The list of colors must be enclosed in parentheses.

Example
```
colormodel=(blue yellow green)
```

Default The TwoColorAltRamp style element

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color legend.

Requirement When the COLORRESPONSE= option is used with the JOIN= option, the COLORRESPONSE= variable values must be constant for each line. If the values for any line are not constant, the COLORRESPONSE= option is ignored, and a warning message is written to the SAS log.

Notes When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

In 2023.10 and prior releases, when GROUP= and COLORRESPONSE= are specified, only a continuous legend is rendered in the graph. Starting with 2023.11, a continuous legend and a discrete legend are rendered in that case.

Tips The color ramp is specified by the COLORMODEL= option. The color ramp represents the range of unique response values.

The GRADLEGEND= option controls the title, position, and border of the legend. If you want only the colors and not the legend, then you can specify NOGRADLEGEND.

See "Using Gradient Color Legends" on page 1793

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, the value of the Y variable is used for the data label.

DATALABELPOS=BOTTOM | CENTER | TOP | BOTTOMLEFT | LEFT | TOPLEFT | BOTTOMRIGHT | RIGHT | TOPRIGHT
specifies the location of the data label with respect to the plot.

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

This option does not support the splitting or rotation of data labels.

ELLIPSE
ELLIPSE=(*options*)

Adds a confidence or prediction ellipse to the scatter plot. *options* can be one or more of the following:

ALPHA=*numeric-value*

specifies the confidence level for the confidence limits.

Default 0.05 (95% confidence)

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect if you do not specify either the CLI option or the CLM option.

CLIP

specifies that X and Y values for the ellipse are ignored when determining the data ranges for the axes.

Clipping occurs if the X or Y value for an ellipse exceeds the axis range.

Default The X and Y values for the ellipse contribute to the data ranges for the axes. If necessary, each axis is extended in order to display the entire ellipse.

TYPE=MEAN | PREDICTED

specifies the type of ellipse.

MEAN specifies a confidence ellipse for the population mean. PREDICTED specifies a prediction ellipse for a new observation. Both ellipse types assume bivariate normal distribution.

Default PREDICTED

Restriction To use this option, all of the X and Y variables must be numeric.

Interaction The GROUP option does not affect the creation of ellipses. The SGSCATTER procedure always uses all of the data points to calculate the confidence or prediction ellipse.

Tip If your graph has a large number of data points, the data markers might obscure the ellipse. You can use the TRANSPARENCY= option in the COMPARE statement to make the markers more transparent.

FILLEDOUTLINEDMARKERS

specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the “Filled” suffix in its name. Otherwise, this option is ignored.

Interaction Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see “Marker Fills and Outlines” on page 1805.

GRADLEGEND=(*options*)
specifies the appearance of a continuous color legend when the COLORRESPONSE= option is used.

options can be one or more of the following:

BORDER
adds a border around the legend.

INTEGER
specifies that integers are used for the gradient legend.

NOTITLE
removes the default title. The default title is the name of the response variable.

POSITION=TOP | BOTTOM | LEFT | RIGHT
specifies the position of the legend within the graph.

Default RIGHT

TITLE="*text-sting*"
specifies the label for the legend.

Default If you do not specify this option, the name of the response variable is displayed as the title.

Interaction This option has no effect unless the COLORRESPONSE= option is also specified.

Tip The color ramp is specified by the COLORMODEL= option. The color ramp represents the range of unique response values.

GRID
creates grid lines for each tick on both axes.

Defaults In 2025.05 and earlier releases, a grid is not displayed.

Starting with 2025.06, depending on the axis type, the DisplayOpts attribute of the GraphLinearAxis, GraphDiscreteAxis, GraphTimeAxis, or GraphLogAxis style element, if it is defined in the current ODS style and its DisplayOpts attribute specifies option GRID. See “Axis Features in the Panel Graphs” on page 1705.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default GraphGridLines style element in the current style for ungrouped data. GraphData1–GraphData*n* style elements in the current style for grouped data.

Interaction This option has no effect unless GRID is also specified.

Examples
```
GRIDATTRS=(color=green pattern=longdash thickness=2)
```

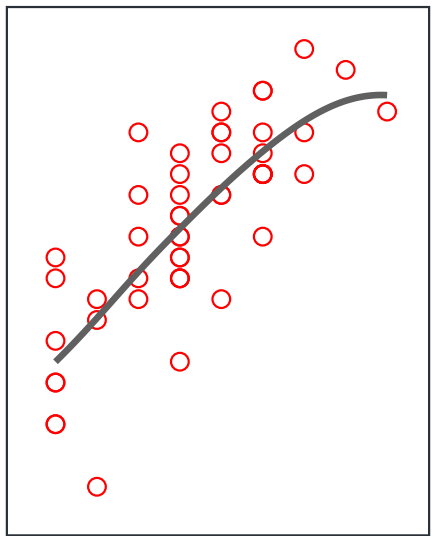
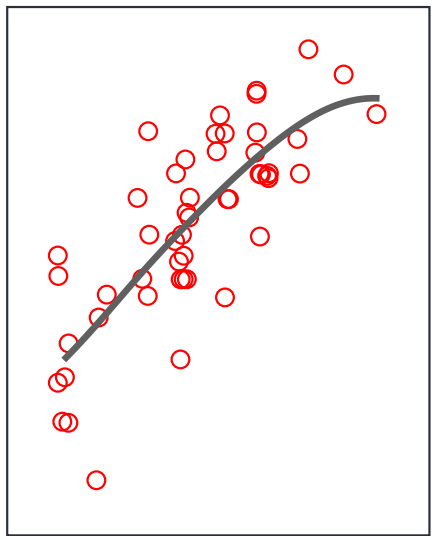
Here is an example that specifies a style element:
```
GRIDATTRS=GraphAxisLines
```

GROUP=*variable*
specifies a classification variable to divide the values into groups. If a fit line is requested, then the GROUP= variable is also applied to the fit plot unless you specify the NOGROUP suboption in the option for the fit plot.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

JITTER
specifies that data markers are offset when multiple observations have the same response value. When the JITTER option is enabled, markers that represent the same response value are offset slightly in order to make all of the markers visible.

The following partial images show the effect of the JITTER option.

Default	JITTER Specified
	

Default Markers that represent the same response value are overlaid, which results in some markers being obscured.

Notes This option affects only how the scatter plot is drawn. It has no effect on other graphics elements, such as a loess, spline, or regression curve.

By default, the width of the jitter space is 40% of the minimal interval width of the axis. When the minimum data interval is very small, the jitter offset might not be noticeable.

JOIN
JOIN=(*options*)

specifies that line segments join all of the scatter points. The line segments connect the scatter points in increasing order along the X axis. The data order of the input data set has no effect on the order of the lines.

You can specify the following options:

LINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the join lines.

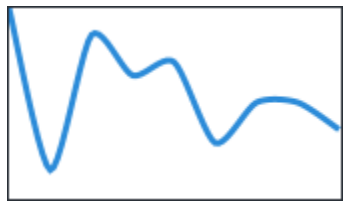
You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphFit style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

SMOOTHCONNECT
specifies that a smoothed line passes through all vertices.

The following graphics fragments show the effect of using SMOOTHCONNECT.

Default Series	SMOOTHCONNECT Specified
	

LEGEND=(*options*)
specifies the appearance of the legend for the scatter plot. *options* can be one or more of the following:

ACROSS=*n*
specifies the number of columns in the legend.

ATTRMAP
specifies that the items defined in an attribute map data set are shown in the legend regardless of whether they are used in this graph.

Requirement You must also specify the ATTRID= on page 1735 option in the COMPARE statement. Otherwise, this option is ignored.

Tips You can use this option to create a standard legend for multiple graphs.

To show all of the items defined in a discrete attribute map data set in any graph that uses it, specify SHOW="ATTRMAP" in the attribute map data set. See "Reserved Discrete Attribute Map Variables" on page 1905.

AUTOITEMSIZE
specifies that all markers in the legend are sized in proportion to the font size used for the legend labels.

The following figures show a legend with fairly large labels. In the first figure, the markers are small compared to the labels. The second figure uses AUTOITEMSIZE to size the markers in proportion to the labels.

Figure 12.4 *Default Marker Size*

Figure 12.5 *AUTOITEMSIZE Used in the Statement*

DOWN=*n*
specifies the number of rows in the legend.

Interaction If you specify both the ACROSS= and DOWN= suboptions, then the DOWN= suboption has no effect.

NOBORDER
removes the border from the legend.

NOTITLE
removes the label from the legend.

POSITION=*position-value*
specifies the position of the legend within the graph.

The positions are as follows:

BOTTOM
places the legend at the bottom of the graph.

LEFT
places the legend at the left side of the graph.

RIGHT
places the legend at the right side of the graph.

TOP
places the legend at the top of the graph.

Default BOTTOM

Note If you specify more than one legend with the same position, then all of your legends are placed at that position.

SORTORDER=ASCENDING | DESCENDING
specifies the sort order to use for the legend entry labels.

TITLE="*text-sting*"
specifies the label for the legend.

By default, the label of the group variable is used.

LOESS
LOESS=(*options*)
Adds a loess fit to the scatter plot. You can specify the following options:

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default 0.05 (95% confidence)

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect if you do not specify the CLM option.

CLM
creates confidence limits for a mean predicted value for each observation.

DEGREE=1 | 2
specifies the degree of the local polynomials to use for each local regression.

1
specifies a linear fit.

2
specifies a quadratic fit.

Default 1

INTERPOLATION=CUBIC | LINEAR
specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

Default CUBIC

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fit line.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphFit style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NOGROUP
specifies that the fit does not use the group variable from the scatter plot.

SMOOTH=*numeric-value*
specifies a smoothing parameter value.

Default A smoothing value is determined automatically.

Range (0,1]

See What is loess regression? on blogs.sas.com.

The LOESS Procedure in *SAS/STAT User's Guide* for information about the fitting methodology that the LOESS option uses.

Restriction To use this option, all of the X and Y variables must be numeric.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element contrastColor, markerSymbol, and markerSize attributes. For grouped data, the GraphDataDefault markerSize attribute, and the GraphData1-GraphData*n* style element markerSymbol and contrastColor attributes.

MARKERFILLATTRS=*style-element*<(COLOR=*color*)> | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphDataDefault style element in the current style for ungrouped data. GraphData1–GraphData*n* style elements in the current style for grouped data.

Interactions This option has no effect unless FILLEDOUTLINEDMARKERS is also specified.

This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default GraphDataDefault style element in the current style for ungrouped data. GraphData1–GraphData*n* style elements in the current style for grouped data. The effective attributes are ContrastColor and LineThickness

Interaction This option has no effect unless FILLEDOUTLINEDMARKERS is also specified.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MINORGRID
creates grid lines at each minor tick on both axes.

Interaction This option has no effect unless the GRID option is also specified or, starting with 2025.06, GRID is specified in the current ODS style. See "Axis Features in the Panel Graphs" on page 1705.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the minor grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default GraphMinorGridLines style element in the current style for ungrouped data. GraphData1–GraphData*n* style elements in the current style for grouped data.

Interaction This option has no effect unless MINORGRID is also specified.

Tip You can use GRIDATTRS= to change the appearance of the major grid lines.

Examples
```
MINORGRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:
```
MINORGRIDATTRS=GraphAxisLines
```

NOGRADLEGEND
turns off the display of the continuous color legend that appears when the COLORRESPONSE= option is used. This option might be useful when you map plot colors using a numeric column but do not want a continuous color legend.

NOLEGEND
removes the legend from the plot.

NOWALL
turns off the display of the graph wall's fill and outline. This option might be useful when your graph contains an annotation, and the wall color interferes with that annotation.

For most styles, the wall outline is the same as the axis lines, and it is impossible to see the difference. Also, the wall fill color is often the same as the graph background. However, if this is not the case with the style that you use for a graph, then you might want to suppress the wall fill and outline.

PBSPLINE
PBSPLINE=(*options*)
adds a fitted, penalized B-spline curve to the scatter plot. *options* can be one or more of the following:

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default 0.05 (95% confidence)

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect if you do not specify either the CLI option or the CLM option.

CLI
creates confidence limits for individual predicted values for each observation.

CLM
creates confidence limits for a mean predicted value for each observation.

DEGREE=*n*
specifies the degree of the spline transformation.

Default 3

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the curve line.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphFit style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NKNOTS=*n*
specifies the number of evenly spaced internal knots.

Default 100

NOGROUP
specifies that the curve does not use the group variable from the scatter plot.

SMOOTH=*numeric-value*
specifies a smoothing parameter value.

Default A smoothing value that minimizes a lack-of-smoothness penalty is determined automatically.

Range [0,1]

Tip Specify SMOOTH=0 for an ordinary B-spline fit.

See The TRANSREG Procedure in *SAS/STAT User's Guide* for information about the fitting methodology that the PBSPLINE option uses.

Restriction To use this option, all of the X and Y variables must be numeric.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

REFTICKS
REFTICKS=(*options*)
duplicates the tick marks from the X and Y axes on the opposite sides of the graph. Specify one or more of the following for *options*:

LABEL
in addition to the tick marks, displays the axis label.

NOTICKS
suppresses the tick marks.

VALUES
in addition to the tick marks, displays the values that are represented by the tick marks.

REG
REG=(*options*)
adds a regression fit to the scatter plot. *options* can be one or more of the following:

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default 0.05

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect if you do not specify either the CLI option or the CLM option.

CLI
creates confidence limits for individual predicted values for each observation.

CLM
creates confidence limits for a mean predicted value for each observation.

DEGREE=*n*
specifies the degree of the polynomial fit.

For example, 1 specifies a linear fit, 2 specifies a quadratic fit, and 3 specifies a cubic fit.

Default 1

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fit line.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphFit style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NOGROUP
specifies that the fit does not use the group variable from the scatter plot.

Restriction To use this option, all of the X and Y variables must be numeric.

SPACING=*n* | (*spacing-options*)
specifies the amount of spacing (in pixels) that is placed between the cells in the graph.

n
specifies the spacing in pixels between all cells.

(*spacing-options*)
Starting with 2023.09, one or both of the following options enclosed in parentheses:

ROW=*row-n*
specifies the spacing in pixels between cell rows.

COLUMN=*column-n*
specifies the spacing in pixels between cell columns.

Default 0

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP

specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT

specifies the horizontal alignment of the value text that is being split.

Default LEFT

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*)

displays data tips using the data obtained from the specified variables. Data tips display information when the cursor is positioned over the graphics element. Provide a space-separated list of variables enclosed in parentheses.

Requirement You must specify the IMAGEMAP=ON option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP=ON;
```

Interaction This option replaces all of the information that is displayed by default.

Note The option affects only the scatter plot in this statement.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPLABEL option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted “text strings” enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the plot components. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default 0.0

Restriction This option does not apply to LOESS, PBSPLINE, and REG plots in the scatter plot matrix.

MATRIX Statement

Creates a scatter plot matrix.

Example: "Example 1: Creating a Scatter Plot Matrix" on page 1770

Syntax

MATRIX *numeric-variable-1 numeric-variable-2* < ... *numeric-variable-n*> </*options*>;

Summary of Optional Arguments

Appearance options

ATTRID=*variable*
: specifies the value of the ID variable in an attribute map data set.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

TRANSPARENCY=*numeric-value*
: specifies the degree of transparency for the plot components.

Data tip options

TIP=(*variable-list*)
: displays data tips using the data obtained from the specified variables.

TIPFORMAT=(*format-list*)
: applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
: applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
: specifies a classification variable to divide the values into groups.

Label options

DATALABEL=*variable*
: specifies a variable that is used to create data labels for each point in the plot.

DATALABELPOS=BOTTOM | CENTER | TOP | BOTTOMLEFT | LEFT | TOPLEFT | BOTTOMRIGHT | RIGHT | TOPRIGHT
specifies the location of the data label with respect to the plot.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Legend options

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used with the COLORRESPONSE= option.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color legend.

GRADLEGEND=(*options*)
specifies the appearance of a continuous color legend when the COLORRESPONSE= option is used.

LEGEND=(*options*)
specifies the appearance of the legend for the scatter plot.

NOGRADLEGEND
turns off the display of the continuous color legend that appears when the COLORRESPONSE= option is used.

NOLEGEND
removes the legend from the graph.

Marker options

MARKERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the markers in the plot.

Plot options

DIAGONAL=(*graph-list*)
adds graphs to the diagonal cells of the plot matrix.

ELLIPSE

ELLIPSE=(*options*)
Adds a confidence or prediction ellipse to each cell that contains a scatter plot.

START=BOTTOMLEFT | TOPLEFT
specifies whether the diagonal starts in the top left corner or the bottom left corner.

Required Argument

numeric-variable-1 numeric-variable-2 < ... numeric-variable-n>
specifies two or more numeric variables for the matrix.

Optional Arguments

ATTRID=*variable*
specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see Chapter 18, "Using Discrete Attribute Maps," on page 1901.

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used with the COLORRESPONSE= option.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The TwoColorAltRamp style element

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color legend.

Notes When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

In 2023.10 and prior releases, when GROUP= and COLORRESPONSE= are specified, only a continuous legend is rendered in the graph. Starting with 2023.11, a continuous legend and a discrete legend are rendered in that case.

Tips The color ramp is specified by the COLORMODEL= option. The color ramp represents the range of unique response values.

The GRADLEGEND= option controls the title, position, and border of the legend. If you want only the colors and not the legend, then you can specify NOGRADLEGEND.

See "Using Gradient Color Legends" on page 1793

DATALABEL=*variable*
specifies a variable that is used to create data labels for each point in the plot.

DATALABELPOS=BOTTOM | CENTER | TOP | BOTTOMLEFT | LEFT | TOPLEFT | BOTTOMRIGHT | RIGHT | TOPRIGHT
specifies the location of the data label with respect to the plot.

Interactions This option has no effect unless you also specify the DATALABEL= option.

This option displays group values for each category when GROUP= is also specified.

DIAGONAL=(*graph-list*)
adds graphs to the diagonal cells of the plot matrix. If you do not specify the DIAGONAL option, the diagonal cells contain the variable names.

graph-list can be one or more of the following:

HISTOGRAM
specifies a histogram.

KERNEL
specifies a kernel density estimate.

NORMAL
specifies a normal density curve.

ELLIPSE
ELLIPSE=(*options*)
Adds a confidence or prediction ellipse to each cell that contains a scatter plot. *options* can be one or more of the following:

ALPHA=*numeric-value*
specifies the confidence level for the ellipse.

Default 0.05

Range 0.01 (complete confidence) to 0.99 (no confidence)

CLIP
specifies that X and Y values for the ellipse are ignored when determining the data ranges for the axes.

Clipping occurs if the X or Y value for an ellipse exceeds the axis range.

Default The X and Y values for the ellipse contribute to the data ranges for the axes. If necessary, each axis is extended in order to display the entire ellipse.

TYPE=MEAN | PREDICTED
specifies the type of ellipse.

MEAN specifies a confidence ellipse for the population mean. PREDICTED specifies a prediction ellipse for a new observation. Both ellipse types assume bivariate normal distribution.

Default PREDICTED

Interaction The GROUP option does not affect the creation of ellipses. The SGSCATTER procedure always uses all of the data points to calculate the confidence or prediction ellipse.

Tip If your graph has a large number of data points, the data markers might obscure the ellipse. You can use the TRANSPARENCY= option in the MATRIX statement to make the markers more transparent.

GRADLEGEND=(*options*)
specifies the appearance of a continuous color legend when the COLORRESPONSE= option is used.

options can be one or more of the following:

BORDER
adds a border around the legend.

INTEGER
specifies that integers are used for the gradient legend.

NOTITLE
removes the default title. The default title is the name of the response variable.

POSITION=TOP | BOTTOM | LEFT | RIGHT
specifies the position of the legend within the graph.

Default RIGHT

TITLE="*text-sting*"
specifies the label for the legend.

Default If you do not specify this option, the name of the response variable is displayed as the title.

Interaction This option has no effect unless the COLORRESPONSE= option is also specified.

Tip The color ramp is specified by the COLORMODEL= option. The color ramp represents the range of unique response values.

GROUP=*variable*
specifies a classification variable to divide the values into groups.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

LEGEND=(*options*)
specifies the appearance of the legend for the scatter plot.

options can be one or more of the following:

ACROSS=*n*
specifies the number of columns in the legend.

ATTRMAP
specifies that the items defined in an attribute map data set are shown in the legend regardless of whether they are used in this graph.

Requirement You must also specify the ATTRID= on page 1746 option in the MATRIX statement. Otherwise, this option is ignored.

Tips You can use this option to create a standard legend for multiple graphs.

To show all of the items defined in a discrete attribute map data set in any graph that uses it, specify SHOW="ATTRMAP" in the attribute map data set. See "Reserved Discrete Attribute Map Variables" on page 1905.

AUTOITEMSIZE
specifies that all markers in the legend are sized in proportion to the font size used for the legend labels.

The following figures show a legend with fairly large labels. In the first figure, the markers are small compared to the labels. The second figure uses AUTOITEMSIZE to size the markers in proportion to the labels.

Figure 12.6 *Default Marker Size*

Figure 12.7 *AUTOITEMSIZE Used in the Statement*

DOWN=*n*
specifies the number of rows in the legend.

Interaction If you specify both the ACROSS= and DOWN= suboptions, then the DOWN= suboption has no effect.

NOBORDER
removes the border from the legend.

NOTITLE
removes the label from the legend.

POSITION=*position-value*
specifies the position of the legend within the graph.

The values are as follows:

BOTTOM
places the legend at the bottom of the graph.

LEFT
places the legend at the left side of the graph.

RIGHT
places the legend at the right side of the graph.

TOP
places the legend at the top of the graph.

Default BOTTOM

Note If you specify more than one legend with the same position, then all of your legends are placed at that position.

SORTORDER=ASCENDING | DESCENDING
specifies the sort order to use for the legend entry labels.

TITLE="*text-sting*"
specifies the label for the legend.

By default, the label of the group variable is used.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element contrastColor, markerSymbol, and markerSize attributes. For grouped data, the GraphDataDefault markerSize attribute, and the GraphData1-GraphData*n* style element markerSymbol and contrastColor attributes.

NOGRADLEGEND
turns off the display of the continuous color legend that appears when the COLORRESPONSE= option is used. This option might be useful when you map plot colors using a numeric column but do not want a continuous color legend.

NOLEGEND
removes the legend from the graph.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, “Using Range Attribute Maps,” on page 1921

“Overview of Attribute Maps” on page 1897

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See “Fit Policies for Axis Tick Values, Curve Labels, and Data Labels” on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See “Fit Policies for Axis Tick Values, Curve Labels, and Data Labels” on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Default LEFT

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

START=BOTTOMLEFT | TOPLEFT
specifies whether the diagonal starts in the top left corner or the bottom left corner.

Default TOPLEFT

TIP=(*variable-list*)
displays data tips using the data obtained from the specified variables. Data tips display information when the cursor is positioned over the graphics element. Provide a space-separated list of variables enclosed in parentheses.

Requirement You must specify the IMAGEMAP=ON option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP=ON;
```

Interaction This option replaces all of the information that is displayed by default.

Note The option affects only the scatter plot in this statement.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPLABEL option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted "text strings" enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the plot components. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default 0.0

Restriction This option does not apply to ELLIPSE plots in the scatter plot matrix.

PLOT Statement

Creates a paneled graph that contains multiple independent scatter plots.

Example: "Example 2: Creating a Graph with Multiple Independent Scatter Plots and Spline Curves" on page 1771

Syntax

PLOT *plot-request(s)* </options>;

Summary of Optional Arguments

Appearance options

ASPECT=*positive-number*
: specifies the aspect ratio of the plot's wall area.

ATTRID=*variable*
: specifies the value of the ID variable in an attribute map data set.

COLUMNS=*n*
: specifies the number of columns in the graph.

GRID
: creates grid lines for each tick on both axes.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the grid lines.

MINORGRID
: creates grid lines at each minor tick on both axes.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
: specifies the appearance of the minor grid lines.

NOBORDER
: removes the border from each plot in the panel.

NOWALL
: turns off the display of the graph wall's fill and outline.

RATTRID=*character-value*
: specifies the value of the ID variable in a range attribute map data set.

ROWS=*n*
: specifies the number of rows in the graph.

SPACING=*n* | (*spacing-options*)
: specifies the amount of spacing (in pixels) that is placed between the cells in the graph, if the PLOT statement creates multiple cells.

TRANSPARENCY=*numeric-value*
: specifies the degree of transparency for the plot components.

Axis options

AXISEXTENT=FULL | DATA
: specifies the extent of the axis line for the axes.

REFTICKS

REFTICKS=(*options*)
: duplicates the tick marks from the X and Y axes on the opposite sides of each cell.

UNISCALE=X | Y | ALL
: specifies that the X axis, Y axis, or both axes are scaled uniformly for all of the cells in the graph.

Data tip options

TIP=(*variable-list*)
: displays data tips using the data obtained from the specified variables.

TIPFORMAT=(*format-list*)

applies formats to the list of data tip variables that you specify in the TIP= option.

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Group options

GROUP=*variable*
specifies a classification variable to divide the values into groups.

Label options

DATALABEL

DATALABEL=*variable*
displays a label for each data point.

DATALABELPOS=BOTTOM | CENTER | TOP | BOTTOMLEFT | LEFT | TOPLEFT | BOTTOMRIGHT | RIGHT | TOPRIGHT
specifies the location of the data label with respect to the plot.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split.

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Legend options

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used with the COLORRESPONSE= option.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color legend.

GRADLEGEND=(*options*)
specifies the appearance of a continuous color legend when the COLORRESPONSE= option is used.

LEGEND=(*options*)
specifies the appearance of the legend for the scatter plot.

NOGRADLEGEND
turns off the display of the continuous color legend that appears when the COLORRESPONSE= option is used.

NOLEGEND
removes the legend from the graph.

Marker options

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

JITTER
specifies that data markers are offset when multiple observations have the same response value.

MARKERATTRS=*style-element*<(***options***)> | (*options*)
specifies the appearance of the markers in the plot.

MARKERFILLATTRS=*style-element*<(COLOR=*color*)> | (COLOR=*color*)
specifies the color of the marker fill.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines.

Plot options

ELLIPSE

ELLIPSE=(*options*)
adds a confidence or prediction ellipse to the scatter plot.

JOIN

JOIN=(*options*)
specifies that line segments join all of the scatter points.

LOESS

LOESS=(*options*)
adds a loess fit to the scatter plot.

PBSPLINE

PBSPLINE=(*options*)
adds a fitted, penalized B-spline curve to the scatter plot.

REG

REG=(*options*)
adds a regression fit to the scatter plot.

Required Argument

plot-request-1 <... plot-request-n>
specifies one or more plot requests. Each plot request specifies the variables to plot and produces a separate cell. All variables must be in the input data set. Multiple plot requests are separated with blanks. You can plot character or numeric variables. A plot request can be either of these:

y-variable*x-variable
plots the values of two variables.

y-variable
variable plotted on the left vertical axis.

x-variable
variable plotted on the horizontal axis.

(y-variable(s))*(x-variable(s))
plots the values of two or more variables and produces a separate cell for each combination of Y and X variables.

That is, each Y*X pair is plotted on a separate set of axes.

y-variable(s)
variables plotted on the left vertical axes.

x-variable(s)
variables plotted on the horizontal axes.

If you use only one *y-variable* or only one *x-variable*, omit the parentheses for that variable, for example,

```
plot (temp rain)*month;
```

This plot request produces two cells, one for TEMP and MONTH and one for RAIN and MONTH.

Optional Arguments

ASPECT=*positive-number*

specifies the aspect ratio of the plot's wall area. The ratio is expressed as a positive decimal fraction representing *wall-height* divided by *wall-width*. For example, 0.75 is a 3/4 aspect ratio, and 1.0 is a square aspect ratio.

Small numbers, such as 0.01, produce a short, wide rectangular area. Larger numbers yield a taller, narrower rectangular area.

Default The wall area is sized to the maximum area that can fill the available space.

ATTRID=*variable*

specifies the value of the ID variable in an attribute map data set. You specify this option only if you are using an attribute map to control visual attributes of the graph. For more information, see Chapter 18, "Using Discrete Attribute Maps," on page 1901.

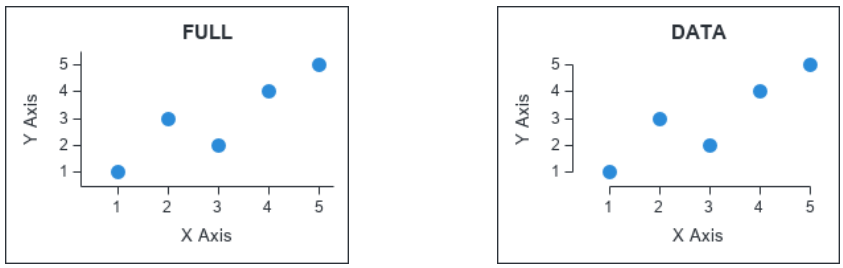
AXISEXTENT=FULL | DATA

specifies the extent of the axis line for the axes. This option turns off the wall border so that the axis lines can be clearly seen.

FULL
: the axis lines extend along the entire length of the axis.

DATA
: the axis lines extend through the data range from the minimum data point to the maximum data point.

The following figure shows a simple example of each value for the X- and Y-axis lines.



Default FULL

COLORMODEL=*style-element* | (*color-list*)
specifies a color ramp that is to be used with the COLORRESPONSE= option.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR	specifies the color for the midpoint of the range of the COLORRESPONSE= column. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value of the COLORRESPONSE= column.

Example `colormodel=TwoColorRamp`

(*color-list*)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Requirement The list of colors must be enclosed in parentheses.

Example `colormodel=(blue yellow green)`

Default The TwoColorAltRamp style element

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=*numeric-column*
specifies the numeric column that is used to map colors to a continuous color legend.

Requirement When the COLORRESPONSE= option is used with the JOIN= option, the COLORRESPONSE= variable values must be constant for each line. If the values for any line are not constant, the COLORRESPONSE= option is ignored, and a warning message is written to the SAS log.

Notes When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

In 2023.10 and prior releases, when GROUP= and COLORRESPONSE= are specified, only a continuous legend is

rendered in the graph. Starting with 2023.11, a continuous legend and a discrete legend are rendered in that case.

Tips The color ramp is specified by the COLORMODEL= option. The color ramp represents the range of unique response values.

The GRADLEGEND= option controls the title, position, and border of the legend. If you want only the colors and not the legend, then you can specify NOGRADLEGEND.

See "Using Gradient Color Legends" on page 1793

COLUMNS=*n*
specifies the number of columns in the graph.

DATALABEL
DATALABEL=*variable*
displays a label for each data point. If you specify a variable, the values of that variable are used for the data labels. If you do not specify a variable, the value of the Y variable is used for the data label.

DATALABELPOS=BOTTOM | CENTER | TOP | BOTTOMLEFT | LEFT | TOPLEFT | BOTTOMRIGHT | RIGHT | TOPRIGHT
specifies the location of the data label with respect to the plot.

Interactions This option has no effect unless you also specify the DATALABEL option.

This option displays group values for each category when GROUP= is also specified.

This option does not support the splitting or rotation of data labels.

ELLIPSE
ELLIPSE=(*options*)
adds a confidence or prediction ellipse to the scatter plot. *options* can be one or more of the following:

ALPHA=*n*
specifies the confidence level for the ellipse.

Default 0.05

Range 0.01 (complete confidence) to 0.99 (no confidence)

CLIP
specifies that X and Y values for the ellipse are ignored when determining the data ranges for the axes.

Clipping occurs if the X or Y value for an ellipse exceeds the axis range.

Default The X and Y values for the ellipse contribute to the data ranges for the axes. If necessary, each axis is extended in order to display the entire ellipse.

TYPE=MEAN | PREDICTED
specifies the type of ellipse.

MEAN
specifies a confidence ellipse for the population mean.

PREDICTED
specifies a prediction ellipse for a new observation.

Default PREDICTED

Note Both ellipse types assume bivariate normal distribution.

Restriction To use this option, all of the X and Y variables must be numeric.

Interaction The GROUP option does not affect the creation of ellipses. The SGSCATTER procedure always uses all of the data points to calculate the confidence or prediction ellipse.

Tip If your graph has a large number of data points, the data markers might obscure the ellipse. You can use the TRANSPARENCY= option in the PLOT statement to make the markers more transparent.

FILLEDOUTLINEDMARKERS
specifies that markers have a fill and an outline.

Requirement The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the "Filled" suffix in its name. Otherwise, this option is ignored.

Interaction Use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify attributes for the fill and outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

GRADLEGEND=(*options*)
specifies the appearance of a continuous color legend when the COLORRESPONSE= option is used.

options can be one or more of the following:

BORDER
adds a border around the legend.

INTEGER
specifies that integers are used for the gradient legend.

NOTITLE
removes the default title. The default title is the name of the response variable.

POSITION=TOP | BOTTOM | LEFT | RIGHT
specifies the position of the legend within the graph.

Default RIGHT

TITLE="*text-sting*"
specifies the label for the legend.

Default If you do not specify this option, the name of the response variable is displayed as the title.

Interaction This option has no effect unless the COLORRESPONSE= option is also specified.

Tip The color ramp is specified by the COLORMODEL= option. The color ramp represents the range of unique response values.

GRID
creates grid lines for each tick on both axes.

Defaults In 2025.05 and earlier releases, a grid is not displayed.

Starting with 2025.06, depending on the axis type, the DisplayOpts attribute of the GraphLinearAxis, GraphDiscreteAxis, GraphTimeAxis, or GraphLogAxis style element, if it is defined in the current ODS style and its DisplayOpts attribute specifies option GRID. See "Axis Features in the Panel Graphs" on page 1705.

GRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default GraphGridLines style element in the current style for ungrouped data. GraphData1–GraphData*n* style elements in the current style for grouped data.

Interaction This option has no effect unless GRID is also specified.

Examples
```
GRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:
```
GRIDATTRS=GraphAxisLines
```

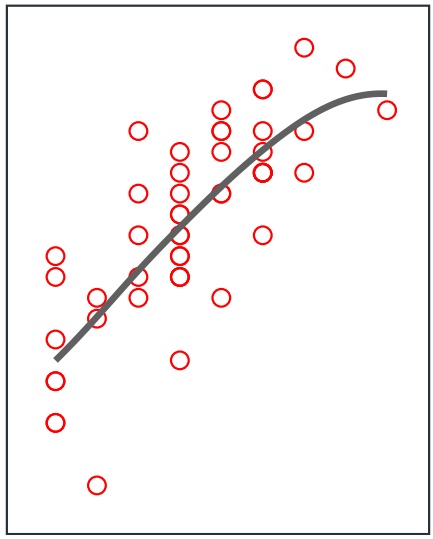
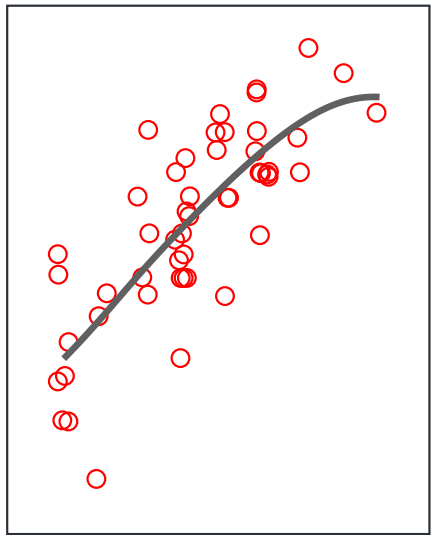
GROUP=*variable*
specifies a classification variable to divide the values into groups. If a fit line is requested, then the GROUP= variable is also applied to the fit plot unless you specify the NOGROUP suboption in the option for the fit plot.

Note When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

JITTER
specifies that data markers are offset when multiple observations have the same response value. When the JITTER option is enabled, markers that

represent the same response value are offset slightly in order to make all of the markers visible.

The following partial images show the effect of the JITTER option.

Default	JITTER Specified
	

Default Markers that represent the same response value are overlaid, which results in some markers being obscured.

Notes This option affects only how the scatter plot is drawn. It has no effect on other graphics elements, such as a loess, spline, or regression curve.

By default, the width of the jitter space is 40% of the minimal interval width of the axis. When the minimum data interval is very small, the jitter offset might not be noticeable.

JOIN
JOIN=(*options*)

specifies that line segments join all of the scatter points. The line segments connect the scatter points in increasing order along the X axis. The data order of the input data set has no effect on the order of the lines.

You can specify the following options:

LINEATTRS=*style-element*<(*options*)> | (*options*)

specifies the appearance of the join lines.

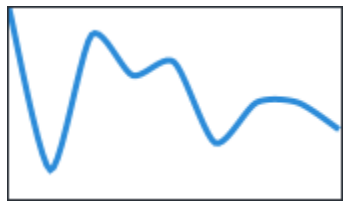
You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphFit style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

SMOOTHCONNECT
specifies that a smoothed line passes through all vertices.

The following graphics fragments show the effect of using SMOOTHCONNECT.

Default Series	SMOOTHCONNECT Specified
	

LEGEND=(*options*)
specifies the appearance of the legend for the scatter plot. *options* can be one or more of the following:

ACROSS=*n*
specifies the number of columns in the legend.

ATTRMAP
specifies that the items defined in an attribute map data set are shown in the legend regardless of whether they are used in this graph.

Requirement You must also specify the ATTRID= on page 1746 option in the PLOT statement. Otherwise, this option is ignored.

Tips You can use this option to create a standard legend for multiple graphs.

To show all of the items defined in a discrete attribute map data set in any graph that uses it, specify SHOW="ATTRMAP" in the attribute map data set. See "Reserved Discrete Attribute Map Variables" on page 1905.

AUTOITEMSIZE
specifies that all markers in the legend are sized in proportion to the font size used for the legend labels.

The following figures show a legend with fairly large labels. In the first figure, the markers are small compared to the labels. The second figure uses AUTOITEMSIZE to size the markers in proportion to the labels.

Figure 12.8 *Default Marker Size*

Figure 12.9 *AUTOITEMSIZE Used in the Statement*

DOWN=*n*
specifies the number of rows in the legend.

Interaction This suboption has no effect if you also specify the ACROSS= suboption.

LOCATION=CELL | OUTSIDE
specifies whether the legend is placed inside an empty cell (CELL) in the plot area or outside of the plot area (OUTSIDE).

The LOCATION=CELL suboption should be specified only when the graph contains an empty cell.

Note If you specify LOCATION=CELL, then auto-wrapping of the legend might produce unexpected results. In that case, use the ACROSS= or DOWN= suboption to specify the number of columns or rows.

NOBORDER
removes the border from the legend.

NOTITLE
removes the label from the legend.

POSITION=*position-value*
specifies the position of the legend within the graph.

The positions are as follows:

BOTTOM
places the legend at the bottom of the graph.

LEFT
places the legend at the left side of the graph.

RIGHT
places the legend at the right side of the graph.

TOP
places the legend at the top of the graph.

Default BOTTOM

Interaction This suboption has no effect if you also specify LOCATION= CELL.

Note If you specify more than one legend with the same position, then all of your legends are placed at that position.

SORTORDER=ASCENDING | DESCENDING
specifies the sort order to use for the legend entry labels.

TITLE="*text-sting*"
specifies the label for the legend.

By default, the label of the group variable is used.

LOESS
LOESS=(*options*)
adds a loess fit to the scatter plot. You can specify the following options:

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default 0.05

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect if you do not specify the CLM option.

CLM
creates confidence limits for a mean predicted value for each observation.

DEGREE=1 | 2
specifies the degree of the local polynomials to use for each local regression.

1
specifies a linear fit.

2
specifies a quadratic fit.

Default 1

INTERPOLATION=CUBIC | LINEAR
specifies the degree of the interpolating polynomials that are used for blending local polynomial fits at the kd tree vertices.

Default CUBIC

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fit line.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Defaults The ContrastColor, LineStyle, and LineThickness attributes of the GraphFit style element in the current style.

For ungrouped data, the GraphFit style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NOGROUP
specifies that the fit does not use the group variable from the scatter plot.

SMOOTH=*numeric-value*
specifies a smoothing parameter value.

Default A smoothing value is determined automatically.

Range (0,1]

Restriction To use this option, all of the X and Y variables must be numeric.

See What is loess regression? on blogs.sas.com.

The LOESS Procedure in *SAS/STAT User's Guide* for information about the fitting methodology that the LOESS option uses.

Restriction To use this option, all of the X and Y variables must be numeric.

MARKERATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of marker attributes, see "Marker Attributes and Symbols" on page 1833.

Default For ungrouped data, the GraphDataDefault style element contrastColor, markerSymbol, and markerSize attributes. For grouped data, the GraphDataDefault markerSize attribute, and the GraphData1-GraphData*n* style element markerSymbol and contrastColor attributes.

MARKERFILLATTRS=*style-element*<(COLOR=*color*)> | (COLOR=*color*)
specifies the color of the marker fill. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphDataDefault style element in the current style for ungrouped data. GraphData1–GraphData*n* style elements in the current style for grouped data.

Interactions This option has no effect unless FILLEDOUTLINEDMARKERS is also specified.

This option overrides any color that is specified with the MARKERATTRS= option.

Tip You can also use the MARKEROUTLINEATTRS= option to specify attributes for the marker outline.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MARKEROUTLINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the marker outlines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For the options, you can specify either or both of the following:

- line color
- line thickness

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default GraphDataDefault style element in the current style for ungrouped data. GraphData1–GraphData*n* style elements in the current style for grouped data. The effective attributes are ContrastColor and LineThickness

Interaction This option has no effect unless FILLEDOUTLINEDMARKERS is also specified.

Tip You can also use the MARKERFILLATTRS= option to specify attributes for the fill.

See For usage information and an example, see "Marker Fills and Outlines" on page 1805.

MINORGRID
creates grid lines at each minor tick on both axes.

Interaction This option has no effect unless the GRID option is also specified or, starting with 2025.06, GRID is specified in the current ODS style. See "Axis Features in the Panel Graphs" on page 1705.

MINORGRIDATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the minor grid lines. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default GraphMinorGridLines style element in the current style for ungrouped data. GraphData1–GraphData*n* style elements in the current style for grouped data.

Interaction This option has no effect unless MINORGRID is also specified.

Tip You can use GRIDATTRS= to change the appearance of the major grid lines.

Examples
```
MINORGRIDATTRS=(color=green pattern=longdash thickness=2)
```

Here is an example that specifies a style element:

```
MINORGRIDATTRS=GraphAxisLines
```

NOBORDER
removes the border from each plot in the panel.

NOGRADLEGEND
turns off the display of the continuous color legend that appears when the COLORRESPONSE= option is used. This option might be useful when you map plot colors using a numeric column but do not want a continuous color legend.

NOLEGEND
removes the legend from the graph.

NOWALL
turns off the display of the graph wall's fill and outline. This option might be useful when your graph contains an annotation, and the wall color interferes with that annotation.

For most styles, the wall outline is the same as the axis lines, and it is impossible to see the difference. Also, the wall fill color is often the same as the graph background. However, if this is not the case with the style that you use for a graph, then you might want to suppress the wall fill and outline.

PBSPLINE
PBSPLINE=(*options*)
adds a fitted, penalized B-spline curve to the scatter plot. *options* can be one or more of the following:

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default 0.05

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect if you do not specify either the CLI option or the CLM option.

CLI
creates confidence limits for individual predicted values for each observation.

CLM
creates confidence limits for a mean predicted value for each observation.

DEGREE=*n*
specifies the degree of the spline transformation.

Default 3

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the curve line.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphFit style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NKNOTS=*n*
specifies the number of evenly spaced internal knots.

Default 100

NOGROUP
specifies that the curve does not use the group variable from the scatter plot.

SMOOTH=*numeric-value*
specifies a smoothing parameter value.

Default A smoothing value that minimizes a lack-of-smoothness penalty is determined automatically.

Range [0,1]

Tip Specify SMOOTH=0 for an ordinary B-spline fit.

See The TRANSREG Procedure in *SAS/STAT User's Guide* for information about the fitting methodology that the PBSPLINE option uses.

Restriction To use this option, all of the X and Y variables must be numeric.

RATTRID=*character-value*
specifies the value of the ID variable in a range attribute map data set. You specify this option only if you are using a range attribute map to control visual attributes of the graph.

See Chapter 19, "Using Range Attribute Maps," on page 1921

"Overview of Attribute Maps" on page 1897

REFTICKS
REFTICKS=(*options*)
duplicates the tick marks from the X and Y axes on the opposite sides of each cell. You can also specify *options*:

LABEL
in addition to the tick marks, displays the axis label.

NOTICKS
suppresses the tick marks.

VALUES
in addition to the tick marks, displays the values that are represented by the tick marks.

REG
REG=(*options*)
adds a regression fit to the scatter plot. *options* can be one or more of the following:

ALPHA=*numeric-value*
specifies the confidence level for the confidence limits.

Default 0.05

Range 0.01 (complete confidence) to 0.99 (no confidence)

Interaction This option has no effect if you do not specify either the CLI option or the CLM option.

CLI
creates confidence limits for individual predicted values for each observation.

CLM
creates confidence limits for a mean predicted value for each observation.

DEGREE=*n*
specifies the degree of the polynomial fit.

For example, 1 specifies a linear fit, 2 specifies a quadratic fit, and 3 specifies a cubic fit.

Default 1

LINEATTRS=*style-element*<(*options*)> | (*options*)
specifies the appearance of the fit line.

You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

For a description of the line options, see "Line Attributes and Patterns" on page 1828.

Default For ungrouped data, the GraphFit style element in the current style. For grouped data, a GraphData1–GraphData*n* style element in the current style. The effective attributes are: ContrastColor, LineStyle, and LineThickness.

NOGROUP
specifies that the fit does not use the group variable from the scatter plot.

Restriction To use this option, all of the X and Y variables must be numeric.

ROWS=*n*
specifies the number of rows in the graph.

Interaction This option has no effect if you specify the COLUMNS= option.

SPACING=*n* | (*spacing-options*)
specifies the amount of spacing (in pixels) that is placed between the cells in the graph, if the PLOT statement creates multiple cells.

n
specifies the spacing in pixels between all cells.

Default 10

(*spacing-options*)
Starting with 2023.09, one or both of the following options enclosed in parentheses:

ROW=*row-n*
specifies the spacing in pixels between cell rows.

COLUMN=*column-n*
specifies the spacing in pixels between cell columns.

Note If you specify only ROW= or COLUMN=, the spacing for the other defaults to 0 instead of 10.

SPLITCHAR="*character-list*"
specifies one or more characters on which the text for data labels is always split. The text is split at every occurrence of the specified split character or characters.

"*character-list*" is one or more characters with no delimiter between each character and enclosed in quotation marks. For example, to specify the split characters a, b, and c, use the following option:

```
SPLITCHAR="abc"
```

When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

If the value does not contain any of the specified split characters, a split does not occur.

Default Values are not split.

Interactions This option has no effect unless DATALABEL is specified.

When the text is split, the split characters are not included in the displayed value by default. If you want the split characters to appear in the values, then also specify SPLITCHARNODROP.

You can specify the justification of the text by using the SPLITJUSTIFY= option.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITCHARNODROP
specifies that the split characters are included in the displayed value.

Interaction This option has no effect unless SPLITCHAR= is also specified.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

SPLITJUSTIFY=LEFT | CENTER | RIGHT
specifies the horizontal alignment of the value text that is being split.

Default LEFT

Interaction This option has no effect unless you specify the SPLITCHAR= option.

See "Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" on page 1802

TIP=(*variable-list*)
displays data tips using the data obtained from the specified variables. Data tips display information when the cursor is positioned over the graphics element. Provide a space-separated list of variables enclosed in parentheses.

Requirement You must specify the IMAGEMAP=ON option in the ODS GRAPHICS statement in order to generate data tips. For example, add the following statement before your procedure:

```
ODS GRAPHICS / IMAGEMAP=ON;
```

Interaction This option replaces all of the information that is displayed by default.

Note The option affects only the scatter plot in this statement.

Tip Use the TIPFORMAT and TIPLABEL options to assign formats and labels to the list of variables.

Example tip=(age weight)

TIPFORMAT=(*format-list*)
applies formats to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of formats enclosed in parentheses. This option provides a way to control the format of the variable data that appears in the data tips.

A one-to-one correspondence exists between the *format-list* and the *variable-list* that is specified for the TIP= option. A format must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a format to a variable, use the AUTO keyword instead.

Default The column format of the tip variable, or BEST6 if no format is assigned to a numeric column

Requirement A format or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPLABEL option to assign labels to the list of variables.

See *SAS Formats and Informats: Reference*

Example tipformat=(auto F5.2)

TIPLABEL=(*label-list*)
applies labels to the list of data tip variables that you specify in the TIP= option.

Provide a space-separated list of quoted "text strings" enclosed in parentheses. This option provides a way to specify labels for the variable data that appears in the data tips.

A one-to-one correspondence exists between the *label-list* and the *variable-list* that is specified for the TIP= option. A label must be provided for each variable, using the same order as the *variable-list*. If you do not want to apply a label to a variable, use the AUTO keyword instead.

Requirement A label or the keyword AUTO must be provided for each variable that is listed in the TIP= option.

Interaction This option has no effect unless TIP= is also specified.

Tip Use the TIPFORMAT option to assign formats to the list of variables.

Example tiplabel=(auto "Class Weight")

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the plot components. Specify a value from 0.0 (completely opaque) to 1.0 (completely transparent).

Default 0.0

Restriction This option does not apply to LOESS, PBSPLINE, and REG plots in the scatter plot matrix.

UNISCALE=X | Y | ALL
specifies that the X axis, Y axis, or both axes are scaled uniformly for all of the cells in the graph. By default, no scaling is performed. If this option is used, then all variables on the specified axis must be of the same type (all numeric or all character). In addition, mixing character and numeric variables with this option can cause blank plots to occur.

Note When multiple variables are specified for X or Y, the tick-value format for the X or Y axis in each plot is determined by the plot variable. If the

variables in the list have different formats, the tick-value formats in the plots differ.

ATTRIB Statement

Associates a format, informat, label, and length with one or more variables.

Type: Procedure statement

Syntax

ATTRIB *variable-list(s) attribute-list(s)*;

Arguments

variable-list(s)
names the variables that you want to associate with the attributes.

> **TIP** List the variables in any form that SAS allows.

attribute-list(s)
specifies one or more attributes to assign to *variable-list*. Specify one or more of these attributes in the ATTRIB statement:

FORMAT=*format*
associates a format with variables in *variable-list*.

Tip The format can be either a standard SAS format or a format that is defined with the FORMAT procedure.

INFORMAT=*informat*
associates an informat with variables in *variable-list*.

Tip The informat can be either a standard SAS informat or an informat that is defined with the FORMAT procedure.

LABEL='*label*'
associates a label with variables in *variable-list*.

See Also

"ATTRIB Statement" on page 142 for more information about the ATTRIB statement.

BY Statement

Orders the output according to the BY groups.

Type: Procedure statement

Syntax

BY <DESCENDING> *variable-1*
<-<DESCENDING> *variable-n*>
<NOTSORTED>;

Required Argument

variable
specifies the variable that the procedure uses to form BY groups. You can specify more than one variable. If you do not use the NOTSORTED option in the BY statement, then either the observations in the data set must be sorted by all the variables that you specify, or they must be indexed appropriately. Variables in a BY statement are called *BY variables*.

Optional Arguments

DESCENDING
specifies that the observations are sorted in descending order by the variable that immediately follows the word DESCENDING in the BY statement.

NOTSORTED
specifies that observations are not necessarily sorted in alphabetic or numeric order. The observations are grouped in another way (for example, chronological order).

The requirement for ordering or indexing observations according to the values of BY variables is suspended for BY-group processing when you use the NOTSORTED option. In fact, the procedure does not use an index if you specify NOTSORTED. The procedure defines a BY group as a set of contiguous observations that have the same values for all BY variables. If observations with the same values for the BY variables are not contiguous, then the procedure treats each contiguous set as a separate BY group.

You cannot use the NOTSORTED option in a PROC SORT step.

Note: You cannot use the GROUPFORMAT option, which is available in the BY statement in a DATA step, in a BY statement in any PROC step.

See Also

"BY Statement" on page 143 for more information about the BY statement.

FORMAT Statement

Associates formats with variables.

Type: Procedure statement

Syntax

FORMAT *variable-1* <-*variable-n*> <*format-1*> <*variable-1* <-*variable-n*> <*format-2*> > - <DEFAULT=*default-format*>;

Required Argument

variable
names one or more variables for SAS to associate with a format. You must specify at least one *variable*.

Tip To disassociate a format from a variable, use the variable in a FORMAT statement without specifying a format in a DATA step or in PROC DATASETS. In a DATA step, place this FORMAT statement after the SET statement. See "Removing a Format" in *SAS DATA Step Statements: Reference*. You can also use PROC DATASETS.

Optional Arguments

format
specifies the format that is listed for writing the values of the variables.

Tip Formats that are associated with variables by using a FORMAT statement behave like formats that are used with a colon modifier in a subsequent PUT statement. For more information about using a colon modifier, see "PUT Statement: List" in *SAS DATA Step Statements: Reference*.

See *SAS Formats and Informats: Reference*

DEFAULT=*default-format*
specifies a temporary default format for displaying the values of variables that are not listed in the FORMAT statement. These default formats apply only to the current DATA step; they are not permanently associated with variables in the output data set.

A DEFAULT= format specification applies to

- variables that are not named in a FORMAT or ATTRIB statement

- variables that are not permanently associated with a format within a SAS data set
- variables that are not written with the explicit use of a format.

Default If you omit DEFAULT=, SAS uses BEST*w*. as the default numeric format and $*w*. as the default character format.

Restriction Use this option only in a DATA step.

Tip A DEFAULT= specification can occur anywhere in a FORMAT statement. It can specify either a numeric default, a character default, or both.

Example "Assigning Formats and Defaults" in *SAS DATA Step Statements: Reference*

See Also

"FORMAT Statement" on page 146 for more information about the FORMAT statement.

LABEL Statement

Assigns descriptive labels to variables.

Type: Procedure statement

Syntax

Form 1: Creates a descriptive label for one or more variables.

LABEL *variable-1*='*text-string*' <...*variable-n*='*text-string*>;

Form 2: Removes the label from one or more variables.

LABEL *variable-1* = ' ' < ...*variable-n*=' '>; | *variable-1* = <...*variable-n*= >

Arguments

variable
specifies the variable that you want to label. You can specify labels for multiple variables in a single LABEL statement:

```
data test;
   L = 5;
   W = 10;
   label L = 'Length' W = 'Width';
run;
```

text-string

specifies a label of up to 256 bytes.

```
data test;
   L = 5;
   label L = 'Length';
run;
```

Restrictions

If the label includes a semicolon (;) or an equal sign (=), you must enclose the label in either single or double quotation marks.

```
label N = 'Number = ';
```

If the label includes single quotation marks ('), you must enclose the label in double quotation marks.

```
label Name = "Owner's Last Name";
```

There is partial support for variable name substitution with the JAVAIMG device. When the label text for the variable exceeds the space allowed, the variable name is used instead by procedures that are labeling an axis or a legend title. With the exception of the SAS/GRAPH GPLOT procedure, the variable name is not used when the JAVAIMG device is specified.

Notes

Quotation marks are not required unless the label includes a semicolon, quotation marks, or an equal sign.

The LABEL statement expects an equal sign (=) after the first variable. If any other character is found, an error occurs. The LABEL statement considers everything after the equal sign, following the first variable, as part of the label until the statement comes to another variable followed by an equal sign, regardless of quotation marks. In this example, the label for x is **`this is x y 4 = this is y`** because the next variable that is followed by an equal sign is z.

```
data test;
   x = 1;
   y = 1;
   z = 1;
   label x = 'this is x' y 4 = 'this is y' z = 'This is z';
run;
```

In this LABEL statement, the x variable has the same label:

```
label x = this is x y 4 = this is y  z = This is z;
```

See

"Character Constants and Character Variables" in *SAS Programmer's Guide: Essentials*.

''

removes a label from a variable. Enclose a single blank space in quotation marks to remove an existing label. You can remove labels from multiple variables in a single LABEL statement:

```
data test;
   L=5; W=10;
```

```
   label L = ' '  W = ' ';
run;
```

Alternatively, you can remove a label by specifying nothing after the equal sign:

```
data test;
   L=5; W=10;
   label L =  W = ;
run;
```

See Also

"LABEL Statement" on page 150 for more information about the LABEL statement.

WHERE Statement

Subsets the input data set by specifying certain conditions that each observation must meet before it is available for processing.

Type: Procedure statement

Syntax

WHERE *where-expression*;

Required Argument

where-expression
is an arithmetic or logical expression that generally consists of a sequence of operands and operators. For more information about where processing, see *SAS DATA Step Statements: Reference*.

Tip You can specify multiple where-expressions.

See Also

"WHERE Statement" on page 151 for more information about the WHERE statement.

Examples: SGSCATTER Procedure

About the SGSCATTER Procedure Examples

Examples in This Section

This section provides basic examples of how to use the SGSCATTER procedure. The following examples are provided:

Example	Description
"Example 1: Creating a Scatter Plot Matrix" (p. 1770)	Shows how to use the MATRIX statement to create a grouped scatter plot matrix.
"Example 2: Creating a Graph with Multiple Independent Scatter Plots and Spline Curves" (p. 1771)	Shows how to use the PLOT statement and the PBSPLINE option to create a graph of independent scatter plots with a fitted, penalized B-spline curve overlaid on each.
"Example 3: Creating a Simple Comparative Panel" (p. 1773)	Shows how to use the COMPARE statement and the GROUP option to create a grouped comparative scatter plot.
"Example 4: Creating a Comparative Panel with Regression Fits and Confidence Ellipses" (p. 1774)	Shows how to use the COMPARE statement and the REG and ELLIPSE options to create a comparative panel of scatter plots with regression fits and confidence ellipses.

Additional Examples in This Book

For additional examples, see Table 12.148 on page 1700.

Examples Available Online

You can find more examples on the Graphically Speaking blog. On the blog page, use **Search Graphically Speaking** to search for SGSCATTER to find the SGSCATTER procedure examples.

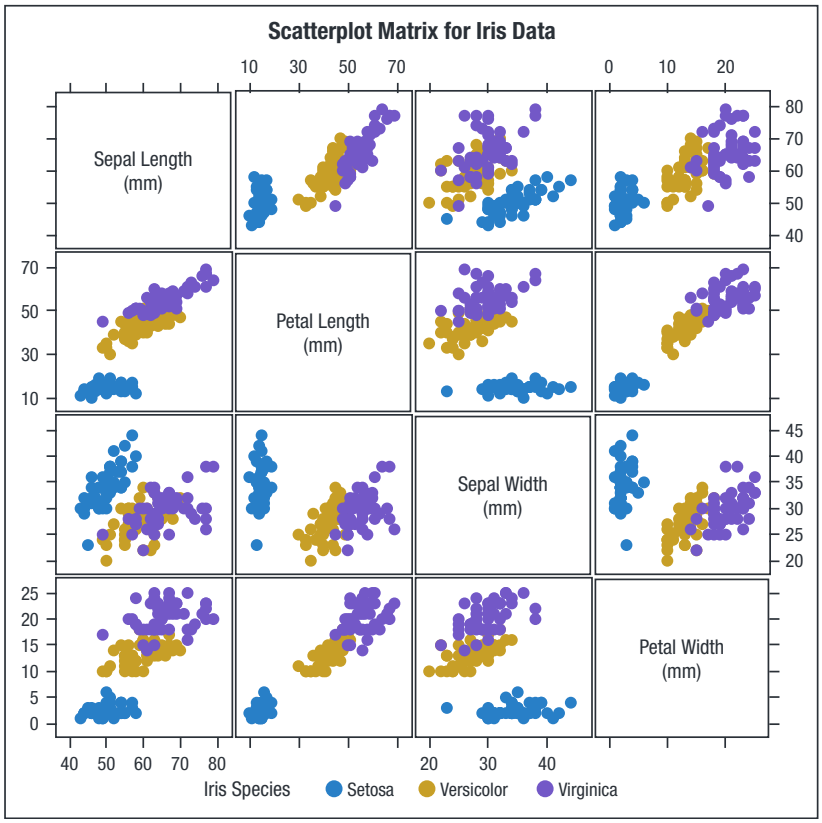
You can also find examples on support.sas.com at the following URL: https://support.sas.com/sassamples/graphgallery/PROC_SGSCATTER.html.

Example 1: Creating a Scatter Plot Matrix

Features:	MATRIX statement GROUP option
Sample library member:	SGSCMAT
Note:	For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a scatter plot matrix with grouped data.

Output



Program

Set the title and footnote and create the scatter plot matrix. In the MATRIX statement, the GROUP= on page 1738 option groups the data by the SPECIES variable.

```
proc sgscatter data=sashelp.iris;
  title "Scatterplot Matrix for Iris Data";
  matrix sepallength petallength sepalwidth petalwidth
         / group=species;
run;
title;
```

Example 2: Creating a Graph with Multiple Independent Scatter Plots and Spline Curves

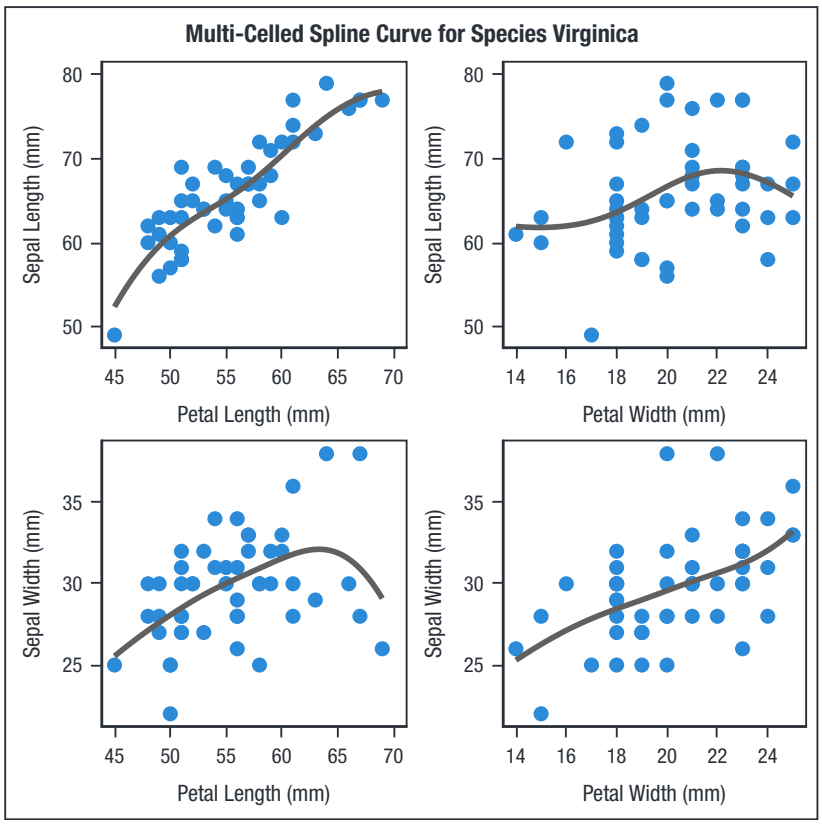
Features: PLOT statement
PBSPLINE option

Sample library member: SGSCPLT

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a graph with multiple independent scatter plots with fitted splines.

Output



Program

Create the scatter plots. In the PLOT statement, the PBSPLINE on page 1757 option fits the spline curves to the scatter points.

```
proc sgscatter data=sashelp.iris(where=(species="Virginica"));
  title "Multi-Celled Spline Curve for Species Virginica";
  plot (sepallength sepalwidth)*(petallength petalwidth)
       / pbspline;
run;
title;
```

Example 3: Creating a Simple Comparative Panel

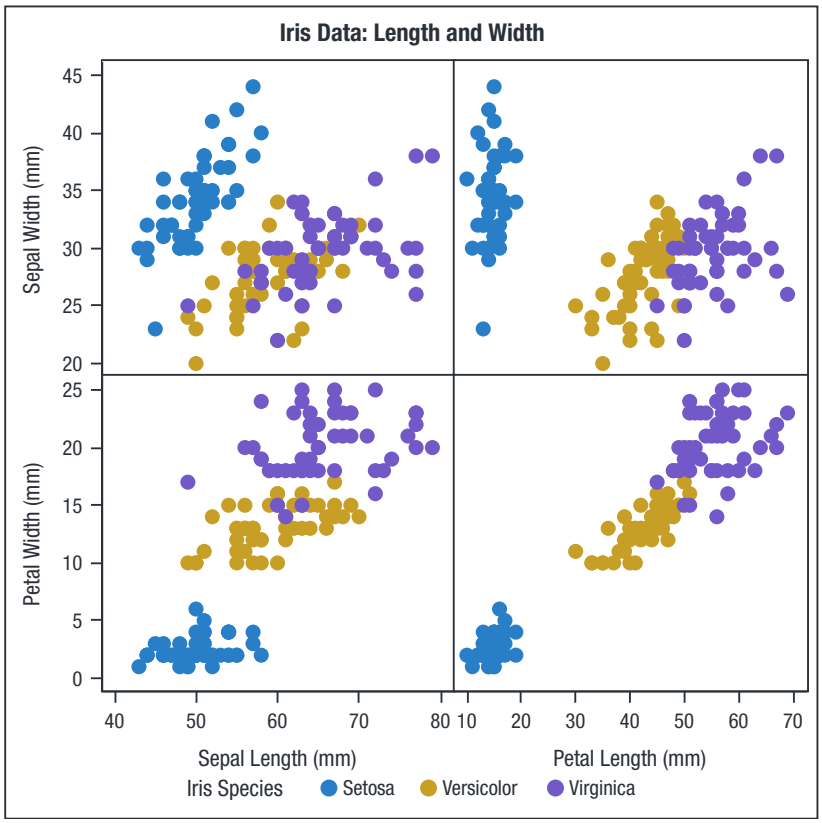
Features: COMPARE statement
GROUP option

Sample library member: SGSCCMP

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a comparative scatter plot with grouped data.

Output



Program

Create the scatter plot. In the COMPARE statement, the GROUP= on page 1721 option groups the data by the SPECIES variable.

```
proc sgscatter data=sashelp.iris;
  title "Iris Data: Length and Width";
  compare x=(sepallength petallength)
          y=(sepalwidth petalwidth)
          / group=species;
run;
title;
```

Example 4: Creating a Comparative Panel with Regression Fits and Confidence Ellipses

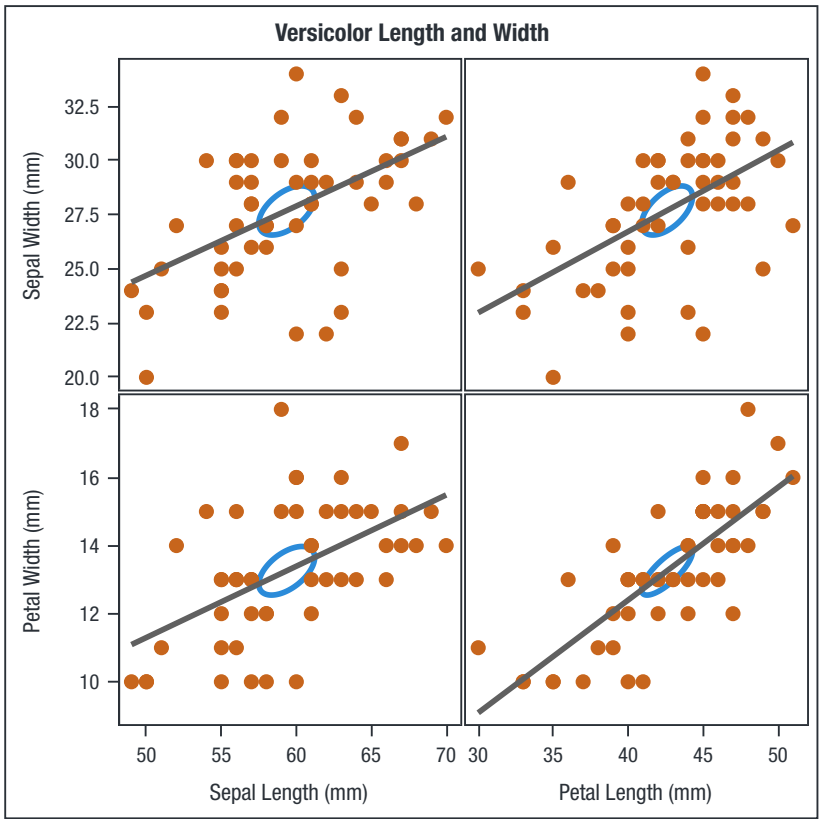
Features: COMPARE statement
REG option
ELLIPSE option
SPACING option

Sample library member: SGSCCM1

Note: For information about the SAS Sample Library, see "About the SASHELP Data Library" on page 88.

This example shows a comparative panel of scatter plots with regression fits and confidence ellipses.

Output



Program

Create the scatter plot. In the COMPARE statement, the REG on page 1729 option fits the regression lines and the ELLIPSE on page 1719 option creates the confidence ellipses and sets the ellipse type to MEAN.The SPACING= on page 1730 option adds spacing between plots.

```
proc sgscatter data=sashelp.iris(where=(species="Versicolor"));
  title "Versicolor Length and Width";
  compare y=(sepalwidth petalwidth)
          x=(sepallength petallength) /
          markerattrs=GraphData4
          reg ellipse=(type=mean) spacing=4;
run;
title;
```

13

Common Concepts

Overview of the Common Concepts ... **1778**

Plot Type Compatibility ... **1779**
- Applicable Procedures ... 1779
- Types of Plots You Can Create Using SGPLOT and SGPANEL ... 1779
- Plot Compatibility ... 1780
- What to Consider When Overlaying Plots ... 1784

Plot Axes ... **1784**
- Applicable Procedures ... 1784
- Axis Types ... 1785
- Axis Statements ... 1785
- Controlling the Axis Features ... 1786

Legends ... **1790**
- Types of Legends ... 1790
- Some of the Uses for a Legend ... 1791
- Using Discrete Legends ... 1791
- Using Gradient Color Legends ... 1793
- Using a Discrete Legend and a Gradient Color Legend ... 1796

Automatic Differentiation of Visual Attributes ... **1797**
- See Also ... 1801

Fit Policies for Axis Tick Values, Curve Labels, and Data Labels ... **1802**
- Overview of Collision Avoidance ... 1802
- Fit Policies for Axes ... 1803
- Fit Policies for Bar Charts ... 1804

Marker Fills and Outlines ... **1805**

Color-Naming Schemes ... **1808**
- Overview of Color-Naming Schemes ... 1808
- RGB Color Codes ... 1809
- CMYK Color Codes ... 1810
- HLS Color Codes ... 1811
- HSV (or HSB) Color Codes ... 1812
- Gray-Scale Color Codes ... 1813
- Color Naming System (CNS) Values ... 1814
- SAS Color Names and RGB Values in the SAS Registry ... 1815
- Converting Color Values between Color-Naming Schemes ... 1816
- Converting Numeric Color Component Values to Color Names ... 1816
- Color Support for Universal Printers ... 1818

Fonts 1819
About Fonts in ODS Graphics 1819
TrueType Fonts Supplied by SAS 1819
International Character Support 1821
Determining What Fonts Are Available 1823
Viewing Font Specifications in the SAS Registry 1823
Listing the Registered Fonts for a Universal Printer 1824
Registering Fonts 1825

Overview of the Common Concepts

These topics describe concepts that are similar among the SGPANEL, SGPLOT, and SGSCATTER procedures.

Table 13.1 *Common Concepts*

"Plot Type Compatibility" (p. 1779)	explains which types of plots can be used together in a graph
"Plot Axes" (p. 1784)	describes the axis types that are supported by the procedure
"Legends" (p. 1790)	explains how legends are created automatically, and how to create customized legends
"Automatic Differentiation of Visual Attributes" (p. 1797)	explains when different style attributes are automatically assigned to plots, and how to force the procedure to use different style attributes if they are not automatically assigned
"Fit Policies for Axis Tick Values, Curve Labels, and Data Labels" (p. 1802)	describes how to split the text for data labels, curve labels, and axis tick mark values when there is not enough room to display the text normally
"Marker Fills and Outlines" (p. 1805)	describes how you can change the appearance of both the marker fill and its outline for graphs that contain markers.
"Color-Naming Schemes" (p. 1808)	describes the color-naming schemes that are supported in ODS Graphics and how to specify colors in your programs.
"Fonts" (p. 1819)	describes how to determine the fonts that are available for your graphs, how to specify fonts in your programs, and how to register your own fonts in SAS.

Plot Type Compatibility

Applicable Procedures

The plots listed in this section apply to the SGPANEL and SGPLOT procedures only, unless otherwise noted.

Types of Plots You Can Create Using SGPLOT and SGPANEL

You can create the following types of plots with the SGPLOT and SGPANEL procedures:

- basic plots
- fit and confidence plots
- distribution plots
- categorization plots

Basic Plots

The plot statements that you can use in SGPLOT and SGPANEL procedure steps to create basic charts are as follows:

BAND	HEATMAP	SERIES
BUBBLE	HEATMAPPARM	SPLINE
BLOCK	HIGHLOW	STEP
DROPLINE	LINEPARM	TEXT
ELLIPSEPARM	NEEDLE	VBARBASIC*
FRINGE	POLYGON	VBARPARM
HBARBASIC*	REFLINE	VECTOR
HBARPARM	SCATTER	

* This statement generates a categorization plot that is compatible with the other basic plots.

Fit and Confidence Plots

The plot statements that you can use in SGPLOT and SGPANEL procedure steps to create fit and confidence plots are as follows:

ELLIPSE	PBSPLINE
LOESS	REG

Distribution Plots

The plot statements that you can use in SGPLOT and SGPANEL procedure steps to create distribution plots are as follows:

DENSITY	HISTOGRAM
HBOX	VBOX

Categorization Plots

The plot statements that you can use in SGPLOT and SGPANEL procedure steps to create categorization plots are as follows:

DOT	HLINE	VLINE
HBAR	VBAR	WATERFALL* **
HBARBASIC	VBARBASIC	

* This plot statement is available only with the SGPLOT procedure.

** This plot statement is not compatible with the other categorization plots.

Plot Compatibility

Not all of the plot types can be used together in the same PROC SGPLOT or PROC SGPANEL procedure step. If you submit a PROC SGPLOT or PROC SGPANEL procedure step that combines two incompatible plot statements, then an error is written to the log. To help you avoid plot overlay issues, the following table shows which of the plot types can be used together and any restrictions that apply.

Table 13.2 Plot Compatibility

Plot	Compatible Plot Types	Restrictions[1]
BAND	Basic Fit and Confidence Distribution	See "BAND Statement" on page 907.
BLOCK	Basic Fit and Confidence Distribution	
BUBBLE	Basic Fit and Confidence Distribution	
DENSITY	Distribution	Can overlay with DENSITY and HISTOGRAM plots only. See "DENSITY Statement" on page 946 for additional information.
DOT	Categorization	Can overlay with HBAR, HBARBASIC, and HLINE plots only. See "DOT Statement" on page 954 for additional information.
DROPLINE	Basic Fit and Confidence	See "DROPLINE Statement" on page 973.
ELLIPSE	Basic Fit and Confidence	See "ELLIPSE Statement" on page 979.
ELLIPSEPARM	Basic Fit and Confidence	See "ELLIPSEPARM Statement" on page 988.
FRINGE	Basic Fit and Confidence Distribution	Cannot overlay with HBOX and VBOX distribution plots.
HBAR	Categorization	Cannot overlay with VBAR, VBARBASIC, VLINE, and WATERFALL plots. See "HBAR Statement" on page 1008 for additional information.

Plot	Compatible Plot Types	Restrictions[1]
HBARBASIC	Basic Categorization	Cannot overlay with VBAR, VBARBASIC, VLINE, and WATERFALL categorization plots.
HBARPARM	Basic Distribution	Cannot overlay with VBARPARM basic plot and VBOX distribution plot. See "HBARPARM Statement" on page 1055 for additional information.
HBOX	Basic Distribution	Cannot overlay with NEEDLE and VBARPARM basic plots, VBOX distribution plot, and WATERFALL categorization plot.
HEATMAP	Basic Fit and Confidence	
HEATMAPPARM	Basic Fit and Confidence	
HIGHLOW	Basic Fit and Confidence Distribution	
HISTOGRAM	Distribution	Can overlay with DENSITY plot only.
HLINE	Categorization	Cannot overlay with VBAR, VBARBASIC, VLINE, and WATERFALL plots. See "HLINE Statement" on page 1145 for additional information.
LINEPARM	Basic Fit and Confidence	See "LINEPARM Statement" on page 1188.
LOESS	Basic Fit and Confidence	
NEEDLE	Basic Fit and Confidence Distribution	Cannot overlay with HBOX distribution plot. See "NEEDLE Statement" on page 1210 for additional information.
PBSPLINE	Basic Fit and Confidence	

Plot	Compatible Plot Types	Restrictions[1]
POLYGON	Basic Fit and Confidence Distribution	See "POLYGON Statement" on page 1238.
REFLINE	All plots (p. 1779)	
REG	Basic Fit and Confidence	
SCATTER	Basic Fit and Confidence Distribution	
SERIES	Basic Fit and Confidence	
SPLINE	Basic Fit and Confidence	
STEP	Basic Fit and Confidence Distribution	See "STEP Statement" on page 1343.
TEXT	Basic Fit and Confidence	
VBAR	Categorization	Cannot overlay with HBAR, HBARBASIC, HLINE, and WATERFALL plots. See "VBAR Statement" on page 1389 for additional information.
VBARBASIC	Basic Categorization	Cannot overlay with HBARPARM basic plot, and HBAR, HBARBASIC, HLINE, and WATERFALL categorization plots.
VBARPARM	Basic Distribution	Cannot overlay with HBARPARM basic plot and HBOX distribution plot. See "VBARPARM Statement" on page 1438 for additional information.
VBOX	Basic Distribution	Cannot overlay with HBARPARM basic plot, and HBOX distribution plot.

Plot	Compatible Plot Types	Restrictions[1]
VECTOR	Basic Fit and Confidence Distribution	
VLINE	Categorization	Cannot overlay with HBAR, HBARBASIC, HLINE, and WATERFALL plots. See "VLINE Statement" on page 1493 for additional information.
WATERFALL	Basic Distribution	Cannot overlay with HBOX distribution plot.

1 Additional restrictions might apply for certain plots. Where indicated, see the plot statement description for additional information.

What to Consider When Overlaying Plots

The procedure draws the plots in your graph in the same order that you specify the plot statements. Because of this, it is important to consider the order of your plot statements so that your plots do not obscure one another. For example, if you specify a BAND statement after a SCATTER statement, then the band plot might obscure the markers in your scatter plot. To correct this, you can reverse the order of the plot statements in your program. (You can also avoid obscuring your data by using the TRANSPARENCY= option to make your plots partially transparent.)

Plot Axes

Applicable Procedures

The information in this section applies to the SGPANEL and SGPLOT procedures only. The SGPANEL procedure generates panels of graphs that have axes for the rows and columns. The SGPLOT procedure generates single-cell graphs that have an X and Y axis.

Axis Types

By default, the type of each axis is determined by the types of plots that use the axis and the data that is applied to the axis.

The procedures support the following axis types:

Discrete
: The axis contains independent data values rather than a range of numeric values. Each distinct value is represented by a tick mark. Discrete is the default axis type for character data.

Linear
: The axis contains a linear range of numeric values. Linear is the default axis type for numeric data.

Logarithmic
: The axis contains a logarithmic range of values. The logarithmic axis type is not used as a default.

Time
: The axis contains a range of time values. Time is the default axis type for data that uses a SAS time, date, or datetime format.

Some types of plot do not support all of the axis types. For example, needle plots cannot use a discrete vertical axis. See the documentation for each plot statement to determine whether any axis type restrictions apply.

Axis Statements

The SGPANEL and SGPLOT procedures contain the following statements that enable you to control various features of the graph axes. The following table lists the axis statements.

Table 13.3 *Axis Statements in the SGPANEL and SGPLOT Procedures*

SGPANEL Procedure	SGPLOT Procedure
COLAXIS	XAXIS
ROWAXIS	X2AXIS
	YAXIS
	Y2AXIS

Controlling the Axis Features

Changing the Axis Type

Within any given layout in the graph display, each plot axis is always of a particular type. In the default cases, the axis type is always LINEAR, DISCRETE, or TIME.

The axis options for each layout statement include a TYPE= option that enables you to specify an axis type that overrides the default selection mechanisms. When you override the default axis type, you must be sure to specify the correct axis type for the plots that you are defining. For every plot in the template language, the documentation indicates what axis types it supports. Plots statements that are specified in the template are ignored if they are incompatible with the axis type.

Each axis type has features specific to that type, and the following axis options enable you to specify features for the different types:

`LINEAROPTS=(`*`linear-suboptions`*`)`

`DISCRETEOPTS=(`*`discrete-suboptions`*`)`

`TIMEOPTS=(`*`time-suboptions`*`)`

`LOGOPTS=(`*`log-suboptions`*`)`

One or more of these options can be specified for an axis, but the specified settings are applied only to the axis type that supports them.

For example, a bar chart has two axes – a TYPE=DISCRETE axis for the X axis and a TYPE=LINEAR axis for the Y axis. If a numeric column (for example, Age) is assigned to the X role, then this column's values are always treated as discrete values, never as a continuous range of values. You cannot request another axis type for the X axis, but you can request a different axis type for the Y axis.

Sometimes you want a specialized axis type depending on the nature of the data. For example, if the data have a very large range of values (orders of magnitude apart), then you could request that the values be displayed on a logarithmic scale. To set a logarithmic scale, use the TYPE=LOG axis option.

Time series data benefit from displaying the X axis with a TYPE=TIME axis. A TIME axis type requires that the column values are SAS Date, Time, or Datetime values.

Note: Certain plot types or layouts might impose restrictions on what type of axis can be assigned. The documentation for each plot identifies any restrictions that might apply to the axes.

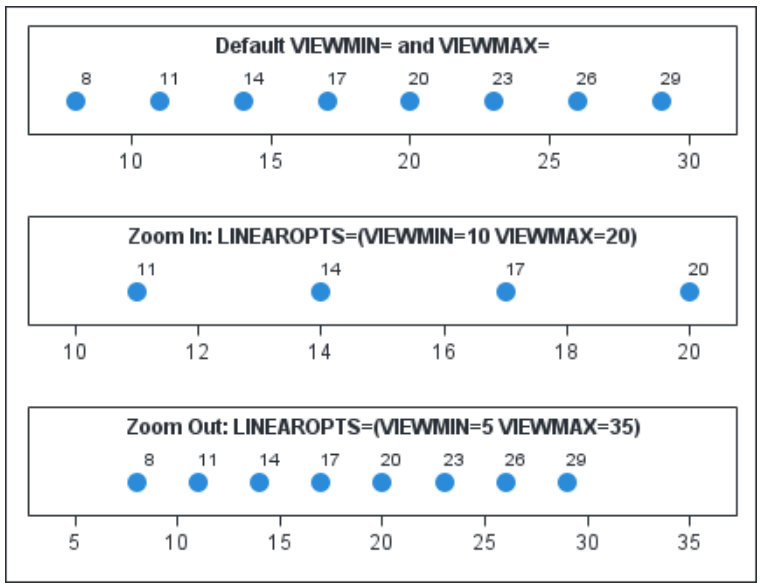
Adjusting the Axis View

The VIEWMIN= and VIEWMAX= axis options can be used to adjust the view of an axis. You can specify minimum data values to include in the display, maximum data values, or both (the specified values might be adjusted by the threshold calculation). By default, the VIEWMIN= value is the minimum data value for the specified axis and the VIEWMAX= value is the maximum data value for the specified axis.

A VIEWMIN= value that is greater than the data minimum or a VIEWMAX= value that is less than the data maximum acts like a “zoom in” operation. The adjusted view reduces the range of values represented on the axis and can sometimes exclude markers, lines, or fills that would normally appear.

A VIEWMIN= value that is less than the data minimum or a VIEWMAX= value that is greater than the data maximum acts like a “zoom out” operation. The adjusted view extends the range of values represented on the axis and sometimes compresses the markers, lines, or fills into a smaller area.

The following figure shows how the view settings can affect how the ticks and data values are displayed.



Adjusting Axis Thresholds

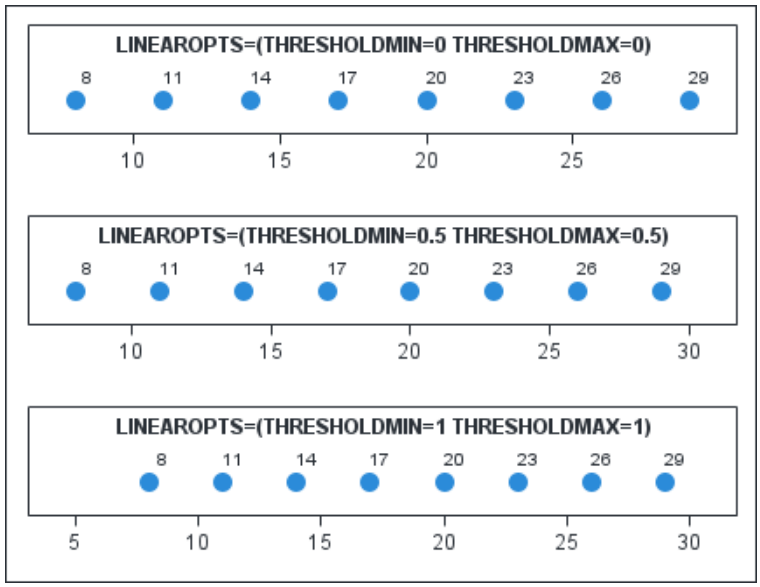
On a continuous, linear axis, the THRESHOLDMIN= and THRESHOLDMAX= axis options can be used to set a bias for including one more tick mark outside of either end of the data range (or VIEWMIN to VIEWMAX range). The threshold range is from 0 (do not include the tick mark) to 1 (include the tick mark). The default is 0.30. The bias at the minimum end of the axis is calculated using the

THRESHOLDMIN= value and the minimum data value (by default) or the VIEWMIN= value (if set).

The bias at the maximum end of the axis is calculated using the THRESHOLDMAX= value and the maximum data value (by default) or the VIEWMAX= value (if set).

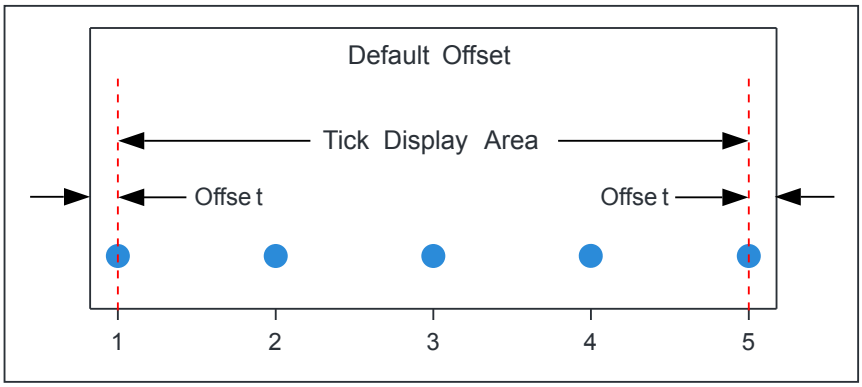
Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range. Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

The following figure shows how the threshold settings can affect the tick display on an axis. In the figure, 8 is the minimum value for the display and 29 is the maximum value.

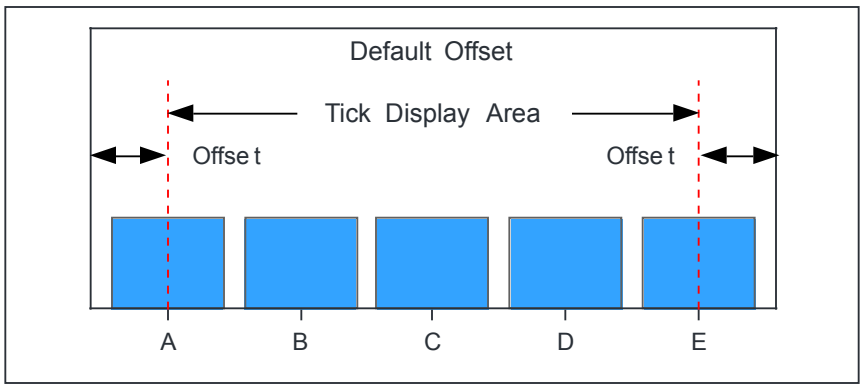


Adjusting Axis Offsets

In addition to axis thresholds, there are also axis offsets. Offsets are small gaps that are potentially added to each end of an axis (before the start of the data range and after the end of the data range). For plots that use markers such as scatter plots, axis offsets are automatically added to allow for markers to appear at the first or last tick without clipping the marker. On the minimum end of the axis, the offset is measured from the beginning of the axis to the midpoint of the first category. On the maximum end, the offset is measured from the end of the axis to the midpoint of the last category. The following figure shows an example of the default axis offsets for a scatter plot that uses an interval X axis.

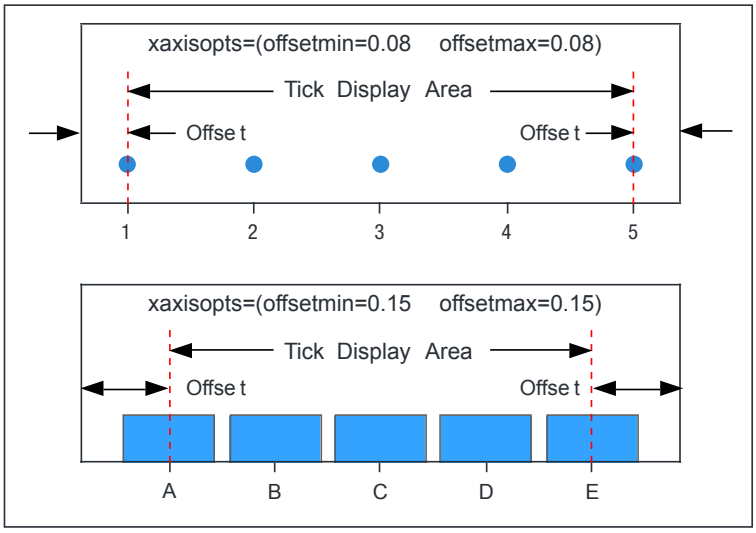


For plots such as box plots, histograms, and bar charts, offsets are added to ensure that the first and last box or bar is not clipped. The following figure shows an example of the default axis offsets for a bar chart.



You can use the OFFSETMIN= and OFFSETMAX= axis options in an axis statement to control the offsets on any type of axis. The OFFSETMIN= option controls the offset on the minimum end of the axis. The OFFSETMAX= option controls the offset on the maximum end of the axis. You can set both OFFSETMIN= and OFFSETMAX= to AUTO, AUTOCOMPRESS, or a numeric value. A numeric value specifies an offset as a proportion to the total length of the axis. A numeric offset value ranges from 0 to 1. However, if you specify a numeric value for both the minimum and maximum offsets, the sum of the offsets must not exceed 1.

The following figure shows axis offsets specified for markers and bars by using the OFFSETMIN= and OFFSETMAX= options.



Legends

Types of Legends

The procedures support two types of legends:

discrete legend
: contains one or more legend entries. Each entry consists of a graphical item (marker, line, ...) and corresponding text that explains the item.

  Here is a simple discrete legend.

gradient color legend
: maps the data range of a response variable to a range of colors.

  Here is a simple gradient legend.

Some of the Uses for a Legend

A graphical legend provides a key to the marker symbols, lines, and other data elements that are displayed in a graph. Here are some of the situations where legends are useful:

- when a plot contains grouped markers (scatter plots, for example).
- when a plot contains lines that differ by color, marker symbol, or line pattern (series plots or step plots, for example).
- when a plot contains one or more lines or bands that require identification or explanation.
- when series plots with different data are overlaid in the graph, or fit lines are displayed with confidence bands, or density plots with different distributions are generated.
- when markers vary in color to show the values of a response variable. For this case, you would generate a continuous color legend.

The procedures do not automatically generate legends for all of the above situations. However, the mechanism for creating legends is simple and flexible.

Using Discrete Legends

Applicable Procedures	SGPANEL
	SGPIE
	SGPLOT
	SGSCATTER

The SGPANEL and SGPLOT procedures create a legend automatically based on the plot statements and options that you specify. The automatic legend functionality determines which information is likely to be useful in the legend. For example, in non-group situations, if there are two or more plot overlays, a legend is generated automatically and the plots are added to the legend.

You can create customized legends by using one or more KEYLEGEND statements. You can use the KEYLEGEND statement to control the contents, title, position, and border of the legend. You can specify the labels that represent your plots in the legend by using the LEGENDLABEL= option in the corresponding plot statements.

TIP Using the LEGENDITEM statement, you can create a legend item that can be included in the legend. For more information, see "LEGENDITEM Statement" on page 1181.

You can override the automatic legend functionality in several ways. In the SGPANEL and SGPLOT procedures, legends are not generated automatically when you specify any of the following:

- KEYLEGEND statement that specifies the name of one or more plots
- NOAUTOLEGEND option in the procedure statement
- CURVELABEL= option for a plot
- a single plot or chart with no overlay and no group variable

In the PROC SGPIE statement, legends are not generated when the category data label is displayed or the NOAUTOLEGEND option is specified.

In the SGSCATTER procedure, you can prevent legends by using the NOLEGEND option in the PLOT, COMPARE, and MATRIX statements.

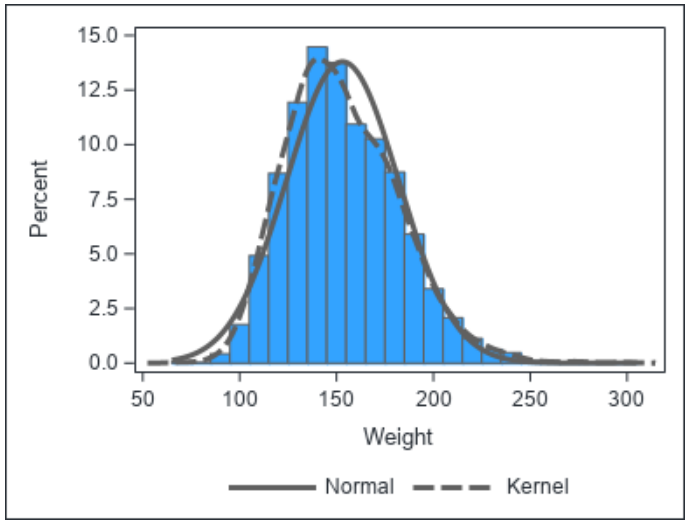
The following rules apply to the content of an auto-generated legend:

- If a group variable is used, the first statement using a group variable is added to the legend. All other statements are ignored.
- You can suppress features of a legend that is generated for a fit plot (LOESS, REG, PBSPLINE) using options such as NOLEGCLI, NOLEGCLM, and NOLEGFIT.

The SGSCATTER procedure creates a legend automatically when you specify a GROUP= variable. You can use the NOLEGEND option to disable the legend. For all of the graph creation statements, you can use the LEGEND= option to specify the attributes of the legend.

The following graph shows the default discrete legend that is generated automatically. In this example, the legend contains the normal and kernel density curves. You can use the KEYLEGEND statement to customize the legend.

Figure 13.1 *Automatically Generated Discrete Legend*




```
proc sgplot data=sashelp.heart;
    histogram weight;
    density weight;
       density weight / type=kernel
      lineattrs=(pattern=mediumdash);
```

```
run;
```

See Also

- “KEYLEGEND Statement” on page 430 (SGPANEL procedure)
- “KEYLEGEND Statement” on page 1170 (SGPLOT procedure)

Using Gradient Color Legends

Applicable Procedures	SGPANEL
	SGPLOT
	SGSCATTER

A color gradient legend maps the data range of a response variable to a range of colors. Gradient legends can be used with the following plot statements in the SGPLOT and SGPANEL procedures.

BUBBLE	HEATMAPPARM	TEXT
DOT	HIGHLOW	VBAR
HBAR	POLYGON	VBARBASIC
HBARBASIC	SCATTER	VBARPARM
HBARPARM	SERIES	VECTOR
HEATMAP	SPLINE	WATERFALL

Note:

- For heat maps, the gradient legend appears automatically.
- The WATERFALL chart applies to the SGPLOT procedure only.

In addition, gradient legends can be used with the PLOT, COMPARE, and MATRIX statements in the SGSCATTER procedure.

To generate the gradient legend, specify the COLORRESPONSE= option in any of those statements. The COLORRESPONSE= option indicates the numeric variable that is used to map colors to a continuous gradient. The COLORRESPONSE= option is used in conjunction with the GRADLEGEND feature.

The GRADLEGEND feature operates in different ways, depending on the procedure.

Table 13.4 GRADLEGEND Feature

Procedure	GRADLEGEND Feature
SGPANEL and SGPLOT	The procedures support one or more GRADLEGEND statements. The GRADLEGEND statement can associate the gradient legend with a plot statement and can specify options for the legend. For example, you can use the GRADLEGEND statement to control the title, position, tick values, and border of the legend.
SGSCATTER	The PLOT, COMPARE, and MATRIX statements support a GRADLEGEND= option that controls the title, position, and border of the legend.

The following rules apply to a color gradient legend:

- The COLORRESPONSE= option must be specified in order for the legend to appear.
- The GRADLEGEND feature is responsible for creating the legend.
- The GRADLEGEND feature has auto-legend capabilities. If you specify a COLORRESPONSE= option, a gradient legend automatically appears even if you do not specify the GRADLEGEND feature.

  Note: When the auto-legend feature is in effect, only one legend is displayed. This is true even if you specify the COLORRESPONSE= option in multiple plot statements. To have more than one gradient legend, specify the desired GRADLEGEND statements, and specify the correct plot name for each GRADLEGEND statement.

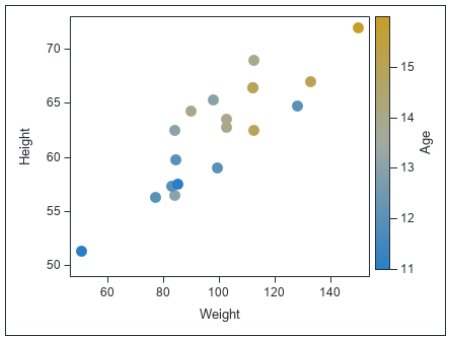
  For heat maps, the gradient legend appears automatically without specifying the COLORRESPONSE= option.

You can override the automatic legend functionality in two ways. Legends are not generated automatically when you specify either of the following:

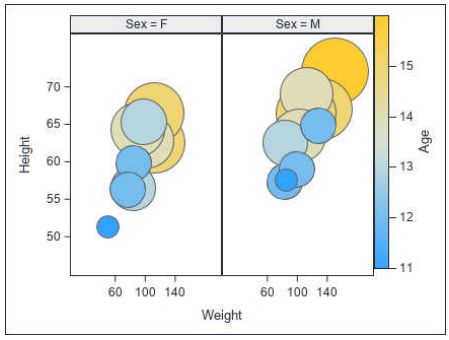
- KEYLEGEND statement in the SGPLOT and SGPANEL procedures
- NOGRADLEGEND option in the PLOT, COMPARE, or MATRIX statements in the SGSCATTER procedure

The following examples create an automatic legend in the SGPLOT, SGPANEL, and SGSCATTER procedures:

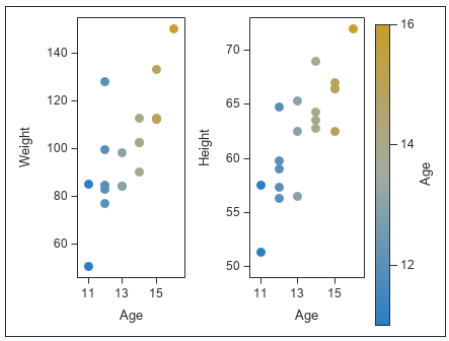
Table 13.5 *Examples of Automatic Gradient Legends*




```
proc sgplot data=sashelp.class;
  scatter x=weight y=height / colorresponse=age;
run;
```




```
proc sgpanel data=sashelp.class;
  panelby sex;
  bubble x=weight y=height size= age /
         colorresponse=age;
run;
```




```
proc sgscatter data=sashelp.class;
  plot (weight height) * age / colorresponse=age;
run;
```

proc sgscatter data=sashelp.class; plot (weight height) * age / colorresponse=age; run;

The COLORMODEL= option, available in the plot statements, enables you to control the appearance of the color ramp that is used.

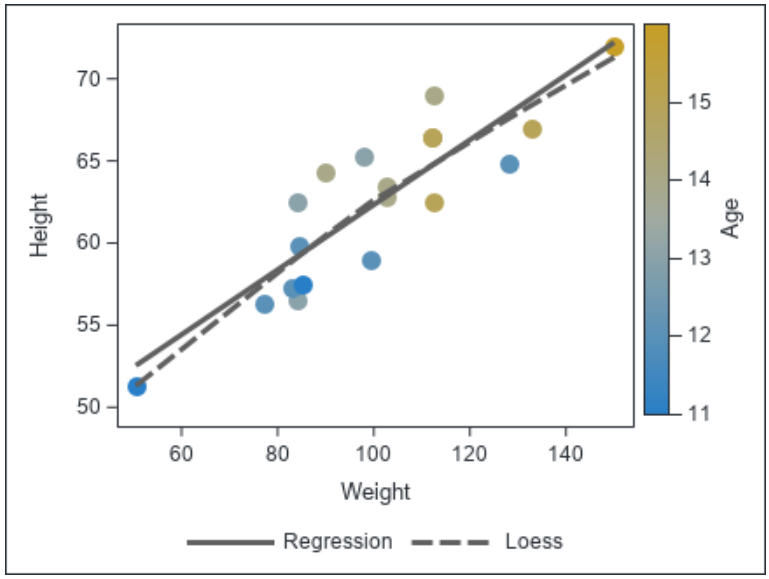
See Also

- "GRADLEGEND Statement" on page 272 (SGPANEL procedure)
- "GRADLEGEND Statement" on page 1003 (SGPLOT procedure)

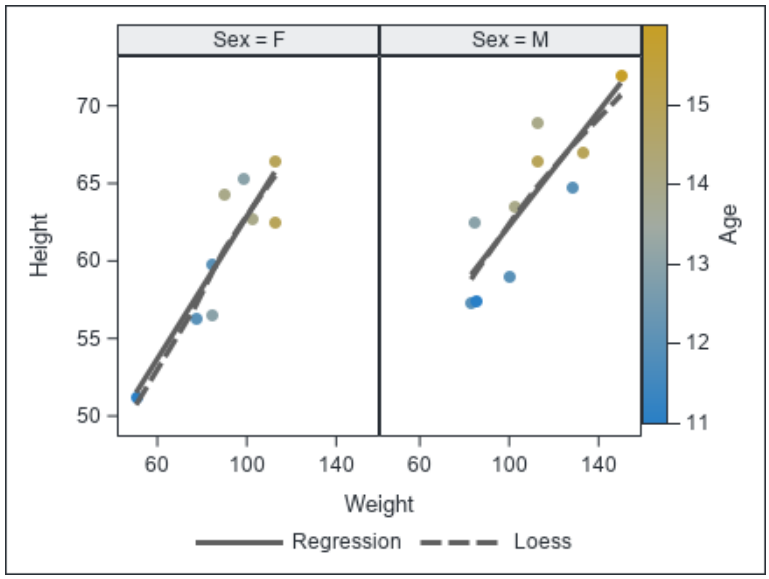
Using a Discrete Legend and a Gradient Color Legend

When a graph overlays plots that show discrete graphical elements and color gradients, both a discrete legend and a gradient legend can be displayed. The following examples create an automatic continuous color legend and discrete legend in the SGPLOT, SGPANEL, and SGSCATTER procedures:

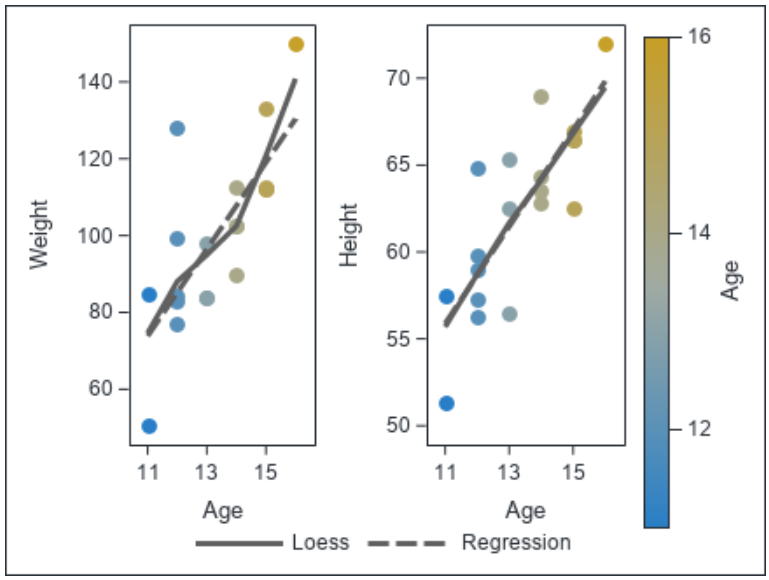
Table 13.6 *Examples of Automatic Gradient Legend and Discrete Legend*




```
proc sgplot data=sashelp.class;
   scatter x=weight y=height / colo
   reg x=weight y=height / nomarke
   loess x=weight y=height / nomarl
run;
```




```
proc sgpanel data=sashelp.class;
   panelby sex;
   scatter x=weight y=height / colo
   reg x=weight y=height / nomarke
   loess x=weight y=height / nomarl
run;
```




```
proc sgscatter data=sashelp.class;
  plot (weight height) * age / col
        loess reg;
run;
```

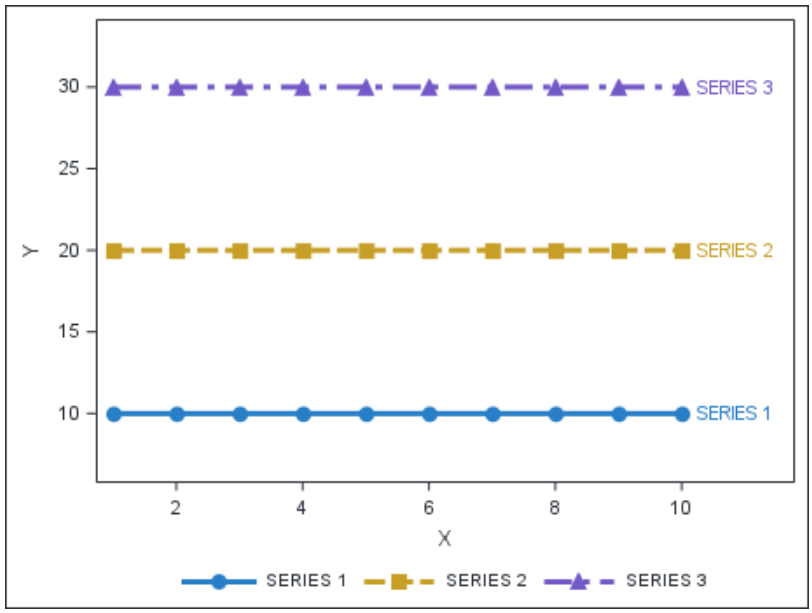
Note: If you want to use a KEYLEGEND statement or GRADLEGEND statement to customize one of the two legends, you must include both legend statements in the procedure step. If you provide only one of the legend statements, only that legend is rendered in the graph.

Automatic Differentiation of Visual Attributes

Applicable Procedures	SGPANEL SGPLOT SGSCATTER

For all three procedures, grouped data values, by default, use the GraphData*n* style elements for the presentation of each unique group value. When ungrouped data values are used in an overlay of multiple plots, by default, the SGPANEL and SGPLOT procedures assign different attributes to each plot when all of the independent plots in the overlay are of the same type (SERIES, SCATTER, and so on). In that case, each plot is sequentially assigned the next unused GraphData*n* style element. As a result, each series plot uses a different line and marker color. If ODS Graphics option ATTRPRIORITY=NONE is in effect, each series plot also uses a different line style and marker symbol. The following figure shows an example.

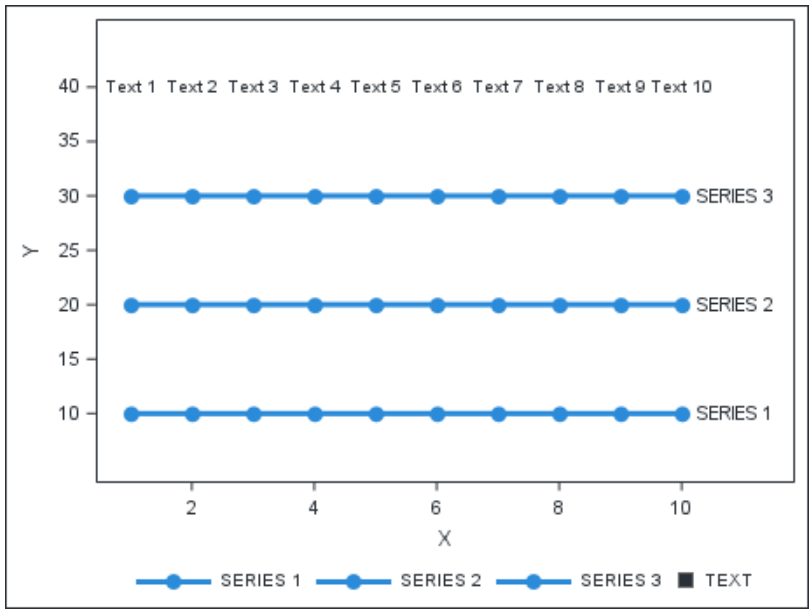
Figure 13.2 Overlay of Series Plots



In this example, the series plots are assigned a GraphData1–GraphData3 style element from the current style sequentially starting with plot SERIES 1 and GraphData1.

When the independent plots in the overlay are of different types, by default, each plot uses the attributes from the GraphDataDefault style element in the current style. The following figure shows an example.

Figure 13.3 Overlay of Series Plots and a Text Plot with No Attribute Cycling



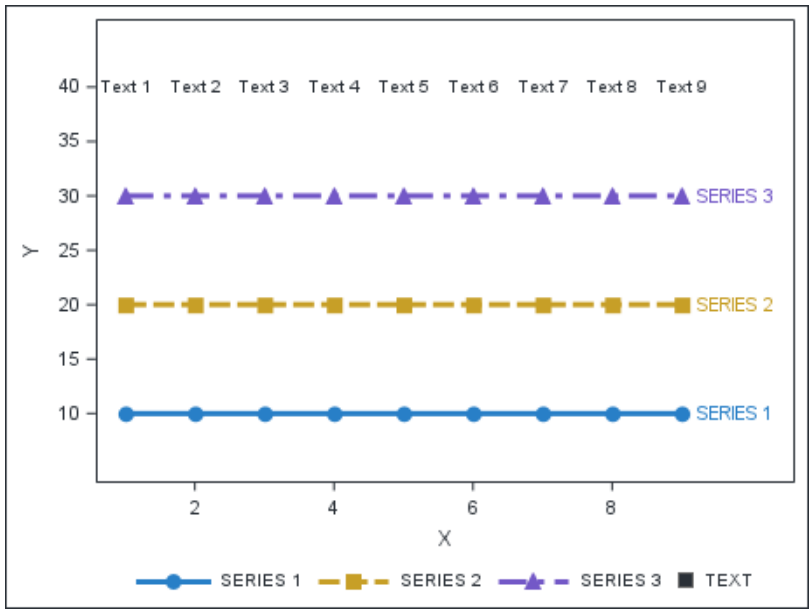
In this example, because the overlay contains a mix of SERIES and TEXT plots, all of the plots use the graph attributes from the GraphDataDefault style element in the current style.

You can use the CYCLEATTRS and NOCYCLEATTRS options in the PROC SGPANEL and PROC SGPLOT statements to override the default behavior. The CYCLEATTRS option forces the procedure to assign different attributes to each plot even when the overlay contains plots of different types. Conversely, the NOCYCLEATTRS option forces the procedure to use the default graph attributes even when all of the plots in the overlay are of the same type. The following example shows how to specify different attributes for each plot when the overlay contains plots of different types.

```
data testdata;
   input TXT $1-6 XVar YVar1 YVar2 YVar3 YVar4;
datalines;
Text 1 1 10 20 30 40
Text 2 2 10 20 30 40
Text 3 3 10 20 30 40
Text 4 4 10 20 30 40
Text 5 5 10 20 30 40
Text 6 6 10 20 30 40
Text 7 7 10 20 30 40
Text 8 8 10 20 30 40
Text 9 9 10 20 30 40
;
run;

ods graphics / attrpriority=none;

proc sgplot data=testdata cycleattrs;
   series x=XVar y=YVar1 / name="sp1" markers legendlabel="SERIES 1"
      curvelabel="SERIES 1";
   series x=XVar y=YVar2 / name="sp2" markers legendlabel="SERIES 2"
      curvelabel="SERIES 2";
   series x=XVar y=YVar3 / name="sp3" markers legendlabel="SERIES 3"
      curvelabel="SERIES 3";
   text x=XVar y=YVar4 text=TXT / name="tp" legendlabel="TEXT"
      textattrs=(size=8pt);
   keylegend "sp1" "sp2" "sp3" "tp";
   yaxis offsetmin=0.15 offsetmax=0.15 label="Y";
   xaxis label="X";
run;
```

Here is the result.

Figure 13.4 Overlay of Series Plots and a Text Plot with Attribute Cycling



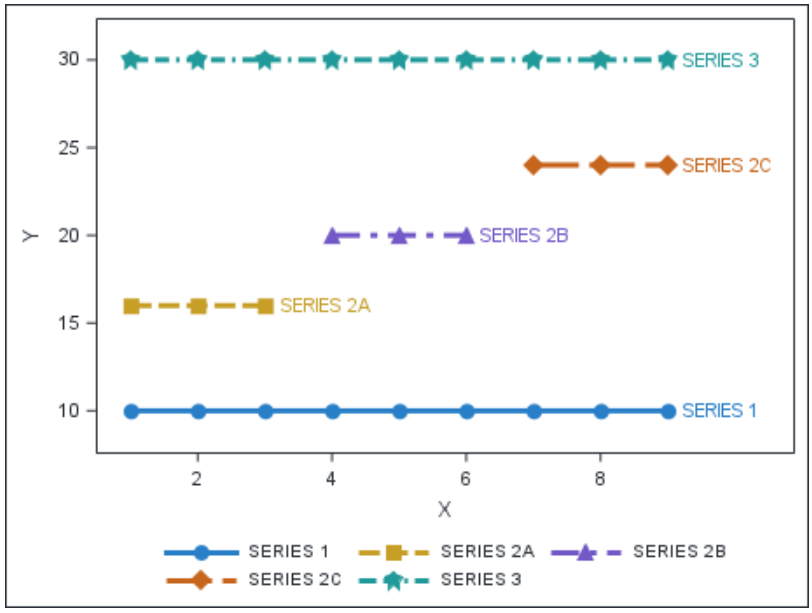
If one of the plots in the overlay uses grouped data, then the grouped plots also participate in the attribute cycle. For example, in Figure 13.4 on page 1800, if plot SERIES 2 is a grouped plot with three group values, then it generates three plots. In that case, the three SERIES 2 plots are assigned the GraphData2–GraphData4 style elements sequentially, and plot SERIES 3 is assigned GraphData5. Here is an example.

```
data testdatagrp;
   input TXT $1-6 GRP $8-16 XVar YVar1 YVar2 YVar3 YVar4;
datalines;
Text 1 SERIES 2A 1 10 16 30 40
Text 2 SERIES 2A 2 10 16 30 40
Text 3 SERIES 2A 3 10 16 30 40
Text 4 SERIES 2B 4 10 20 30 40
Text 5 SERIES 2B 5 10 20 30 40
Text 6 SERIES 2B 6 10 20 30 40
Text 7 SERIES 2C 7 10 24 30 40
Text 8 SERIES 2C 8 10 24 30 40
Text 9 SERIES 2C 9 10 24 30 40
;
run;

ods graphics / attrpriority=none;

proc sgplot data=testdatagrp;
   series x=XVar y=YVar1 / name="sp1" markers legendlabel="SERIES 1"
      curvelabel="SERIES 1";
   series x=XVar y=YVar2 / name="sp2" group=GRP markers curvelabel;
   series x=XVar y=YVar3 / name="sp3" markers legendlabel="SERIES 3"
      curvelabel="SERIES 3";
   keylegend "sp1" "sp2" "sp3";
   yaxis offsetmin=0.1 offsetmax=0.1 label="Y";
   xaxis offsetmin=0.05 offsetmax=0.18 label="X";
run;
```

```
title;
```

Here is the result.

Output 13.1 *Overlay of Grouped and Ungrouped Series Plots with Attribute Cycling*



For more information about grouped plots, see "Grouped Data" on page 1862.

For both grouped and ungrouped data, you can use the data attribute options DATACOLORS=, DATACONTRASTCOLORS=, DATAFILLPATTERNS=, DATALINEPATTERNS=, and DATASYMBOLS= in the STYLEATTRS statement to specify visual attributes directly. In that case, the data attribute options override the corresponding GraphData*n* style attributes for all of the plots. You can also specify appearance options in a plot statement to specify visual attributes directly for a specific plot. Appearance options in a plot statement override the corresponding GraphData*n* style element attributes and the data attribute options for that plot only.

See Also

- "CYCLEATTRS|NOCYCLEATTRS" on page 163 (SGPANEL procedure)
- "CYCLEATTRS|NOCYCLEATTRS" on page 891 (SGPLOT procedure)

Fit Policies for Axis Tick Values, Curve Labels, and Data Labels

Overview of Collision Avoidance

Applicable Procedures	SGPANEL SGPLOT SGSCATTER

Plot statements that support data labels or curve labels includes the ability to split the text for those labels when there is not enough room to display the text normally. Similarly, you can split the text for axis tick mark values. The text is split into two or more lines.

The following table lists the tasks that you can perform along with the option that is used to perform each task:

Table 13.7 *Tasks and Related Options for Collision Avoidance*

Task	Option
split the text at one or more characters that you specify	SPLITCHAR=
specify that the split characters are included in the displayed value	SPLITCHARNODROP
align the text horizontally (center, left, or right)	SPLITJUSTIFY=

These options are available for most of the plot statements in the SGPANEL and SGPLOT procedures. In the SGSCATTER procedure, the COMPARE, MATRIX, and PLOT statements support the options.

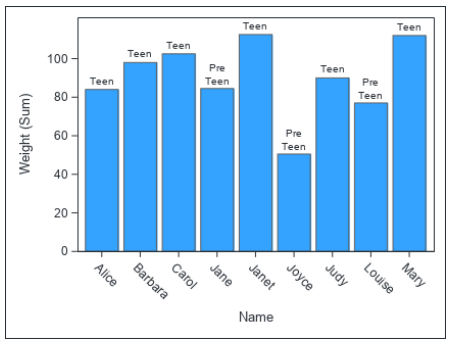
In the following example, the DATA step creates a column named SPEC with two values: **Teen** and **Pre-Teen**.

```
data class;
   set sashelp.class;
   where sex="F";
   length label $10;
   if age > 12 then label='Teen';
   else label='Pre-Teen';
```

```
run;
```

In this bar chart example, the procedure splits **Pre-Teen** at the hyphen (-) whenever the text does not fit within the bar width.

Table 13.8 *Character Split Example*




```
ods graphics / width=4.5in;
proc sgplot data=class;
    vbar name  / response=weight
      datalabel=label splitchar="-";
run;
```

Notice that the split character does not appear in the labels. To force the split character to appear, specify the SPLITCHARNODROP option.

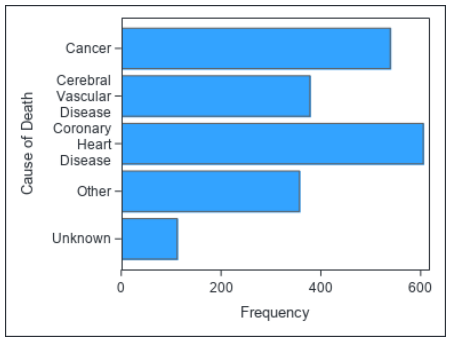
Fit Policies for Axes

For axes in the SGPANEL and SGPLOT procedures, you can specify split options for the fit policy. The FITPOLICY= option specifies the method that is used to fit tick mark values on the axis when there is not enough room to draw them normally.

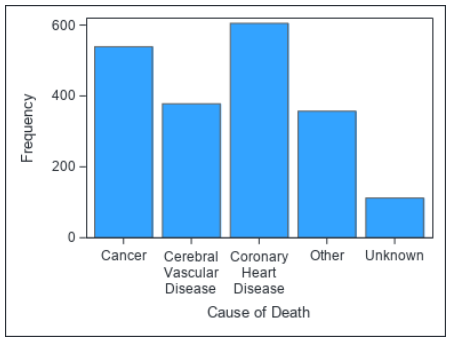
The FITPOLICY= option supports a number of values that split, stagger, and rotate the values. FITPOLICY= also combines these options. For example, FITPOLICY=SPLITROTATE first attempts to use SPLIT and then ROTATE to fit the values.

The following examples use the fit policy to split axis values. The default split character is a space. You can override the default and specify the split character using the SPLITCHAR= option (not shown in the example). All three of these examples use the SPLITALWAYS fit policy. You might instead specify FITPOLICY=SPLIT if you want splitting to occur only on collision.

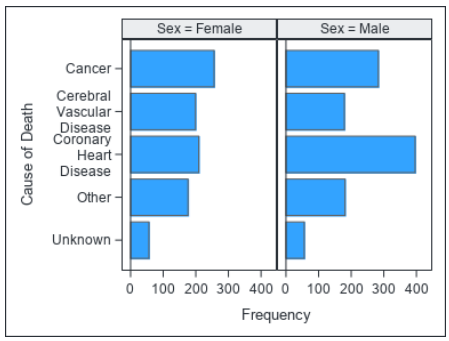
Table 13.9 *Examples That Show Split Axis Values*




```
ods graphics / width=4in;
proc sgplot data=sashelp.heart;
  hbar deathcause;
  yaxis fitpolicy=splitalways;
run;
ods graphics / reset=width;
```




```
ods graphics / width=4in;
proc sgplot data=sashelp.heart;
  vbar deathcause;
  xaxis fitpolicy=splitalways;
run;
ods graphics / reset=width;
```




```
ods graphics / width=4in;
proc sgpanel data=sashelp.heart;
  panelby sex;
  hbar deathcause;
  rowaxis fitpolicy=splitalways;
run;
ods graphics / reset=width;
```

Fit Policies for Bar Charts

In vertical bar charts, fit policies are available for data labels that appear above the bars. The DATALABELFITPOLICY= option specifies the method that is used to fit the data labels when the bars are not wide enough to display the labels normally.

The DATALABELFITPOLICY= option specifies whether the labels are rotated or split, or whether they are allowed to collide.

Marker Fills and Outlines

Applicable Procedures	SGPANEL SGPLOT SGSCATTER

When your graph contains markers, such as those found in scatter plots, the MARKERATTRS= option enables you to specify the marker color, size, and symbol. This option is also available for plots, such as step plots, that support the creation of markers.

There might be times when you want to change the appearance of the marker fill and its outline separately. The following additional options enable you to specify attributes for the fill and outline of your markers separately:

- FILLEDOUTLINEDMARKERS specifies that markers have a fill and an outline.
- MARKERFILLATTRS= specifies the color of the marker fill.
- MARKEROUTLINEATTRS= specifies the color and thickness of the marker outline.

These options are available for the dot and scatter plots, series and step plots, line and needle plots, and fit plots in the SGPANEL and SGPLOT procedures. They are also available for the PLOT and COMPARE statements in the SGSCATTER procedure.

The options are typically used along with the MARKERATTRS= option and enable markers to be more customized. The options have a cumulative, overriding effect when used with MARKERATTRS=.

To specify attributes for marker fills and outlines, do the following:

1 Specify FILLEDOUTLINEDMARKERS in the plot statement.

   For PROC SGSCATTER, specify FILLEDOUTLINEDMARKERS in the PLOT or COMPARE statement.

2 Make sure the marker uses a filled symbol. The marker symbol, derived either from the applied style or specified with the MARKERATTRS= option, must have the “Filled” suffix in its name. For example, the marker might be CircleFilled, DiamondFilled, TriangleFilled, and so on.

   See marker symbols on page 1834.

If the above two conditions are met, then you can use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options.

When FILLEDOUTLINEDMARKERS is specified, the marker attributes are obtained as follows:

- MARKERATTRS= provides the marker size and symbol. This option does not provide the marker fill color.
- MARKERFILLATTRS= provides the marker fill color.
- MARKEROUTLINEATTRS= provides the color and thickness of the marker outline.

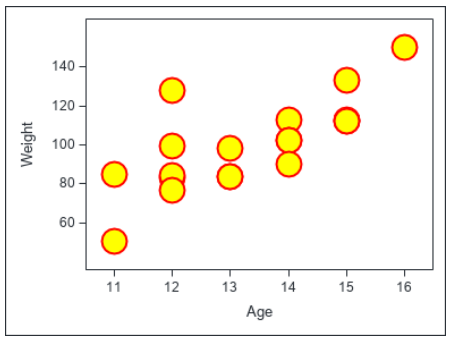
The following table shows this information at a glance:

Table 13.10 *Marker Attributes That Are Affected by the Marker Options When FILLEDOUTLINEDMARKERS Is Specified*

Option	Color	Size	Symbol	Outline Color	Outline Thickness
MARKERATTRS		X	X		
MARKERFILLATTRS	X				
MARKEROUTLINEATTRS				X	X

The following example shows how the three options work together to produce a scatter plot with yellow markers and red outlines.

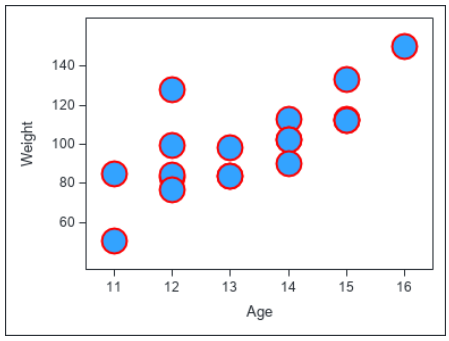
Table 13.11 *Yellow Markers with Red Outlines*




```
proc sgplot data=sashelp.class;
scatter x=age y=weight / filledoutlinedmarkers
   markerfillattrs=(color=yellow)
   markeroutlineattrs=(color=red thickness=2)
   markerattrs=(symbol=circlefilled size=25 );
run;
```

In the following example, the MARKERFILLATTRS= option is not specified. Because this option determines marker color when FILLEDOUTLINEDMARKERS is used, the marker color becomes the default color for the style (Illuminate). (The COLOR attribute in the MARKERATTRS= option is ignored.)

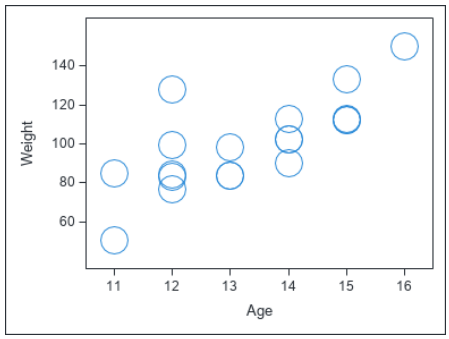
Table 13.12 *Marker Color Uses the Default Value*




```
proc sgplot data=sashelp.class;
scatter x=age y=weight / filledoutlinedmarkers
   markeroutlineattrs=(color=red thickness=2)
   markerattrs=(symbol=circlefilled size=25
          color=yellow);
run;
```

In the following example, the markers are specified as CIRCLE, which is not filled. As a result, the options used to specify the fill and outline attributes are ignored.

Table 13.13 *Markers Not Filled, Defaults Used*




```
proc sgplot data=sashelp.class;
scatter x=age y=weight / filledoutlinedmarkers
   markerfillattrs=(color=yellow)
   markeroutlineattrs=(color=red thickness=2)
   markerattrs=(symbol=circle size=25);
run;
```

Note: The following code produces the same result as the previous example. In the following code, the SYMBOL= option is not specified, and the default marker symbol is used. Because the graph uses the Illuminate ODS style, the default marker symbol is CIRCLEFILLED.

```
proc sgplot data=sashelp.class;
scatter x=age y=weight / filledoutlinedmarkers
   markerfillattrs=(color=yellow)
   markeroutlineattrs=(color=red thickness=2)
   markerattrs=(size=25);
run;
```

Color-Naming Schemes

Overview of Color-Naming Schemes

The valid color-naming schemes in SAS are as follows:

- RGB (red green blue)
- CMYK (cyan magenta yellow black)
- HLS (hue lightness saturation)
- HSV (hue saturation brightness), also called HSB
- Gray scale
- SAS color names (from the SAS Registry)
- SAS Color Naming System (CNS)

Table 13.14 on page 1808 shows examples of each color-naming scheme.

Table 13.14 *Examples of SAS Color-Naming Schemes*

Color-Naming Scheme	Example
RGB	`COLOR=(CX98FB98)`

CMYK	`COLOR=(FF00FF00)`
	`COLOR=("00FFFF00")`
	`COLOR=(CMYK00FFFF00)`

HLS	`COLOR=(H14055FF)`
HSV	`COLOR=(V0F055FF)`
Gray Scale	`COLOR=(GRAY4F)`
SAS Registry Colors	`COLOR=(palegreen)`

CNS Color Names	`COLOR=("very light purplish blue")`
	`COLOR=(VeryLightPurplishBlue)`
	`COLOR=(Very_Light_Purplish_Blue)`

You can also mix color-naming schemes in the same statement. For example:

```
DATACOLORS=(CXEE0044 "vivid blue" darkgreen);
```

Each of the color-naming schemes has its advantages and disadvantages based on how the output is used. For example, if you are creating a report that will be viewed only online, then specifying colors using the RGB naming scheme or the SAS color names defined in the registry might produce better results. If you are creating a report for publishing in printed form, you might want to use the CMYK color-naming scheme.

Note: If you specify an invalid color name, the default color is used instead.

For additional information about color-naming schemes, see *Effective Color Displays: Theory and Practice* by David Travis, *Computer Graphics: Principles and Practice* by Foley, van Dam, Feiner, and Hughes, and http://colorbrewer2.org.

RGB Color Codes

The RGB color-naming scheme is usually used to define colors for a display screen. This color-naming scheme is based on the properties of light. An RGB color code defines a color by combining red, green, and blue colors in different ratios. All the colors combined together create white. The absence of all color creates black.

Color names are in the form CX*rrggbb*, where the following is true:

- CX indicates to SAS that this is an RGB color specification.
- *rr* is the red component.
- *gg* is the green component.
- *bb* is the blue component.

The components are hexadecimal numbers in the range 00–FF (0% to 100%). Each hexadecimal number indicates how much of the red, green, or blue is included in the color. Lower percentage values are darker and higher values are lighter. This scheme allows for up to 256 levels of each color component (more than 16 million different colors).

Table 13.15 *Examples of RGB Color Values*

Color	RGB Value
Red	CXFF0000
Green	CX00FF00
Blue	CX0000FF
White	CXFFFFFF
Black	CX000000

Any combination of the color components is valid. Some combinations match the colors produced by predefined SAS color names.

CMYK Color Codes

The CMYK color-naming scheme is used in four-color printing. CMYK is based on the principles of objects reflecting light. Combining equal values of cyan, magenta, and yellow produces process black, which might not appear as pure black. The black component (K) of CMYK can be used to specify the level of blackness in the output. A lack of all colors produces white when the output is printed on white paper.

To specify the colors from a printer's Pantone Color Lookup Table, you can use the CMYK color-naming scheme. Specify colors in terms of their cyan, magenta, yellow, and black components. Color names are in the form CMYK*ccmmyykk* or K*ccmmyykk*, where the following is true:

- CMYK or K is an optional prefix that indicates to SAS that this is a CMYK color specification.
- *cc* is the cyan component.
- *mm* is the magenta component.
- *yy* is the yellow component.
- *kk* is the black component.

The color-value components are hexadecimal numbers in the range 00–FF, where higher values are darker and lower values are brighter. This scheme allows for up to 256 levels of each color component. When the color value begins with a letter (A–F), you can omit the CMYK or K prefix. When the color value begins with a number (0–9), you must use the CMYK or K prefix, or enclose the value in quotation marks.

Note: For some graph style elements such as GraphData1–GraphDataN, the color specification must begin with a letter. Values that begin with a number such as "000000FF" are not valid. In those cases, use the CMYK or K prefix.

Table 13.16 *Examples of CMYK Color Values*

Color	CMYK Value
Red	"00FFFF00"[1]
Green	FF00FF00
Blue	CMYKFFFF0000
White	"00000000"[1]

Color	CMYK Value
Process black (using cyan, magenta, and yellow ink)	FFFFFF00
Pure black (using only black ink)	"000000FF"[1]

1 Not valid for some graph style elements.

Note: You can specify a CMY value by specifying zero (00) for *kk*, the color's black component.

CMYK color specifications are for output devices that support four colors. If you specify CMYK colors on an output device that supports three colors such as RGB, the CMYK colors are converted to colors in the three-color space. Because the four-color space supports many more colors than a three-color space, the CMYK colors might map to different colors in the three-color space. To preserve your CMYK colors in that case, specify a device that supports the CMYK color space. Some of the SAS universal printers support CMYK colors. See "Color Support for Universal Printers" on page 1818.

HLS Color Codes

The HLS color scheme specifies colors in terms of hue, lightness, and saturation levels. It is based on the Tektronix Color Standard. HLS color names are of the form H*hhhlss*, where the following is true:

- H indicates to SAS that this is an HLS color specification.
- *hhh* is the hue component, which is expressed as an angle.
- *ll* is the lightness component.
- *ss* is the saturation component.

The components are hexadecimal numbers. The hue component is in the range 000–168 hexadecimal, which represents an angular value in the range 0–360 decimal. Hue starts with the primary color blue at 0 degrees, and then progresses to red at 120 degrees, to green at 240 degrees, and back to blue at 360 degrees. Both the lightness and saturation components are in the range 00–FF hexadecimal (0–255 decimal), which provides 256 levels that represent 0% to 100% for each component.

Table 13.17 *Examples of HLS Color Codes*

Color	HLS Color Code
Red	H07880FF

Color	HLS Color Code
Green	H0F080FF
Blue	H00080FF
Light gray	H000BB00
White[1]	H*xxx*FF00, such as H000FF00
Black[1]	H*xxx*0000 such as H0000000

1 When the saturation is set to 00, the color is a shade of gray that is determined by the lightness value. Therefore, white is defined as H*xxx*FF00 and black as H*xxx*0000, where *xxx* can be any hue.

HSV (or HSB) Color Codes

The HSV color-naming scheme specifies colors in terms of hue, saturation, and value (or brightness) components. HSV color names are of the form V*hhhssvv*, where the following is true:

- V indicates to SAS that this is an HSV color specification.
- *hhh* is the hue component, which is expressed as an angle.
- *ss* is the saturation component.
- *vv* is the value or brightness component.

The components are hexadecimal numbers. The hue component is in the range 000–168 hexadecimal, which represents an angular value in the range 0–360 decimal. Hue starts with the primary color red at 0 degrees, and then progresses to green at 120 degrees, to blue at 240 degrees, and back to red at 360 degrees. Both the saturation and value (brightness) components are in the range 00–FF hexadecimal (0–255 decimal), which provides 256 levels for each component.

Table 13.18 *Examples of HSV (or HSB) Color Codes*

Color	HSV Color Code
Red	V000FFFF
Green	V078FFFF
Blue	V0F0FFFF
Light gray[1]	V*xxx*00BB such as V07900BB
White[1]	V*xxx*00FF such as V07900FF

Color	HSV Color Code
Black[1]	V*xxx*00000 such as V0790000

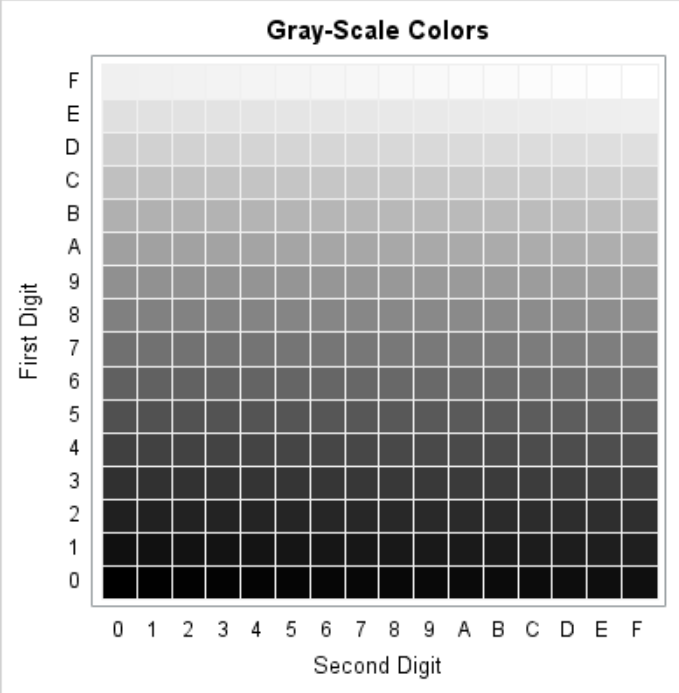
1 When the saturation is set to 00, the color is a shade of gray. The value component determines the intensity of gray level. The *xxx* value can be any hue.

Gray-Scale Color Codes

Gray-scale colors are specified using the word GRAY and a lightness value in the form GRAY*hh*. The value *hh* is the lightness of the gray and is a hexadecimal number in the range 00–FF, which provides 256 levels on the gray scale.

Note: GRAY, without a lightness value, is a SAS color name defined in the SAS registry. (See "SAS Color Names and RGB Values in the SAS Registry" on page 1815.) Its value is CX808080. Invalid color specifications are mapped to GRAY.

The following figure shows the gray-scale color for each hexadecimal value.



Color Naming System (CNS) Values

For CNS, you specify a color value by specifying lightness, saturation, and hue, in that order, using the terms shown in the following table.

Table 13.19 *Color Naming System Values*

Lightness	Saturation	Hue
Black	Gray	Blue
Very Dark	Grayish	Purple
Dark	Moderate	Red
Medium	Strong	Orange/Brown
Light	Vivid	Yellow
Very Light		Green
White		

Follow these rules when you are determining the CNS color name:

- You should not use the lightness values Black and White with saturation or hue values.
- If you do not specify default values, medium is the default lightness value and vivid is the default saturation value.
- Gray is the only saturation value that can be used without a hue.
- Unless the color that you want is black, white, or some form of gray, you must specify at least one hue.

You can use one or two hue values in the CNS color name. When you use two hue values, the hues must be adjacent to each other in the following list: blue, purple, red, orange/brown, yellow, green, and then returning to blue. Two hue values result in a color that is a combination of both colors. Use the suffix “ish” to reduce the effect of a hue when two hues are combined. Reddish purple is less red than red purple. The color specified with the “ish” suffix must precede the color without the “ish” suffix.

You can write color names in the following ways:

- without space separators between words
- with an underscore to separate words
- with a space to separate words, enclosed in quotation marks

For example, the following color specifications are valid:

- verylightmoderatepurplishblue
- very_light_moderate_purplish_blue
- "very light moderate purplish blue"

Note: If a CNS color name is also a color name in the SAS Registry, the SAS Registry color value takes precedence. Some CNS color names and color names in the SAS Registry have different color values. Dark blue is an example.

To use a CNS color value when the color name is also in the SAS Registry, do one of the following:

- Include a space to separate the words and enclose the entire color name in quotation marks. Here is an example:

```
color="dark blue"
```

- Include an underscore to separate the words. Here is an example:

```
color=dark_blue
```

SAS Color Names and RGB Values in the SAS Registry

The SAS Registry provides a set of color names and RGB values that you can use to specify colors. These color names and RGB values are common to most web browsers. You can specify the name itself or the RGB value associated with that color name. To view the color names as associated RGB values that are defined in the registry, submit the following code:

```
proc registry list
 startat="COLORNAMES";
run;
```

SAS prints the output in the SAS log. Here is a partial listing.

```
NOTE: Contents of SASHELP REGISTRY starting at subkey [COLORNAMES]
[  COLORNAMES]
  Active="HTML"
[    HTML]
    AliceBlue=hex: F0,F8,FF
    AntiqueWhite=hex: FA,EB,D7
    Aqua=hex: 00,FF,FF
    Aquamarine=hex: 7F,FD,D4
    Azure=hex: F0,FF,FF
    Beige=hex: F5,F5,DC
    ...
```

You can also create your own color values by adding them to the SAS registry. .For more information, see "REGISTRY Procedure" in *Base SAS Procedures Guide*.

Converting Color Values between Color-Naming Schemes

If the SAS/GRAPH software is included in your SAS installation, several macros are provided with SAS/GRAPH that enable you to perform selected conversions between color-naming schemes. The following table shows the conversions that are possible using these macros.

From Color Value	To Color Value	Conversion Macro
CMY	RGB	%CMY
CNS	HLS	%CNS
RGB	HLS	%RGB2HLS
HLS	RGB	%HLS2RGB

For information about these macros, see *SAS/GRAPH: Reference*.

Converting Numeric Color Component Values to Color Names

If the SAS/GRAPH software is included in your SAS installation, several macros are provided with SAS/GRAPH that enable you to convert numeric color component values to color names. The following table shows the macros that you can use for this purpose.

From Numeric Color Component Values	To Color Name	Conversion Macro
cyan, magenta, yellow, black	CMYK	%CMYK
hue, lightness, saturation	HLS	%HLS
hue, saturation, value	HSV	%HSV
red, green, blue	RGB	%RGB

The following example shows how to use the %HSV color utility macro to convert HSV color component values 357, 22, and 90, to an HSV color name.

```
%COLORMAC;
data _null_;
  put "%HSV(357,22,90)";
run;
```

Note: The %COLORMAC macro compiles all of the SAS/GRAPH color utility macros. You need to run it only once during a SAS session.

The result is V16538E6. The following example shows how to use the %RGB color utility macro to convert RGB numeric color component values 231, 179, and 180 to an RGB color name.

```
/* Compute the RGB percentages */
data _null_;
  r = 231;
  g = 179;
  b = 180;
  call symputx("r", round((r/255)*100));
  call symputx("g", round((g/255)*100));
  call symputx("b", round((b/255)*100));
run;

/* Convert to RGB color name */
%COLORMAC;
data _null_;
  put "%RGB(&r,&g,&b)";
run;
```

Because the %RGB color utility macro accepts integer values in the range 0–100 as a percentage of 255, you must convert the 255-based values to integer percentages in order to use them in the %RGB macro call. The result is CXE8B3B5. The result from the %RGB macro is not exact due to rounding. If you want more exact results, use the following program.

```
data _null_;
  r = 231;
  g = 179;
  b = 180;
  rgb="CX" || put(r,hex2.) || put(g,hex2.) || put(b,hex2.);
  put rgb;
run;
```

The result is CXE7B3B4.

To convert the RGB color name to an HLS color name, use the %RGB2HLS macro as shown in the following program.

```
%COLORMAC;
data _null_;
  put "%RGB2HLS(CXE7B3B4)";
run;
```

The result is H077CD84.

For more information about the SAS/GRAPH color utility macros, see *SAS/GRAPH: Reference*.

Color Support for Universal Printers

All Universal Printers support 24-bit RGB colors. Most printers support 32-bit CMYK colors or 32-bit RGBA (transparency) colors. The following table lists the Universal Printers and their respective color support.

Table 13.20 *Color Support for Universal Printers*

Universal Printer	Color Support	Supports Transparency
EMF	RGBA (32–bit)	Yes
SASEMF	RGBA only for bitmap images. RGB for vector elements.	Yes, for bitmap images
GIF	RGBA (24–bit with support for transparent backgrounds)	Yes
PCL5c	RGBA	Yes
PDF	CMYK and RGBA	Yes, for RGBA colors
PNG	RGBA (32–bit)	Yes
PostScript	CMYK and RGB RGBA for GIF images	No No
SVG	RGBA (32–bit)	Yes
TIFF	RGBA	Yes
TIFFk	CYMK	No

Fonts

About Fonts in ODS Graphics

ODS Graphics provides a variety of fonts, or typefaces, to display text and special characters for your graphics output. SAS provides a number of TrueType fonts that you can use in your programs. By default, ODS styles use system fonts, including the TrueType fonts shipped with SAS, for titles, labels, and other text in ODS Graphics output. You can modify the default fonts by modifying the ODS styles or by using font options in procedure statements. You can specify special characters using character codes or hexadecimal codes.

TrueType Fonts Supplied by SAS

When you install SAS, a number of TrueType fonts are available. TrueType fonts that are supplied by SAS can be categorized into four groups: Windows Glyph List (WGL) Pan-European character set fonts; graphic symbol; multilingual; and monolingual Asian. The following tables show the fonts that are supplied by SAS.

Note: The fonts that are supplied by SAS are automatically registered in the SAS registry when you install SAS. To register other TrueType Fonts, see "Registering Fonts" in *SAS/GRAPH: Reference*.

Table 13.21 *Multilingual TrueType Fonts*

Font Name	Language Supported	Font Description
Arial Unicode MS[1]	Arabic, Armenian, Basic Latin, Bengali, Bopomofo, Cyrillic, Devanagari, Georgian, Greek and Coptic, Gujarati, Gurmukhi, hangul jamo, Hebrew, hiragana, Kanbun, Kannada, katakana, Lao, Malayalam, Oriya, Tamil, Telugu, Oriya, Tamil, Telugu, Thai Tibetan.	sans-serif

Font Name	Language Supported	Font Description
Times New Roman Uni[1]	Arabic, Basic Latin, Bopomofo, Cyrillic,Devanagari, Georgian, Greek and Coptic,Gujarati, hangul jamo, Hebrew, hiragana, Kanbun, katakana, Lao, Mongolian, Tamil, Telugu, Thai, Tibetan.	serif

1 The Arial Unicode MS and Times New Roman fonts replace the Monotype Sans WT and Thorndale Duospace WT fonts, respectively.

Table 13.22 *Monolingual Asian TrueType Fonts*

Language Supported	Font Name	Character Set
Korean	Gulim, Dotum	KSC5601
	Batang, Gungsuh	KSC5601
Japanese	MS Gothic, MS UI Gothic, MS PGothic	Shift JIS
	MS Mincho, MS PMincho	Shift JIS
Simplified Chinese	CSongGB18030C-Light CSongGB18030-LightHWL MYingHei_18030_C-Medium MYingHei_18030_C-MediumHWL	GB18030 and GB2312
Traditional Chinese[1]	Hei	Big5
	MingLiU, PMingLiU	Big5

1 Hei, MingLiU, and PMingLiu support HKSCS2004 (Hong Kong Supplemental Character Set) characters.

Table 13.23 *Anova UI and HelveticaNeueforSAS TrueType Fonts Distributed by SAS*

Font Name
Anova UI Italic[1]
Anova UI Roman[1]
Anova UI Bold Italic[1]

Font Name
Anova UI Bold Roman[1]
Anova UI Light Italic[1]
Anova UI Light Roman[1]
HelveticaNeueforSAS
HelveticaNeueforSAS Bold Italic
HelveticaNeueforSAS Bold
HelveticaNeueforSAS Italic
HelveticaNeueforSAS Light
HelveticaNeueforSAS Light Italic

1 The Anova UI fonts are available starting with 2025.06.

International Character Support

TrueType fonts support a wide range of international characters. These fonts support Unicode characters. To print international characters, specify a font that supports Unicode characters and use the Unicode value for each character. TrueType fonts support a wide range of international characters. Here is an example that includes the Euro currency symbol in a title:

```
title1 font="Arial Unicode MS" "Total Sales in " '20AC'x;
```

Here is an example uses the PRINT statement to print data that contains world currency symbols:

```
%macro addobs(name, char);                    /* 1 */
   name="&name";
   code=upcase("&char");
   char=kcvt("&char"x, 'ucs2b', 'utf8');
   output;
%mend addobs;

data utf8char;                                /* 2 */
   length name $40;
   label name="Name" code="Code" char="Character";
   %addobs(Cent Sign, 00A2);
   %addobs(Pound Sign, 00A3);
   %addobs(Currency Sign, 00A4);
   %addobs(Yen Sign, 00A5);
   %addobs(Rupee Sign, 20A8);
```

```
   %addobs(Euro Sign, 20AC);
   %addobs(Dong Sign, 20AB);
   %addobs(Euro-currency Sign, 20A0);
   %addobs(Colon Sign, 20A1);
   %addobs(Cruzeiro Sign, 20A2);
   %addobs(French Franc Sign, 20A3);
   %addobs(Lira Sign, 20A4);
run;
title "Currency Symbols";
proc print data=utf8char label                    /* 3 */
   style(data)={fontsize=16pt fontfamily='Arial Unicode MS'};
   id name code;
run;
```

1 Macro %ADDOBS generates an observation for a Unicode character code. The KCVT function converts the Unicode code to a UTF-8 character, which is stored in variable Char. See "KCVT Function" in *SAS National Language Support (NLS): Reference Guide*.

2 DATA step UTF8CHAR uses macro %ADDOBS to create a data set that contains several international characters.

3 The PRINT procedure prints the data. Option STYLE specifies font Arial Unicode MS, which supports Unicode characters.

Here is the output.

Currency Symbols

Name	Code	Character
Cent Sign	00A2	¢
Pound Sign	00A3	£
Currency Sign	00A4	¤
Yen Sign	00A5	¥
Rupee Sign	20A8	₨
Euro Sign	20AC	€
Dong Sign	20AB	₫
Euro-currency Sign	20A0	₠
Colon Sign	20A1	₡
Cruzeiro Sign	20A2	₢
French Franc Sign	20A3	₣
Lira Sign	20A4	₤

Determining What Fonts Are Available

The fonts listed in "TrueType Fonts Supplied by SAS" in *SAS/GRAPH: Reference* are available on all systems where SAS is installed. It is recommended that you use these fonts when possible. Additional system fonts that are available to your application and the methods for determining those fonts depend on the following:

- the operating environment that you are working in
- the device or universal printer that you are using

You can add additional fonts to your system for use by SAS/GRAPH, but all fonts must be registered with the FONTREG procedure. See *Base SAS Procedures Guide* for more information.

All of the fonts that have been registered with the FONTREG procedure are listed in the SAS registry. To display the list of registered fonts, use the REGISTRY procedure to list registry key CORE/PRINTING/FREETYPE/FONTS. See "REGISTRY Procedure" in *Base SAS Procedures Guide*.

The SAS/GRAPH fonts are available on all systems where SAS/GRAPH is installed, but they are provided primarily for special uses. See "Using SAS/GRAPH Fonts" in *SAS/GRAPH: Reference* for more information.

When you are deciding what font to use, consider all operating environments in which your SAS code will be run. If you specify a font that is not available on other systems, then your output will appear different on those systems. If you specify one of the fonts that are installed with SAS (see "TrueType Fonts Supplied by SAS" in *SAS/GRAPH: Reference*), then your output will appear the same on all systems. It is recommended that you use system fonts whenever possible.

Viewing Font Specifications in the SAS Registry

To view the font settings in the SAS registry, use the REGISTRY procedure to display the registry contents starting at registry key **ODS** ⇨ **Fonts**. See "REGISTRY Procedure" in *Base SAS Procedures Guide*.

Each entry in the registry consists of a name, such as `<MTsans-serif>` or `<MTmonospace>` followed by its value, such as "Albany AMT" or "Cumberland AMT".

Note: The Fonts key contains subkeys that specify which fonts to use based on the locale.

For more information, see "The SAS Registry" in *SAS Programmer's Guide: Essentials*.

Listing the Registered Fonts for a Universal Printer

You can use the QDEVICE procedure to view the list of fonts that have been registered in the SAS registry, including fonts that you registered with the FONTREG procedure. The following code creates macro %FONTLIST, which you can use to list the available fonts for a Universal Printer.

```
%macro fontlist(name);
   proc qdevice report=font out=fonts;
      printer "&name";
      var font ftype fstyle fweight;
   run;

   data fontdata;
      set fonts;
      drop ftype;
      length type $16;
      nullChar = index(font,'00'x);
      if (nullChar) then do;
         font = substr(font,1,nullChar);
      end;
      if ftype = "System"
         then do;
         if substr(font,2,3) = "ttf" then type = "TrueType";
            else if substr(font,2,3) = "at1" then type = "Adobe Type1";
            else if substr(font,2,3) = "cff" then type = "Adobe CFF/
Type2";
            else if substr(font,2,3) = "pfr" then type = "Bitstream
PFR";
            else type = "System";
         if type ^= "System" then font = substr(font,7,length(font)-6);
            else if substr(font,1,1) = "@"
               then font = substr(font, 2,length(font)-1);
         end;
      else type = "Printer Resident";
      drop nullChar;
   run;

   proc sort;
      by font;
   run;

   title "Fonts Supported by the %upcase(&name) Printer";
   proc print label;
      label fstyle="Style" fweight="Weight" font="Font" type="Type";
   run;
%mend fontlist;
```

Macro argument *name* specifies the name of the Universal Printer. The following statement lists the fonts that are available with the SVG Universal Printer:

```
%fontlist(svg)
```

Here is an example of the output:

Fonts Supported by the SVG Printer

Obs	Font	Style	Weigth	Type
1	Albany AMT	Italic	Bold	TrueType
2	Albany AMT	Italic	Normal	TrueType
3	Albany AMT	Roman	Bold	TrueType
4	Albany AMT	Roman	Normal	TrueType
5	Anova	Italic	Normal	TrueType
6	Anova	Roman	Normal	TrueType
7	Anova Bold	Italic	Normal	TrueType
8	Anova Bold	Roman	Normal	TrueType
9	Anova Light	Italic	Normal	TrueType
10	Anova Light	Roman	Normal	TrueType
11	Anova UI	Italic	Normal	TrueType
12	Anova UI	Roman	Normal	TrueType
13	Anova UI Bold	Italic	Normal	TrueType
14	Anova UI Bold	Roman	Normal	TrueType
15	Anova UI Light	Italic	Normal	TrueType
16	Anova UI Light	Roman	Normal	TrueType
17	Arial Symbol	Italic	Bold	TrueType
18	Arial Symbol	Italic	Normal	TrueType
19	Arial Symbol	Roman	Bold	TrueType
20	Arial Symbol	Roman	Normal	TrueType

For more information, see "QDEVICE Procedure" in *Base SAS Procedures Guide*.

Registering Fonts

In addition to the TrueType fonts that come installed with SAS, SAS supports PostScript Type1 fonts. The following table shows the font prefix and file extension for TrueType and Type1 fonts:

Table 13.24 *Font Types Supported by SAS*

Type	Tag	File Extension
TrueType	<ttf>	.ttf
Type1	<at1>	.pfb [1]

1 Data from a .pfm file is used to generate output using the WMF device and the EMF universal printer. This file is not required to register Type1 fonts using PROC FONTREG. If you do not register a .pfm file, you might not have the desired results.

The fonts that are supplied by SAS are automatically registered in the SAS registry when you install SAS. Custom fonts such as fonts that are installed after you install SAS or system fonts that are not available in SAS must be registered manually in the SAS registry by using the FONTREG procedure.

Note: Registering a font in the SAS registry means that SAS knows where to find the font file. The font file is not actually used until the font is called for in a SAS program.

The FONTREG procedure enables you to update the SAS registry to include custom fonts, which can then be used in SAS output. PROC FONTREG uses FreeType font-rendering to recognize and incorporate various types of font definitions. Fonts of any type that can be incorporated and used by SAS are known collectively in this documentation as fonts in the FreeType library.

To register custom fonts in SAS, submit the following SAS program.

```
proc fontreg;
   fontpath 'pathname';
run;
```

The FONTPATH statement specifies the directory that contains the fonts and **pathname** is the directory path of the fonts. The directory path must be accessible to the SAS Compute server.

Note: Do not move or delete the font files after you have registered your fonts in the SAS registry.

In SAS Data and AI Studio known as SAS Studio prior to 2026.06), when you run the FONTREG procedure in a Program Editor window, the registered fonts are made available for the duration of the session. When you sign out, your custom font registration is lost. You must register your custom fonts each time you start a new SAS Data and AI Studio session. You can add the FONTREG procedure step code to your SAS Data and AI Studio autoexec file to register your fonts automatically each time you start SAS Data and AI Studio. For more information, see *SAS Data and AI Studio: User's Guide*.

14

Commonly Used Attribute Options

General Syntax for Attribute Options **1827**
See Also 1828
Line Attributes and Patterns **1828**
See Also 1832
Fill Attributes **1832**
See Also 1833
Marker Attributes and Symbols **1833**
See Also 1835
Text Attributes **1835**
See Also 1836
Units of Measurement **1836**

General Syntax for Attribute Options

Most statements provide options that enable you to specify attributes for the lines, fills, data markers, or text that is used in the display. For example, many plots provide a DATALABELATTRS= option that specifies the attributes of the data labels. This section discusses the general syntax for those options and the valid values that they accept.

A statement's attribute options use the following general syntax:

OPTION-NAME= *style-element* | *style-element (options)* | *(options)*

style-element
name of a style element.

A style element is a named collection of style attributes that affects specific parts of your output. For example, a style element might specify the color and font properties of title text or the fill properties of a bar chart. An ODS style is a

collection of style elements that provides specific visual attributes for your SAS output.

Only style attributes relevant for rendering the fill, line, data marker, or text are used.

See "Specifying a Style Element" on page 1848

Example
```
density height / lineattrs=graphfit2;
```

style-element (options)
name of a style element, plus individual *options* to be used as style overrides. Any *options* not specified are derived from the specified *style-element*.

See "Specifying a Style Element with Hardcoded Values" on page 1850

Example
```
density height / lineattrs=graphfit2
    (pattern=dashdashdot);
```

(options)
individual *options*. Any *options* not specified are derived from the default style element.

See "Specifying Hardcoded Values" on page 1849

Example
```
density height / lineattrs= (pattern=dashdashdot color=red);
```

Depending on the attribute option used, the *options* might be one of the following commonly used option types:

- "Line Attributes and Patterns" on page 1828
- "Fill Attributes" on page 1832
- "Marker Attributes and Symbols" on page 1833
- "Text Attributes" on page 1835

See Also

"Style Elements for Use with ODS Graphics" on page 1871

Line Attributes and Patterns

Line options and patterns specify the appearance of particular lines in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

When specifying the *options*, use a *name=value* pair format enclosed in parentheses. If you specify more than one *name=value* pair, separate them by a space. For example:

```
(pattern=dashdashdot color=red)
```

Here is an example that uses the LINEATTRS= option for a series plot:

```
lineattrs=(pattern=thindot color=lightgreen thickness=10)
```

options can be one or more of the following attributes.

COLOR=*color*
specifies the color of the line. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

PATTERN=*line-pattern*
specifies the line pattern for the line. Line patterns can be specified as a pattern name or pattern number.

Table 14.1 *Named Line Patterns*

Pattern Number	Pattern Name	Example
1	Solid	
2	ShortDash	
4	MediumDash	
5	LongDash	
8	MediumDashShortDash	
14	DashDashDot	
15	DashDotDot	
20	Dash	
26	LongDashShortDash	
34	Dot	
35	ThinDot	
41	ShortDashDot	
42	MediumDashDotDot	

Valid pattern numbers range from 1 to 46. Not all pattern numbers have names. The following table lists all possible line patterns. We recommend that you use the named patterns because they have been optimized to provide good discriminability when used in the same plot.

Table 14.2 *Full List of Line Patterns*

Pattern Number	Pattern Name	Example
1	Solid	
2	ShortDash	
3		
4	MediumDash	
5	LongDash	
6		
7		
8	MediumDashShortDash	
9		
10		
11		
12		
13		
14	DashDashDot	
15	DashDotDot	
16		
17		
18		
19		
20	Dash	

Pattern Number	Pattern Name	Example
21		
22		
23		
24		
25		
26	LongDashShortDash	
27		
28		
29		
30		
31		
32		
33		
34	Dot	
35	ThinDot	
36		
37		
38		
39		
40		
41	ShortDashDot	
42	MediumDashDotDot	
43		

Pattern Number	Pattern Name	Example
44		
45		
46		SGPANEL, SGPLOT, and SGRENDER procedures:
		SGMAP and SGSCATTER procedures:

THICKNESS=*numeric-value <units>*
specifies the thickness of the line. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

See Also

- "General Syntax for Attribute Options" on page 1827
- "Style Elements for Use with ODS Graphics" on page 1871

Fill Attributes

Fill options specify the appearance of an area fill in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

When specifying the *options*, use a *name=value* pair format enclosed in parentheses. If you specify more than one *name=value* pair, separate them by a space. For example:

```
(color=red transparency=.5)
```

Here is an example that uses the FILLATTRS= option for a bar chart:

```
fillattrs=(color=green transparency=.2)
```

options can be one or more of the following attributes.

COLOR=*color*
specifies the fill color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

TRANSPARENCY=*number*
specifies the degree of the transparency of the filled area.

Default 0

Range 0 (completely opaque) to 1 (completely transparent)

Note Not all fill options enable you to specify the transparency attribute. For example, the FILLATTRS= option in the BAND statement does not support transparency. The reason is that the BAND plot statement includes its own TRANSPARENCY= option.

Example
```
fillattrs=(transparency=0.5)
```

See Also

- "General Syntax for Attribute Options" on page 1827
- "Style Elements for Use with ODS Graphics" on page 1871

Marker Attributes and Symbols

Marker options specify the appearance of the markers in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

When specifying the *options*, use a *name=value* pair format enclosed in parentheses. If you specify more than one *name=value* pair, separate them by a space. For example:

```
(symbol=diamond color=red)
```

Here is an example that uses the MARKERATTRS= option for a scatter plot:

```
markerattrs=(color=lightgreen size=10 symbol=trianglefilled)
```

options can be one or more of the following attributes.

COLOR=*color*
specifies the color of the markers. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

SIZE=*numeric-value <units>*
specifies the size of the markers. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

SYMBOL=*symbol-name*
specifies the symbol for the markers.

Table 14.3 Supported Marker Symbols

Symbol Name	Plot Symbol	Symbol Name	Plot Symbol
ArrowDown		StarFilled	
Asterisk		Tack	
Circle		Tilde	
CircleFilled		Triangle	
Diamond		TriangleFilled	
DiamondFilled		TriangleDown	
GreaterThan		TriangleDownFilled	
Hash		TriangleLeft	
HomeDown		TriangleLeftFilled	
HomeDownFilled		TriangleRight	
IBeam		TriangleRightFilled	
LessThan		Union	
Plus		X	
Square		Y	
SquareFilled		Z	
Star			

See Also

- “General Syntax for Attribute Options” on page 1827
- “Style Elements for Use with ODS Graphics” on page 1871

Text Attributes

Text options specify the appearance of particular text elements in the plot. You can specify the appearance by using a style element or by specifying specific options. If you specify a style element, you can also specify options to override specific appearance attributes.

When specifying the *options*, use a *name=value* pair format enclosed in parentheses. If you specify more than one *name=value* pair, separate them by a space. For example:

```
(family="Arial" size=10pt color=red)
```

Here is an example that uses the DATALABELATTRS= option for a bar chart:

```
datalabelattrs=(Color=lightgreen Family="Arial" Size=8 Style=italic Weight=bold)
```

options can be one or more of the following attributes.

COLOR=*color*
specifies the color of the text. You can specify colors using a number of different color-naming schemes. For more information, see “Color-Naming Schemes” on page 1808.

FAMILY="*font-family*"
specifies the font family for the text. The SAS ODS styles use TrueType system fonts. For more information, see “Fonts” on page 1819.

SIZE=*numeric-value* <*units*>
specifies the font size of the text. The default unit is pixels. For a list of measurement units that are supported, see “Units of Measurement” on page 1836.

Note For some data and curve labels, font sizes might be automatically reduced in order to achieve less overlapping. This behavior affects all data labels with non-fixed positions, and curve labels for which you have specified a curve label position of MIN or MAX.

STYLE=ITALIC | NORMAL
specifies the font style of the text.

Note If the text font does not support italic typeface, STYLE=ITALIC is ignored.

WEIGHT=BOLD | NORMAL
specifies the font weight of the text.

Note If the text font does not support bold typeface, WEIGHT=BOLD is ignored.

System fonts can be used by the SAS ODS Graphics procedures. These fonts include TrueType, Type1, and Opentype fonts. Examples of system fonts include Albany AMT, Monotype Sorts, and Arial. System fonts must be installed on the operating system, and then registered with SAS using the FONTREG procedure. For more information, see "FONTREG Procedure" in *Base SAS Procedures Guide*.

See Also

- "General Syntax for Attribute Options" on page 1827
- "Style Elements for Use with ODS Graphics" on page 1871

Units of Measurement

Some options enable you to specify the unit of measurement as part of the value.

For example, when using the LINEATTRS= option, you can specify the measurement unit for line thickness. When using the DATALABELATTRS= option, you can specify the font size unit for your data labels.

The following table contains the units that are available:

Table 14.4 *Measurement Units*

Unit	Description
CM	centimeters
IN	inches
MM	millimeters
PCT or %	percentage
PT	point size, calculated at 72 dots per inch
PX	pixels

PART 5

Controlling the Procedure Output

Chapter 15
Controlling the Appearance of Your Graphs 1839

Chapter 16
Managing Your Graphics with ODS 1883

1838

15

Controlling the Appearance of Your Graphs

Overview 1840

Understanding Styles 1841
About Styles and Style Elements 1841
About the Default Styles 1842
Recommended Styles 1842
Viewing a Style Template 1845

Specifying Styles 1847

Using Plot Options to Control Graph Appearance 1847
Overview of the Appearance Options 1847
Specifying a Style Element 1848
Specifying Hardcoded Values 1849
Specifying a Style Element with Hardcoded Values 1850

Using Data Skins 1851

Subpixel Rendering 1857

Output for Grouped versus Non-Grouped Data 1858
Non-Grouped Data 1858
Grouped Data 1862

Modifying Style Templates 1864

Using Fill Patterns to Distinguish Categories of Data 1865
About Using Fill Patterns 1865
Using Plot Options to Display Fill Patterns 1866
Using Gray-Scale Styles to Distinguish Grouped Bar Charts 1870

Style Elements for Use with ODS Graphics 1871

Overview

Along with table and page attributes, ODS styles contain a collection of graphical attributes such as color, marker shape, line pattern, fonts, and so on. Many carefully designed styles are shipped with SAS that enhance the visual impact of the graphics.

In addition to creating visually appealing graphics, many key elements of effective graphics are built into the various elements of the styles, such as the following:

- provide maximum contrast between backgrounds and data
- provide for ease of discrimination among different groups
- provide equal emphasis for data with equal importance

The ODS styles are used to produce professional looking graphics without the need for further modification. However, you have several options for modifying the appearance of your graphs if you choose to do so.

You can customize your graphical output at three levels.

The following table shows the three levels of customization.

Table 15.1 *Levels of Customization*

Method	Description	Level of Complexity
Change the style. See "Specifying Styles" on page 1847 .	Specify a style with the STYLE= option on the ODS destination statement to change the appearance of the entire graph. Requires no further modification. The active style affects all SAS output and remains in effect until the ODS style or destination is changed.	low
Change attributes for graphs that specify a group variable. See "Grouped Data" on page 1862.	Change group colors, markers, and so on, within the procedure, without having to change the ODS style template. The appearance options override the corresponding defaults from the current style. The appearance options affect only the procedure in which they are specified.	medium
Use plot options. See "Using Plot Options to	Specify an appearance option in one or more plot statements to change	medium

Method	Description	Level of Complexity
Control Graph Appearance" on page 1847 .	various aspects of your graph. The appearance options override the corresponding defaults from the current style. **Note:** Not all appearance attributes can be specified in this way. The appearance options affect only the plot statement in which they are specified.	
Modify individual style elements. See "Modifying Style Templates" on page 1864 .	Specify or change style attributes in order to modify a style element. This requires the use of PROC TEMPLATE style statements. The active style affects all SAS output and remains in effect until the ODS style or destination is changed.	high

Understanding Styles

About Styles and Style Elements

ODS styles are produced from compiled style templates written in PROC TEMPLATE style syntax. An ODS style template is a collection of style elements that provides specific visual attributes for your SAS output.

The style elements of a style are designed to ensure the goals of effective graphics. Each style element is a named collection of style attributes such as color, marker symbol, line style, font face, as well as many others. Each graphical element of a plot, such as a marker, a bar, a line or a title, derives its visual attributes from a specific style element from the active style.

Changing the style for an ODS destination is the easiest way to change a graph's appearance. Changing the current style requires only the use of the STYLE= option on an ODS destination statement. For more information, see "Specifying Styles" on page 1847.

Note: The style that a destination uses is applied to tabular output as well as graphical output.

About the Default Styles

Every ODS output destination is associated with a default style. These default styles are different for each destination. Therefore, your output might look different depending on which destination you use.

For a table that lists the default styles for ODS destinations, see "Working with Styles " in *SAS Output Delivery System: Procedures Guide*.

You can display a list of the available styles by submitting the following PROC TEMPLATE statements:

```
proc template;
   list styles;
run;
```

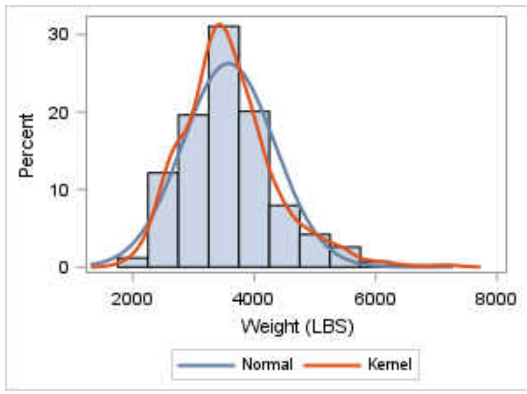
Recommended Styles

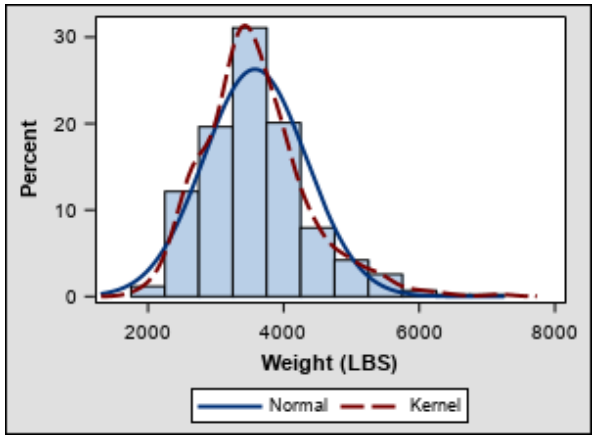
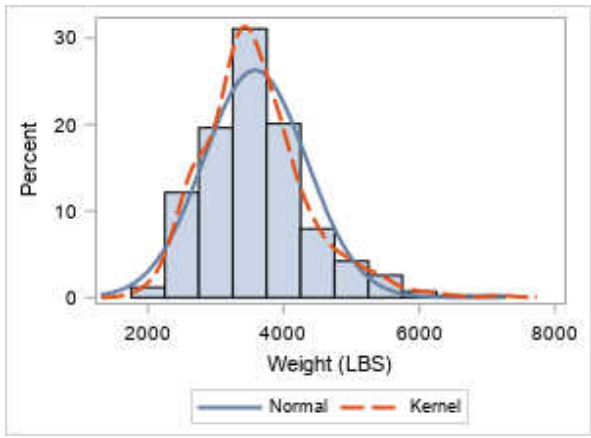
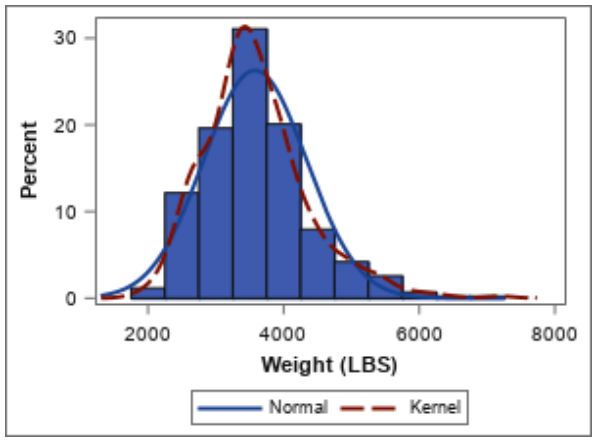
SAS ships a set of styles that have been designed by GUI experts to address the needs of different situations, while ensuring the principles of effective graphics.

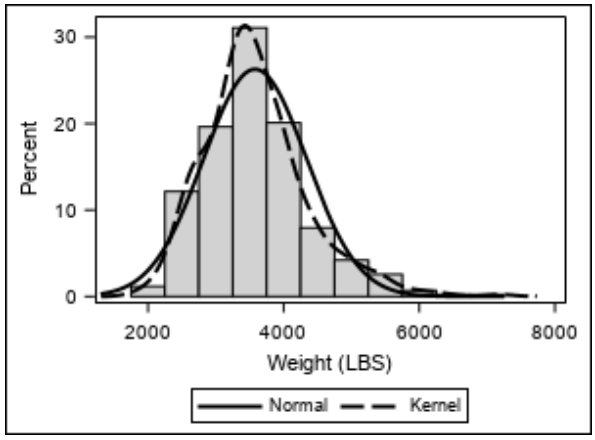
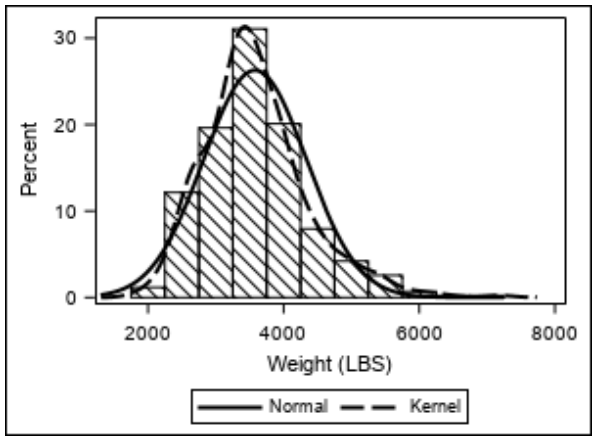
The following table shows a subset of the styles shipped with SAS that are suitable for statistical graphics. The output shown was generated by running the following code:

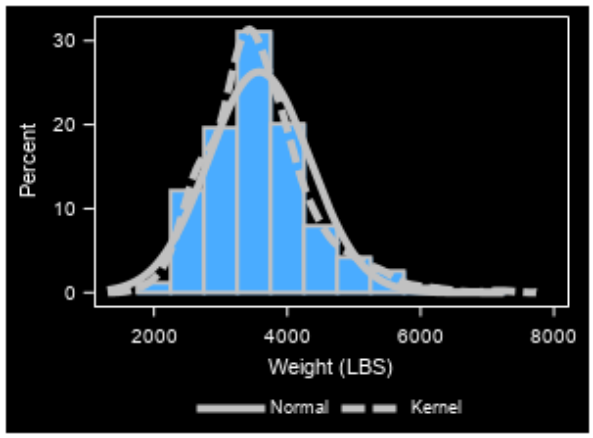
```
proc sgplot data=sashelp.cars;
   histogram weight;
   density weight / type=normal;
   density weight / type=kernel;
run;
```

Table 15.2 *Recommended ODS Styles*

Style	Example
HTMLENCORE[1] ■ white background ■ white wall ■ sans-serif fonts ■ table colors match the graph colors ■ a lighter color scheme for HTML content ■ group distinctions based primarily on color rather than markers, line styles, and fill patterns, when displayed	

Style	Example
DEFAULT [1] ■ gray background ■ white wall ■ sans-serif fonts ■ group distinctions based on color, line styles, and fill patterns, when displayed	
STATISTICAL [1] ■ white background ■ white wall ■ sans-serif fonts ■ contrasting color scheme of blues, reds, greens, and yellows for markers, lines, fill colors, and fill patterns ■ group distinctions based on color, line styles, and fill patterns, when displayed	
DAISY [1] ■ white background ■ white wall ■ sans-serif fonts ■ a higher contrast color scheme for lines, markers, and fill colors, and fill patterns ■ group distinctions based on color, line styles, and fill patterns, when displayed ■ recommended for creating accessible graphs (see "Using ODS Styles to Create Accessible Output" in *Creating Accessible SAS Viya Platform Output Using ODS and ODS Graphics*)	

Style	Example
JOURNAL and JOURNAL3[1, 2] ■ white background ■ white wall ■ sans-serif fonts ■ gray-scale color scheme for fill colors, and black-only color scheme for markers, lines, and fill patterns ■ group distinctions based on color, line styles, and fill patterns, when displayed ■ for bar charts, gray-scale fill color scheme and back-only fill pattern color scheme by default (JOURNAL3 only)	
JOURNAL2 [2] ■ white background ■ white wall ■ sans-serif fonts ■ gray-scale color scheme for fill colors, and black-only color scheme for markers, lines, and fill patterns ■ group distinctions based on color, line styles, and fill patterns, when displayed ■ for band plots, bar charts, box plots, ellipse plots, high-low plots, histograms, and polygon plots, black-only color scheme for fill patterns, by default ■ no solid filled areas—a minimal ink style ■ recommended for creating accessible graphs (see "Using ODS Styles to Create Accessible Output" in *Creating Accessible SAS Viya Platform Output Using ODS and ODS Graphics*)	

Style	Example
HIGHCONTRAST ■ black background ■ black wall ■ sans-serif fonts ■ a white-on-black high-contrast color scheme for lines, markers, fill colors, and fill patterns ■ group distinctions based primarily on color rather than markers, line styles, and fill patterns, when displayed	

1 Supports fill patterns.

2 The JOURNAL2 and JOURNAL3 styles, by default, render grouped bars with fill patterns. For more information, see "Using Fill Patterns to Distinguish Categories of Data" on page 1865.

Viewing a Style Template

Use the TEMPLATE procedure to write the source for a style template to the SAS log for viewing. The following example shows how to print the Illuminate style template source to the SAS log:

```
proc template;
   path sashelp.tmplmst;
   source Styles.Illuminate;
run;
```

Here is a partial listing of the output:

```
1    proc template;
2    path sashelp.tmplmst;
3    source Styles.Illuminate;
define style Styles.Illuminate;
   parent = styles.HTMLEncore;
   class color_list
      "Colors used in the default style" /
      'fgB2' = cx525f70
      'fgB1' = cx626f80
      'bgA3' = cxffffff
      'fgA2' = cx696969
      'bgA2' = cxffffff
      'fgA1' = cx2a3037
      'bgA1' = cxffffff
      'fgA' = cx696969
      'bgA' = cxffffff;
   class colors
      "Abstract colors used in the default style" /
      'headerfgemph' = color_list('fgA1')
      'headerbgemph' = color_list('bgA1')
      'headerfgstrong' = color_list('fgA1')
      'headerbgstrong' = color_list('bgA1')
      'headerfg' = color_list('fgA1')
      'headerbg' = color_list('bgA1')
      'datafgemph' = color_list('fgA1')
      'databgemph' = color_list('bgA3')
      'datafgstrong' = color_list('fgA1')
      'databgstrong' = color_list('bgA3')
      'datafg' = color_list('fgA1')
      'databg' = color_list('bgA3')
      'batchfg' = color_list('fgA1')
      'batchbg' = color_list('bgA3')
      'tableborder' = color_list('fgA2')
      'tablebg' = color_list('bgA1')
      'notefg' = color_list('fgA')
      'notebg' = color_list('bgA')
      'bylinefg' = color_list('fgA2')
      'bylinebg' = color_list('bgA2')
      'captionfg' = color_list('fgA1')
      'captionbg' = color_list('bgA')
      'proctitlefg' = color_list('fgA')
      'proctitlebg' = color_list('bgA')
      'titlefg' = color_list('fgA')
      'titlebg' = color_list('bgA')
      'systitlefg' = color_list('fgA')
      'systitlebg' = color_list('bgA')
      'Conentryfg' = color_list('fgA2')
      'Confolderfg' = color_list('fgA')
      'Contitlefg' = color_list('fgA')
      'link2' = color_list('fgB2')
      'link1' = color_list('fgB1')
      'contentfg' = color_list('fgA2')
      'contentbg' = color_list('bgA2')
      'docfg' = color_list('fgA')
      'docbg' = color_list('bgA');
...
```

For more information, see "TEMPLATE Procedure: Managing Template Stores" in *SAS Output Delivery System: Procedures Guide* and "ODS Styles Reference" in *SAS Output Delivery System: User's Guide*.

Specifying Styles

Each ODS destination has a default style that is set by SAS. By changing the style, you can create an entirely different appearance for your graphs. You can change the default ODS style in your IDE. For more information, refer to the documentation for your IDE.

As an alternative, you can specify STYLE=*style-template* in your ODS destination statement in your program. The following example shows how to change the default style for the ODS HTML5 destination to Illuminate:

```
ods html 5 style=illuminate;
```

For more information, see *SAS Output Delivery System: User's Guide*.

Using Plot Options to Control Graph Appearance

Overview of the Appearance Options

ODS styles enable you to control the overall appearance of the graphs. The ODS Graphics procedures also enable you to control the appearance of particular graphics elements in a graph. Graphics elements include lines, bars, markers, text, and so on.

Many ODS Graphics procedure statements have options and suboptions that control the appearance of different parts of a plot or graph. Default visual attributes of various graphics elements are derived from the specific style elements of the active style. By using appearance options in your procedure statements, you can change the appearance of one or more aspects of your graph without changing the overall style.

For example, the following statement specifies that the mean value confidence limits (for a regression plot) using the GraphConfidence2 style element instead of the default GraphConfidence style element:

```
reg x=height y=weight / clmattrs=GraphConfidence2;
```

You can specify values for appearance options by using three different methods:

- specify a style element.

```
density height / lineattrs=graphfit2;
```

Note: This is the preferred method.

See "Specifying a Style Element" on page 1848.

- specify hardcoded values.

```
density height / lineattrs=(pattern=dashdashdot color=red);
```

  This method overrides one or more attributes of the style element. See "Specifying Hardcoded Values" on page 1849.

- specify a style element, but override one or more attributes with hardcoded values. This method combines the first two methods.

```
density height / lineattrs=graphfit2 (pattern=dashdashdot);
```

  See "Specifying a Style Element with Hardcoded Values" on page 1850.

The appearance options and their values are specific to each statement. For complete documentation about the statements and their options, see the syntax section for the appropriate procedure and statement.

See Also

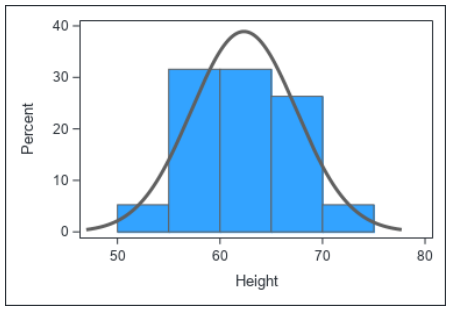
"About Styles and Style Elements" on page 1841

Specifying a Style Element

You can change the visual attributes that are used for a graphics element by specifying a particular style element for any appearance option. Graphics elements include lines, bars, markers, text, and so on.

For example, the graph shown here is a histogram with a normal density curve. By default, the visual attributes of the density curve are specified by the style attributes within the style element GraphFit.

Table 15.3 *Histogram with Density Curve That Uses the Default Style Element*



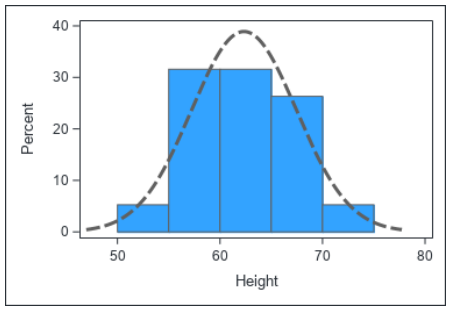

```
proc sgplot data=sashelp.class
    noautolegend;
  histogram height;
  density height;
run;
```

You can use the LINEATTRS= option in the DENSITY statement to change the style element that is used for the line attributes. For example, when you are using the Illuminate style, specifying the GraphFit2 style element changes the density curve line to a gray dashed line.

```
density height / lineattrs=graphfit2;
```

The full code is below.

Table 15.4 *Histogram with Density Curve That Specifies a Style Element*




```
proc sgplot data=sashelp.class
    noautolegend;
  histogram height;
  density height / lineattrs=graphfit2;
run;
```

Specifying a style element is the preferred way to change the appearance of a graphics element. Visual attributes that are referenced by the style element are chosen to provide consistency and appropriate emphasis based on display principles for statistical graphics. If you later change the style for the graph, the graphics element is compatible with the new style.

See Also

- "About Styles and Style Elements" on page 1841
- "Style Elements for Use with ODS Graphics" on page 1871

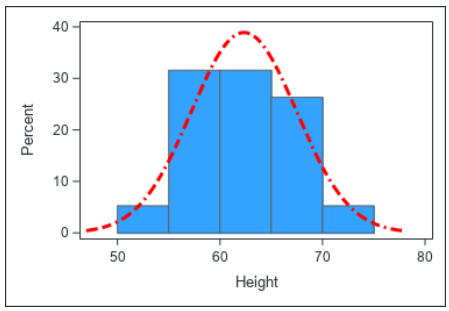
Specifying Hardcoded Values

In some cases it is important to have a specific visual effect, but it is not necessary for the plot to be reused with different styles. In such a case, you can specify hardcoded values for the visual attributes in the appearance option.

For example, you could use the following statement to set the density curve attributes:

```
density height / lineattrs=(pattern=dashdashdot color=red);
```

Table 15.5 *Using Hardcoded Values*




```
proc sgplot data=sashelp.class
    noautolegend;
  histogram height;
  density height /
   lineattrs=(pattern=dashdashdot color=red);
run;
```

The result is a red density curve with a DashDashDot pattern.

In general, it is recommended that you specify the style element rather than hardcoded values. Style elements are designed to derive all of the relevant visual attributes from the applied style. When you specify a hardcoded value for the line color, as in this example, the color is no longer derived from the specified style element. If you later change the style that is applied to the graph, the hardcoded attribute might conflict with the new style. Also, a hardcoded color could be exactly the same color as the wall color of another style, and become indistinguishable from the wall.

For example, if you change the overall style to Journal, which uses only gray-scale colors, the color for the curve is still red. This result might not be desirable.

You use a standard syntax to hardcode colors, line thickness, line patterns, and marker symbols.

For more information about this syntax, see the following topics:

- “Color-Naming Schemes” on page 1808
- “Units of Measurement” on page 2079
- List of Line Patterns on page 1829
- List of Marker Symbols on page 1834

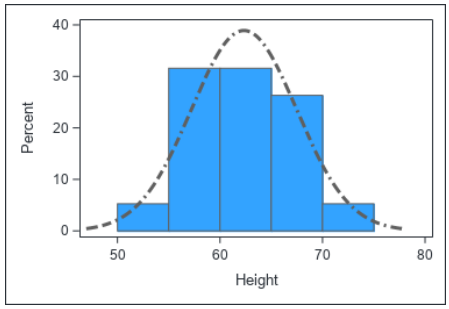
Specifying a Style Element with Hardcoded Values

You can specify a style element, but override one or more attributes of that style element with hardcoded values.

This example modifies a density curve. The example specifies the GraphFit2 style element for line attributes, but also specifies a particular line pattern with a hardcoded value.

```
density height / lineattrs=graphfit2 (pattern=dashdashdot);
```

Table 15.6 *Using a Hardcoded Value with a Style Element, Default Style*




```
proc sgplot data=sashelp.class
      noautolegend;
  histogram height;
  density height / lineattrs=graphfit2
    (pattern=dashdashdot);
run;
```

If you change the overall style to Journal, the color for this curve changes as well. However, the pattern remains DashDashDot because it was hardcoded.

Always be careful when you hardcode a value. When you specify a hardcoded value for the line pattern, as in this example, the pattern is no longer derived from the specified style element. If you later change the style that is applied to the graph, the hardcoded attribute might conflict with the new style.

Using Data Skins

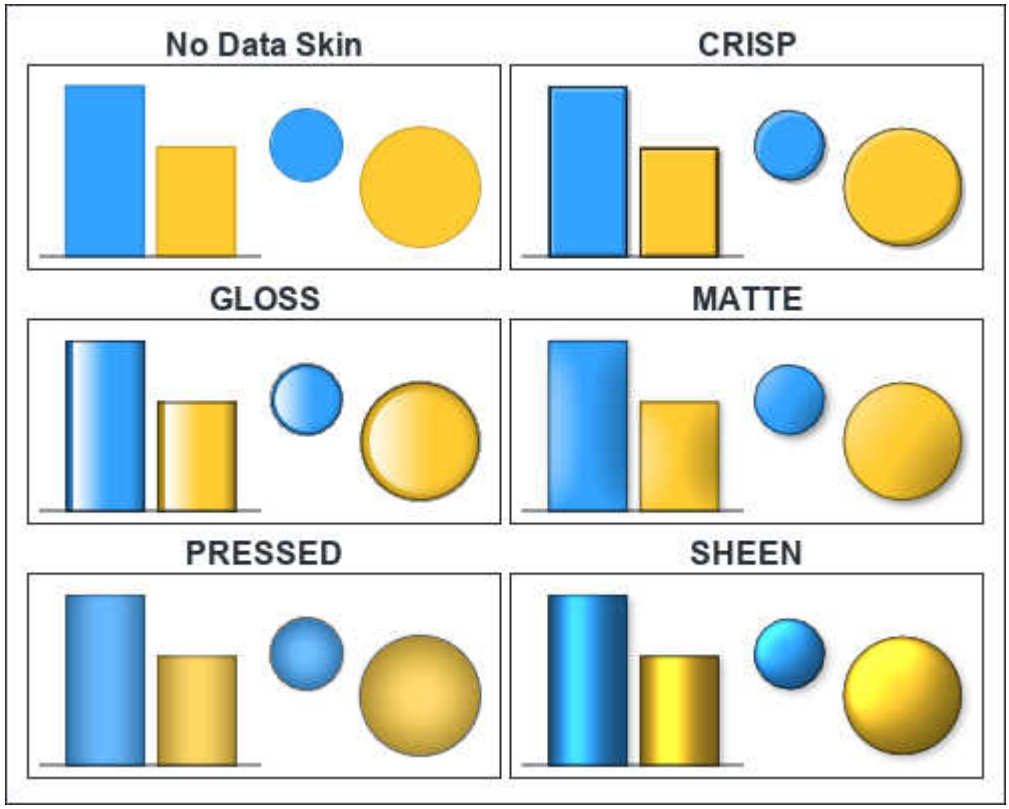
Data skins add a heightened visual effect to two-dimensional plots. Each skin uses shading, highlighting, and shadowing to give the appearance of contour and depth to certain elements of a graph, including the legend. For plots, the effect is generated by filters and is applied to filled areas, markers, and lines. When a data skin is applied to a filled area, it does not change the underlying fill color and pattern of the area. Typically, a data skin sets the area fill outline color to black. The outline color is controlled by the filters that generate the skin and is not controlled by the ODS style attributes or any custom outline attributes that are specified. For very small or very narrow filled areas, the data skin might not draw an outline around the filled area. For filled outlined markers, the outline color is determined by the ODS style attributes or by any custom marker attributes that are specified.

The effect that a data skin has on a filled area depends on the skin type, and on the size and color of the filled area. Because the ODS style determines the fill color by default, the effect can depend on the ODS style. Some skins have a greater effect than others. Most of the skins work best with lighter colors over a medium to large filled area. Over small filled areas and with some fill colors, the effect can be significantly reduced.

Note: Some ODS styles such as JOURNAL2 use pattern fill for certain areas rather than color fill. For these styles, data skins have no effect on the pattern-filled areas.

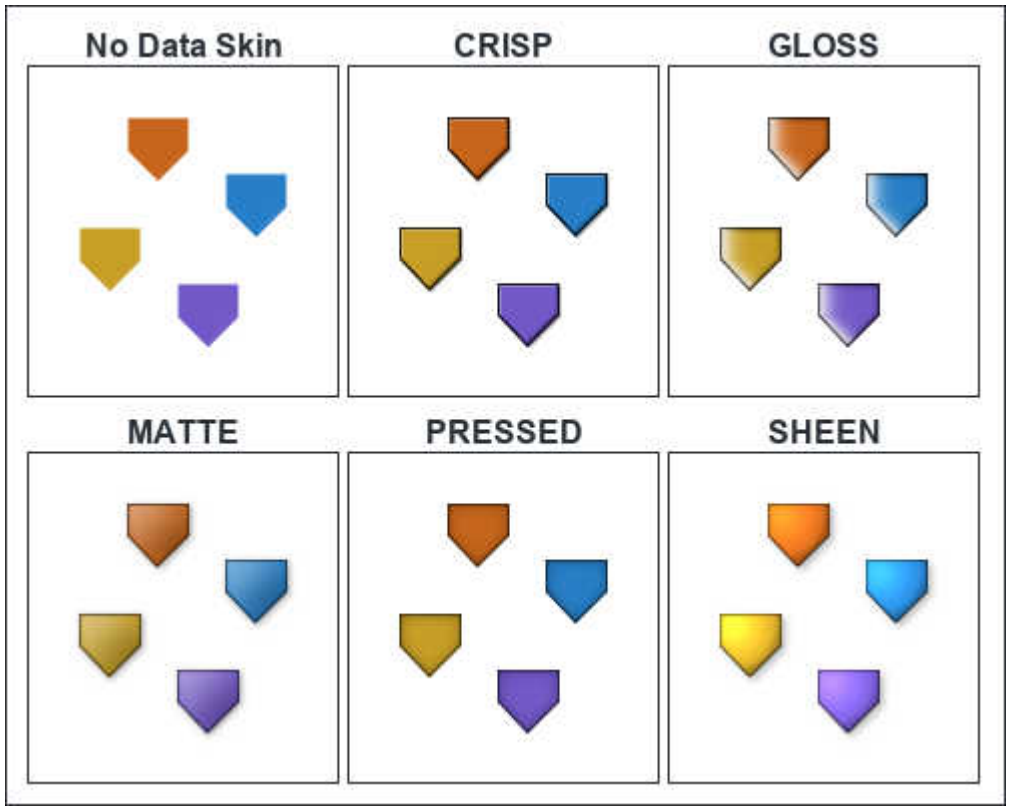
You can apply data skins to filled areas, markers, and lines in a plot. The data skins include CRISP, GLOSS, MATTE, PRESSED, and SHEEN. The following figure shows the effect of each data skin on filled bars and bubbles with the default Illuminate ODS style. A display with no data skin applied is included for comparison.

Figure 15.1 *Data Skins Applied to Filled Bars and Bubbles*



The following figure shows each of the data skins applied to large HOMEDOWNFILLED markers.

Figure 15.2 *Data Skins Applied to HOMEDOWNFILLED Markers*

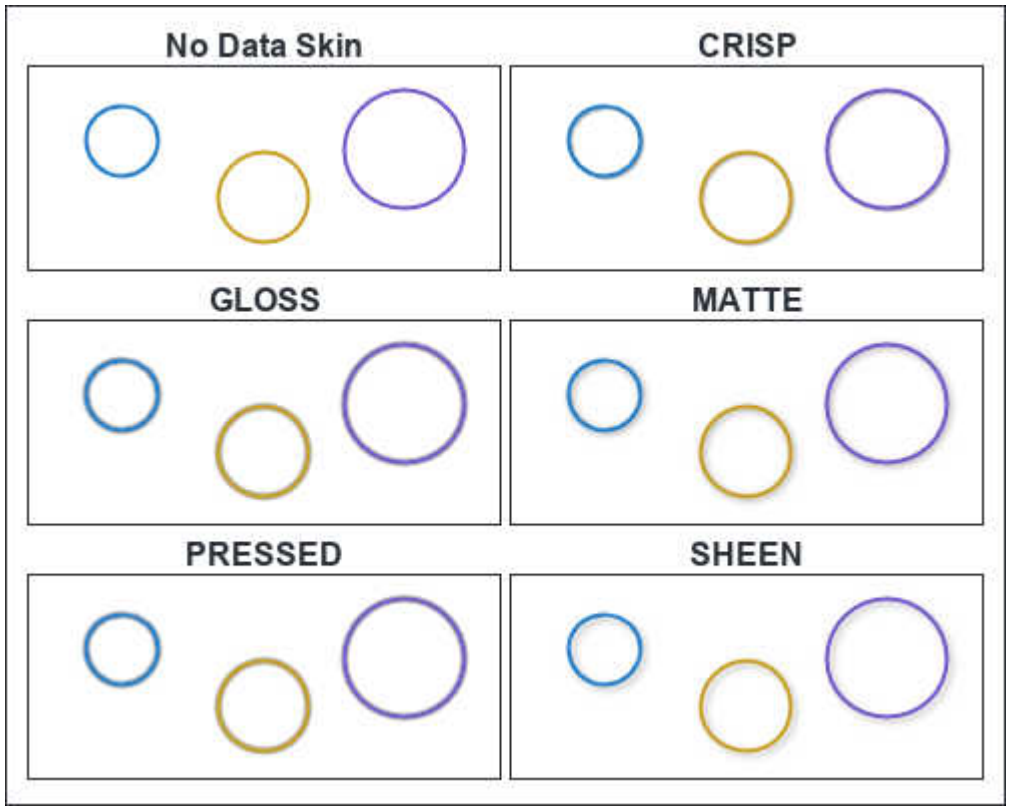


The effect of a data skin on filled markers is more apparent when the markers are enlarged.

Except for the GLOSS data skin, the data skins also affect the appearance of outlines for unfilled bubbles and markers. They do not affect the outlines of unfilled bars, boxes, and so on. As with filled areas, the effect of data skins on outlines varies with skin type and outline color. It is also more apparent when the thickness of the outlines is increased. The skins do not change the color of the outlines. They add subtle effects such as drop shadows that enhance their appearance.

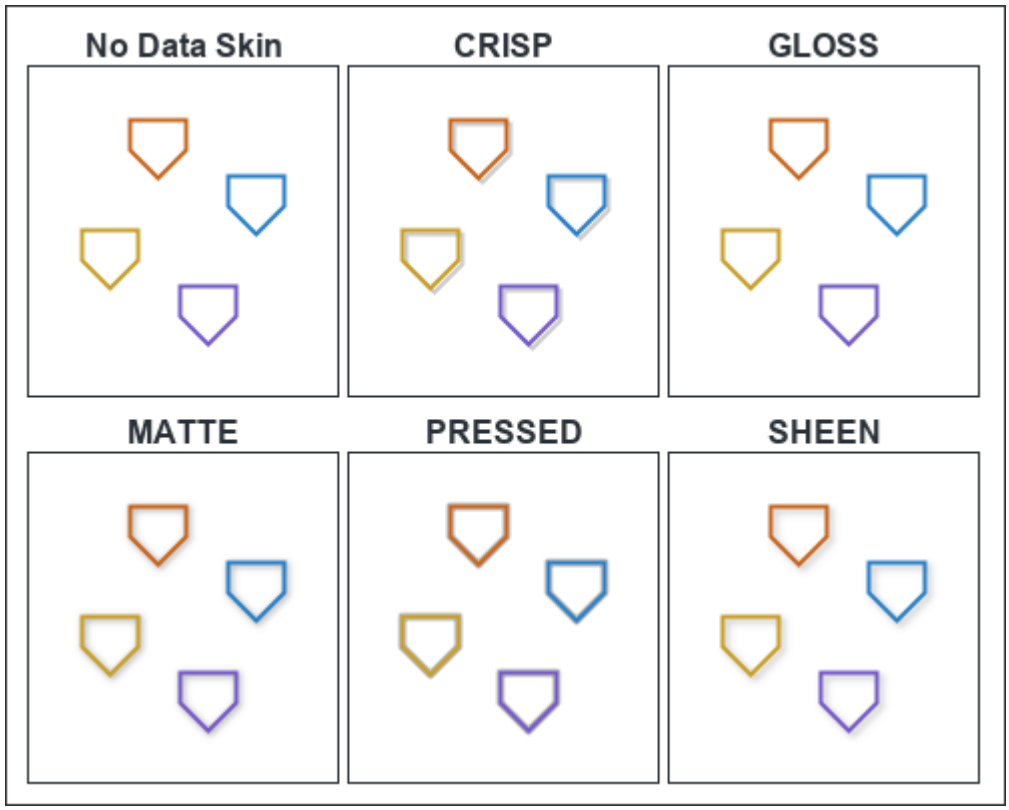
The following figure shows each of the data skins applied to unfilled bubbles.

Figure 15.3 *Data Skins Applied to Unfilled Bubbles*



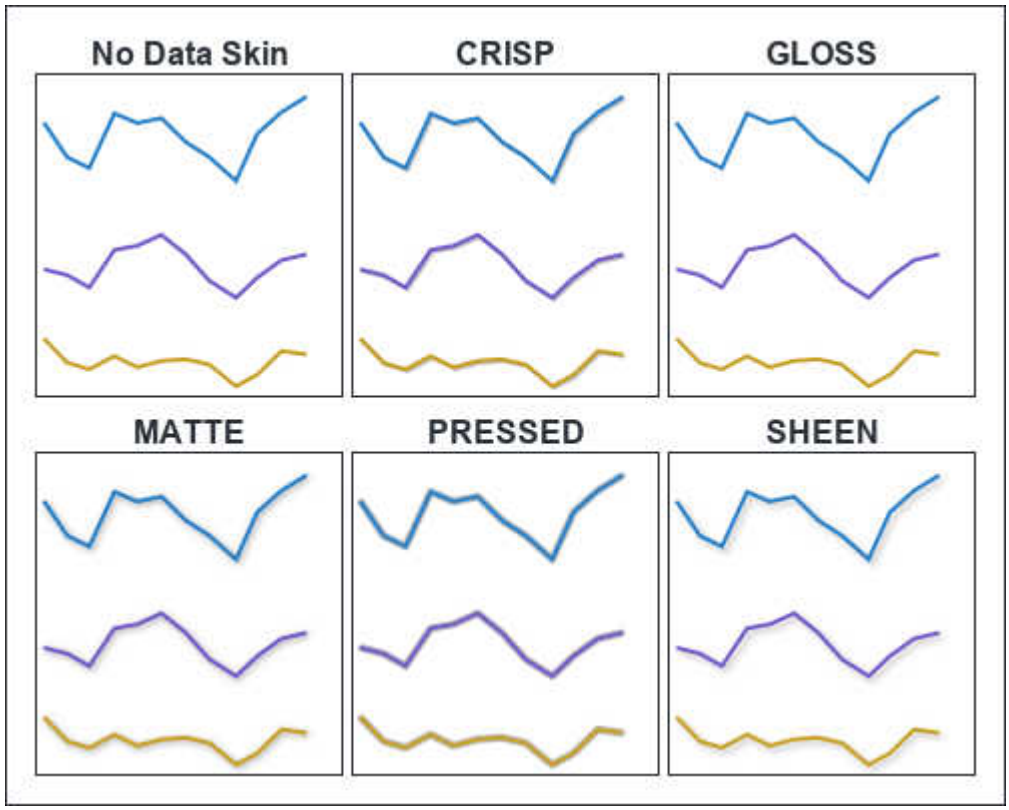
The following figure shows each of the data skins applied to unfilled markers.

Figure 15.4 *Data Skins Applied to HOMEDOWN Markers*



As with unfilled bubble and marker outlines, data skins affect the appearance of plot lines by adding subtle effects such as drop shadows. The following figure shows each of the data skins applied to plot lines.

Figure 15.5 *Data Skins Applied to Plot Lines*



For all plots that support data skins, the GraphSkin:DataSkin style element in the active style specifies by default the data skin that is applied. For an individual plot, you can use the DATASKIN= option in the plot statement to override the data skin that is specified by the current style. You can set the following values for the style GraphSkin:DataSkin element and the DATASKIN= options: NONE, SHEEN, GLOSS, PRESSED, CRISP, or MATTE.

In many cases, the maximum number of skinned graphical elements is limited to 200 per plot for performance reasons. For graphs that contain multiple plots, this limit applies to each plot and not to the entire graph. A skinned graphical element can be a bar, bubble, marker, series line, and so on. It does not necessarily correlate with the number of observations in the plot data. If this limit is exceeded for a plot, the specified data skin is not applied to that plot, and the following warning appears in the SAS log:

```
NOTE: Data skin has been disabled because the threshold has been
reached. You can set DATASKINMAX=nnn in the ODS GRAPHICS statement
to restore data skin.
```

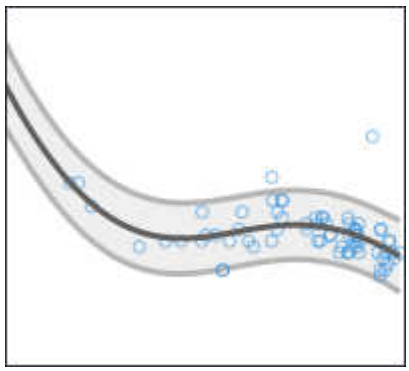
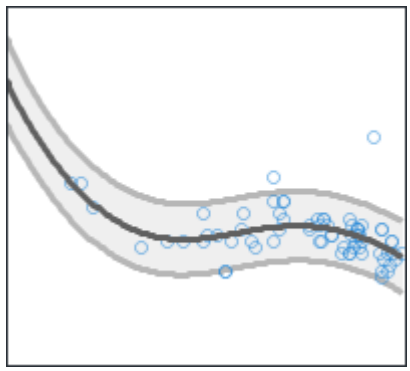
In that case, you can use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the threshold to the value specified in the note (*nnn*) or to a higher value.

Note: A plot that contains a large number of skinned graphical elements might take several minutes to render.

Subpixel Rendering

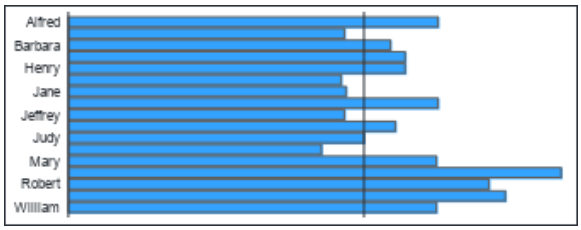
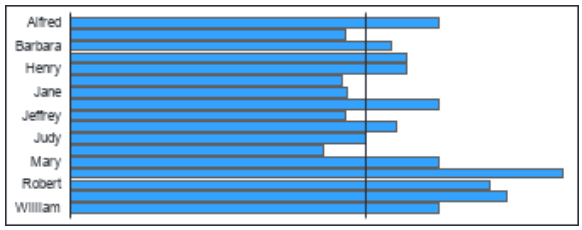
Subpixel rendering generates smooth curves and more precise spacing. Without subpixel rendering, curves can appear slightly jagged as shown in the following examples.

Table 15.7 *Effect of Subpixel Rendering on Curves*

With Subpixel Rendering	Without Subpixel Rendering
	

Subpixel rendering also has an effect on bar charts. When charts have a large number of bars that are very close together, slight variations in spacing between the bars that occur due to integer rounding can become more obvious. Subpixel rendering enables more precise bar spacing as shown in the following examples.

Table 15.8 *Effect of Subpixel Rendering on Spaces between Bars*

With Subpixel Rendering	Without Subpixel Rendering
	

Subpixel rendering is always enabled for the SGMAP, SGPIE, and SGSCATTER procedures. For the SGPANEL and SGPLOT procedures, you can use the SUBPIXEL or NOSUBPIXEL option in an ODS GRAPHICS statement or in the PROC SGPANEL and PROC SGPLOT statements to control subpixel rendering.

Note: Antialiasing must be enabled for the SUBPIXEL option to have any effect on the SGPANEL and SGPLOT procedures. Antialiasing is enabled by default. To re-enable antialiasing, use the ANTIALIAS=ON option in an ODS GRAPHICS statement.

If you specify SUBPIXEL or NOSUBPIXEL in an ODS GRAPHICS statement, that setting is used as the default for the SGPANEL and SGPLOT procedures. If neither option is specified in an ODS GRAPHICS statement, by default, the system applies subpixel rendering when it makes sense for the graph. You can override the default for subpixel rending for SGPANEL or SGPLOT by specifying the SUBPIXEL or NOSUBPIXEL option in the PROC SGPANEL or PROC SGPLOT statement.

Output for Grouped versus Non-Grouped Data

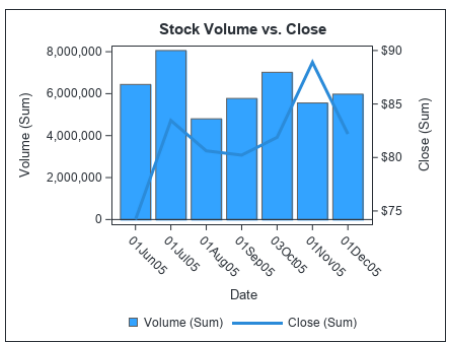
Non-Grouped Data

The Default Appearance of Non-Grouped Output

For the SGPLOT and SGPANEL procedures, many plots use the same style element, GraphDataDefault, for graphics elements such as lines, bars, and markers. These plots have the same default appearance.

In the graph that is produced by the following code, the bar chart and line chart use visual attributes from the GraphDataDefault element. The visual attributes for the line (and markers if specified) are designed to coordinate with the bar fill areas.

Table 15.9 *Default Appearance of Non-Grouped Bars and Lines*

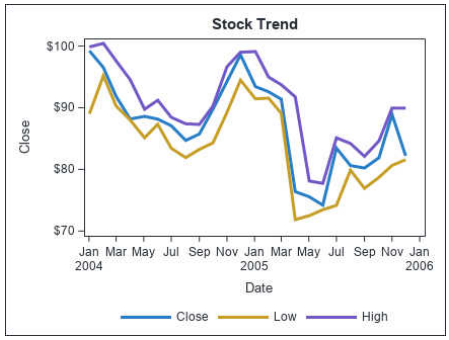



```
ods graphics on / width=4.5in;
proc sgplot data=sashelp.stocks
    (where=(date >= "01jun2005"d
     and stock = "IBM"));
   title "Stock Volume vs. Close";
   vbar date / response=volume;
   vline date / response=close y2axis;
run;
title;
ods graphics / reset=all;
```

However, when identical plot types are overlaid, the procedures instead use style elements named GraphData1 to GraphData*n* (where n=12 for most styles).

In the following graph, the series lines have different colors and are easy to distinguish. The plots use the GraphData1, GraphData2, and GraphData3 style elements.

Table 15.10 *Default Appearance When Three Series Plots Are Overlaid*




```
ods graphics on / width=4.5in;
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2004"d
    and stock = "IBM"));
  title "Stock Trend";
  series x=date y=close;
  series x=date y=low;
  series x=date y=high;
run;
title;
ods graphics / reset=all;
```

The GraphData*n* style elements use different marker symbols and fill colors to ensure that the plots differ in appearance. All line and marker colors are of different hues but with the same brightness, which means that all 12 colors can be distinguished but none stands out more than another. Fill colors are based on the same hue but have less saturation, making them similar but more muted than the corresponding contrast colors.

Line patterns can also vary depending on the style that is used for the graph. Some styles vary line patterns so that elements in color plots can be distinguished even when the plot is sent to a black-and-white printer. Line patterns do not vary for the Illuminate style or any style that is defined with the ATTRPRIORITY="Color" option.

Note: If you have a bar-line overlay with multiple lines, the lines cycle through GraphData*n*, whereas the bar remains GraphDataDefault.

Other plots use more specialized style elements for their appearance. For example, the line attributes for density, loess, PBSpline, and regression curves are determined by the GraphFit style element.

Here are some interactions for plots with specialized style elements:

- If you have a histogram with multiple density curves, the curve style pattern changes with the number of curves. For one curve, GraphFit is used. For two curves, GraphFit and GraphFit2 are used. For three or more, all curves switch over to GraphData*n* The histogram uses GraphDataDefault.
- If you have multiple fit plots (loess, regression, or PBSpline), the behavior is like the density curves. For one fit, GraphFit is used. For two fits, GraphFit and GraphFit2 are used. For three or more, all fits switch to GraphData*n*. (This behavior also applies to the SGSCATTER procedure when multiple fits are specified.)

- If multiple ellipses are overlaid on another plot (such as a scatter), the ellipses cycle through GraphData*n*, whereas the plot uses GraphDataDefault.

In general, if all of the plots have the same statement name (all scatter, all series, and so on), then the plots automatically cycle.

The CYCLEATTRS | NOCYCLEATTRS options in the SGPLOT and SGPANEL procedure statements can be used to manually control whether the plots have unique attributes. CYCLEATTRS forces cycling in situations where cycling is not automatic. The NOCYCLEATTRS option forces cycling to be off. These options are described in the next section.

Changing the Appearance of Non-Grouped Output

You can use plot options to specify particular GraphData*n* style elements for your plots in order to achieve a different appearance.

In the following graph, the style elements GraphData3, GraphData4, and GraphData5 are used to change the default appearance of the series lines in the graph.

Table 15.11 *Overlaid Series Plots with Particular GraphDatan Elements Specified*

```
ods graphics on / width=4.5in;
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2004"d
    and stock = "IBM"));
  title "Stock Trend";
  series x=date y=close
     / lineattrs= GraphData3;
  series x=date y=low
     / lineattrs= GraphData4;
  series x=date y=high
     / lineattrs= GraphData5;
run;
title;
ods graphics / reset=all;
```

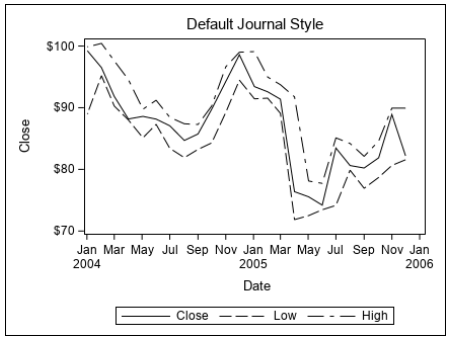
Note: To achieve a different appearance for the series lines, you can use any style element that has line properties. However, the GraphData*n* style elements are very suitable for this purpose. The GraphData*n* style elements have been carefully constructed with different hues but with the same brightness, which means that all 12 colors can be distinguished but none stands out more than another.

For more information, see "Specifying a Style Element" on page 1848.

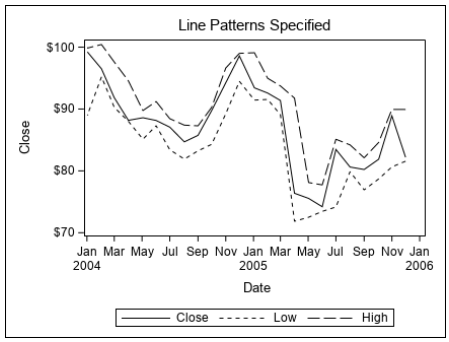
Sometimes, you might want to control one or more attributes of a style element. You can use plot options to override, or hardcode, an attribute in the style element.

The following two graphs show the same plots before and after the PATTERN= option is used. The PATTERN= option is used with two series lines to force a different line pattern.

Table 15.12 *The Effect of Specifying a Hardcoded Line Pattern*




```
ods html5 style=journal
             path="file-path" file="file-name";

proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2004"d
    and stock = "IBM"));
  title "Default Journal Style";
  series x=date y=close;
  series x=date y=low;
  series x=date y=high;
run;
title;

ods html5 close;
```



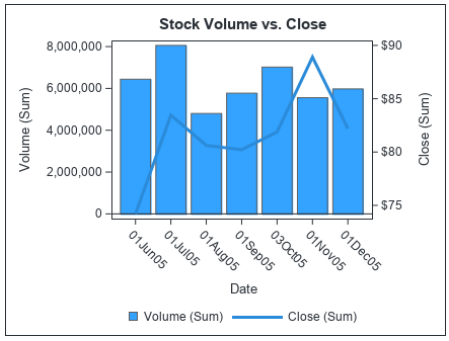

```
ods html5 style=journal
             path="file-path" file="file-name";
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2004"d
    and stock = "IBM"));
  title "Line Patterns Specified";
  series x=date y=close;
  series x=date y=low
     / lineattrs= (pattern=2);
  series x=date y=high
     / lineattrs= (pattern=4);
run;
title;

ods html5 close;
```

For more information, see "Specifying Hardcoded Values" on page 1849 and "Specifying a Style Element with Hardcoded Values" on page 1850.

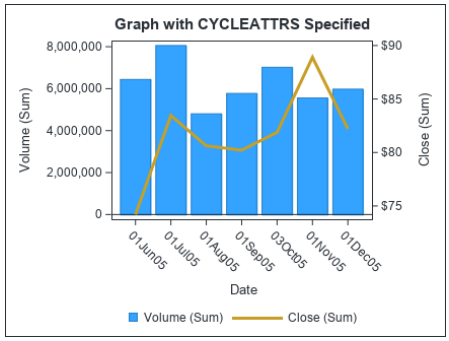
You can also use the CYCLEATTRS | NOCYCLEATTRS options in the SGPLOT and SGPANEL procedure statements to control whether the plots have unique attributes. CYCLEATTRS forces cycling in situations where cycling is not automatic. If the plots do not have unique attributes by default, then you can specify the CYCLEATTRS option to force unique attributes for each plot in the graph. The NOCYCLEATTRS option prevents the procedure from assigning unique attributes.

The following two graphs show the same plots before and after the CYCLEATTRS option is used.

Table 15.13 *The Effect of Specifying the CYCLEATTRS Option*




```
proc sgplot data=sashelp.stocks;
  where(date >= "01jun2005"d and
     stock = "IBM");
  title "Stock Volume vs. Close";
  vbar date / response=volume;
  vline date / response=close y2axis;
run;
title;
```




```
proc sgplot data=sashelp.stocks
    CYCLEATTRS ;
  where(date >= "01jun2005"d and
     stock = "IBM");
  title "Graph with CYCLEATTRS Specified";
  vbar date / response=volume;
  vline date / response=close y2axis;
run;
title;
```

In the second graph, the procedure retrieves the attributes from GraphData1 and GraphData2. The first graph retrieves all of its data attributes from GraphDataDefault.

Note: Depending on the size of your graph, the output might not look identical to what is shown here. See "About the Examples in This Book" on page 87.

Grouped Data

The Appearance of Grouped Output

The GROUP= option is used to plot data when a classification or grouping variable is available. By default, this option automatically uses the GraphData*n* style elements for the presentation of each unique group value.

You can specify the color, line, and marker properties for group values directly. Appearance options override the corresponding defaults from the current style. These appearance options affect only the procedure in which they are specified.

To specify appearance options for a procedure, do the following:

- In the SGPLOT and SGPANEL procedures, use the STYLEATTRS statement. Specify the DATACOLORS=, DATACONTRASTCOLORS=, DATALINEPATTERNS=, and DATASYMBOLS= options.
- Specify the DATACOLORS=, DATACONTRASTCOLORS=, DATALINEPATTERNS=, and DATASYMBOLS= options in the PROC SGSCATTER statement.

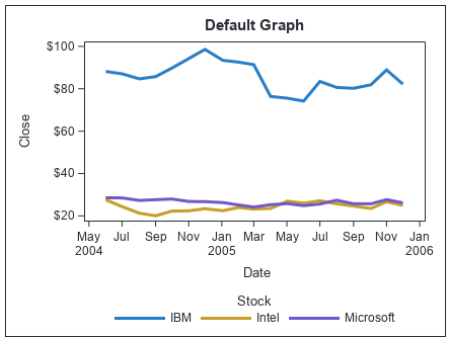
You can also use attribute maps to specify particular visual attributes for group values of the data. Attribute mapping does not require the input data set to be ordered by the group variable. Attribute mapping is explained in the following section.

Making the Appearance of Grouped Data Independent of Data Order

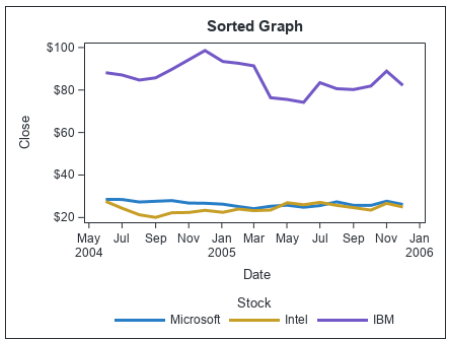
When unique group values are gathered, they are internally recorded in the order in which they appear in the data. They are not subsequently sorted. If an input data source is modified, sorted, or filtered, the order of the group values and their associations with GraphData*n* might change.

The following two graphs show the same plots before and after the data is sorted.

Table 15.14 *The Effect That Sorting Has on Grouped Plot Attributes*




```
ods graphics on / width=4.5in;
proc sgplot data=sashelp.stocks;
  where date >= "01jun2004"d;
  title "Default Graph";
  series x=date y=close / group= stock;
run;
title;
```



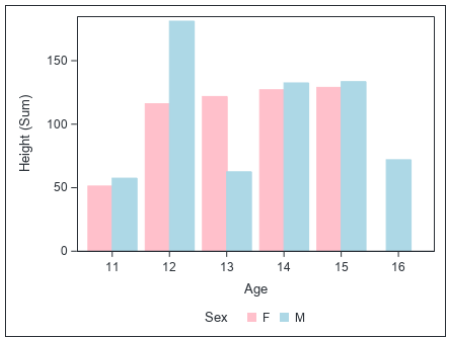

```
proc sort data=sashelp.stocks out=stocks;
  by descending stock;
run;

proc sgplot data=stocks;
  where date >= "01jun2004"d;
  title "Sorted Graph";
  series x=date y=close / group= stock;
run;
title;
```

Because the position of the data in the sorted data set has changed, the sorted graph uses different GraphData*n* style elements for the group values.

In many cases, this might not be a problem if you do not care which line pattern, marker symbols, or colors are associated with particular group values. However, in some cases you might care. For example, if you create many plots grouped by gender, you might want a consistent set of visual properties for females and males across plots, regardless of the input data order.

SG attribute maps enable you to ensure that particular visual attributes are applied to the data independent of data order.

The following output shows a bar chart that uses the MYATTRMAP discrete attribute map data set for its pink and blue bar colors. The FILLCOLOR= values determine the color of the bars, and the LINECOLOR= values determine the color of the border edges around the bars.




```
data myattrmap;
length linecolor $ 9 fillcolor $ 9;
input ID $ value $ linecolor $ fillcolor $;
datalines;
myid  F pink pink
myid  M lightblue lightblue
;
run;
proc sgplot data=sashelp.class
   dattrmap=myattrmap;
 vbar age / response=height group=sex
   groupdisplay=cluster attrid=myid;
run;
```

For complete information about attribute maps, see Chapter 17, “Using Attribute Maps to Control Visual Attributes,” on page 1897.

Modifying Style Templates

Style templates are created and modified with the TEMPLATE procedure. You can use the DEFINE STYLE statement to create your own style from scratch or from an existing style. When you create styles from existing styles, you can modify the individual style elements.

For example, the following program shows the style element GraphDataDefault, as defined in the Default style:

```
proc template;
   define style Styles.Default;

... more style elements ...

 class GraphDataDefault /
        endcolor = GraphColors("gramp3cend")
        neutralcolor = GraphColors("gramp3cneutral")
        startcolor = GraphColors("gramp3cstart")
        markersize = 7px
```

```
            markersymbol = "circle"
            linethickness = 1px
            linestyle = 1
            contrastcolor = GraphColors("gcdata")
            color = GraphColors("gdata");
```

You can use the DEFINE STYLE statement to create a new style from the Default style and modify the GraphDataDefault style element.

The following program creates the new style MyStyleDefault, which inherits all of its style elements and style attributes from the Default style, and customizes the GraphDataDefault style element:

```
proc template;
   define style MyStyleDefault;
   parent=Styles.Default;
   style GraphDataDefault from GraphDataDefault /
         markersize = 10px
         markersymbol = "square";
   end;
run;
```

For complete documentation on using PROC TEMPLATE to modify and create styles, see the *SAS Output Delivery System: Procedures Guide*.

Using Fill Patterns to Distinguish Categories of Data

About Using Fill Patterns

Fill patterns help distinguish between different categories of data and are useful with group data. The procedures provide two ways to display fill patterns:

- use plot options to display and control the appearance of fill patterns. These options are available with various plots and charts.
- use SAS gray-scale styles to display fill patterns for grouped bar charts.

Accessibility note: This feature helps support accessibility. Graphs that use this feature do not rely on color alone to distinguish categories of data.

Using Plot Options to Display Fill Patterns

Plot Options That Display Fill Patterns

The FILLPATTERN and FILLPATTERNATTRS= options are available with a number of plots and charts. These options enable you to display and control the appearance of fill patterns in the plot.

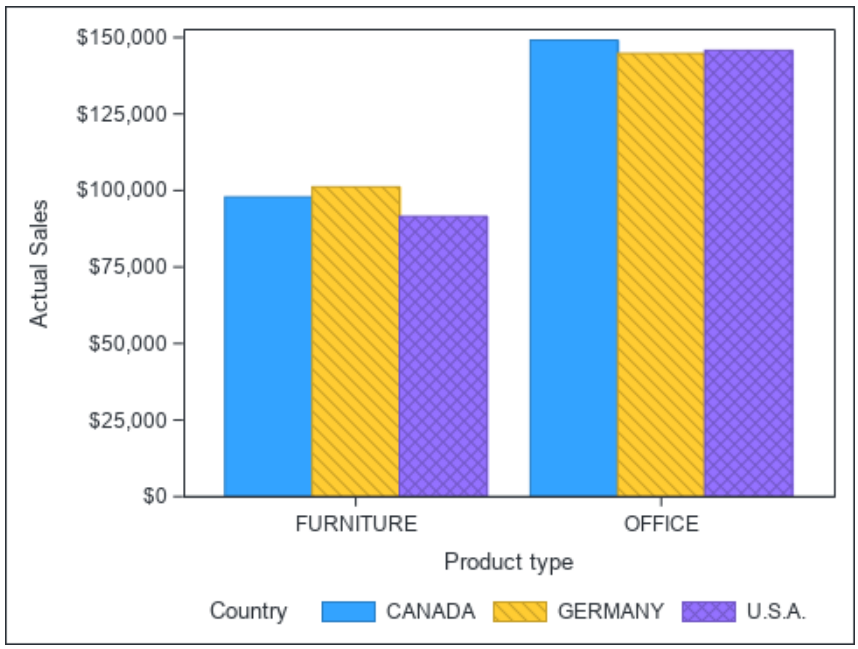
The options are available with the following types of plots, whether generated using the SGPLOT procedure or the SGPANEL procedure:

BAND	HBARBASIC	POLYGON
BUBBLE	HBARPARM	VBAR
ELLIPSE	HBOX	VBARBASIC
ELLIPSEPARM	HIGHLOW	VBARPARM
HBAR	HISTOGRAM	VBOX

Adding Fill Patterns to a Grouped Plot

Here is example of a grouped vertical bar chart with fill patterns:

Figure 15.6 *Fill Pattern for Grouped Bars*



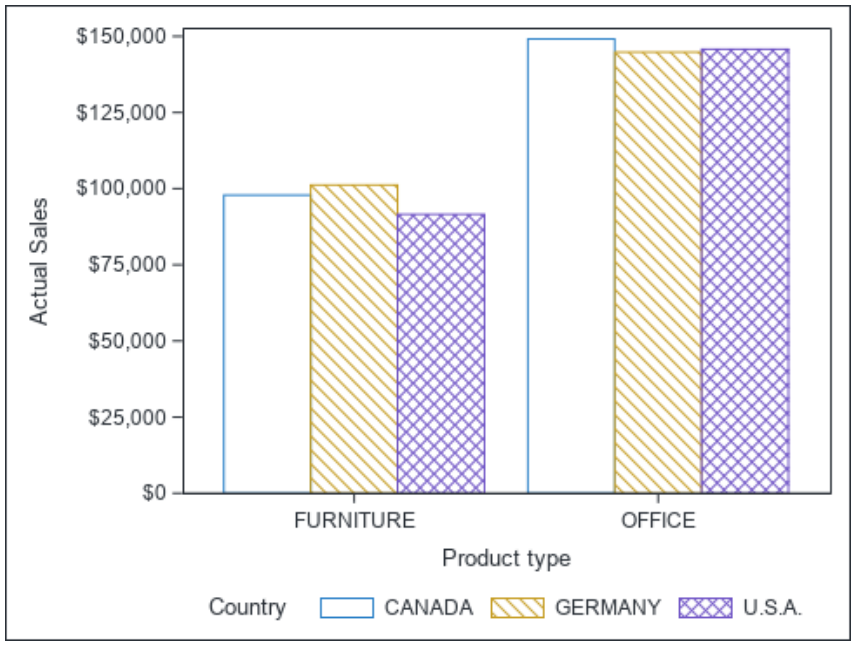
The following code was used to produce the example output.

```
ods graphics / attrpriority=none;
proc sgplot data=sashelp.prdsale;
  vbar prodtype / response=actual
     group=country groupdisplay=cluster
     fillpattern;
run;
```

Note: When you specify a group, depending on the style that is applied to the graph, color might be used to distinguish the group values. To ensure that group values are distinguished by their fill patterns, use the ATTRPRIORITY option in the ODS GRAPHICS statement as shown in the previous example.

You can use the plot's NOFILL option to suppress the fill color, which might make it easier to see the fill patterns.

Figure 15.7 *Fill Pattern without Fill Color*



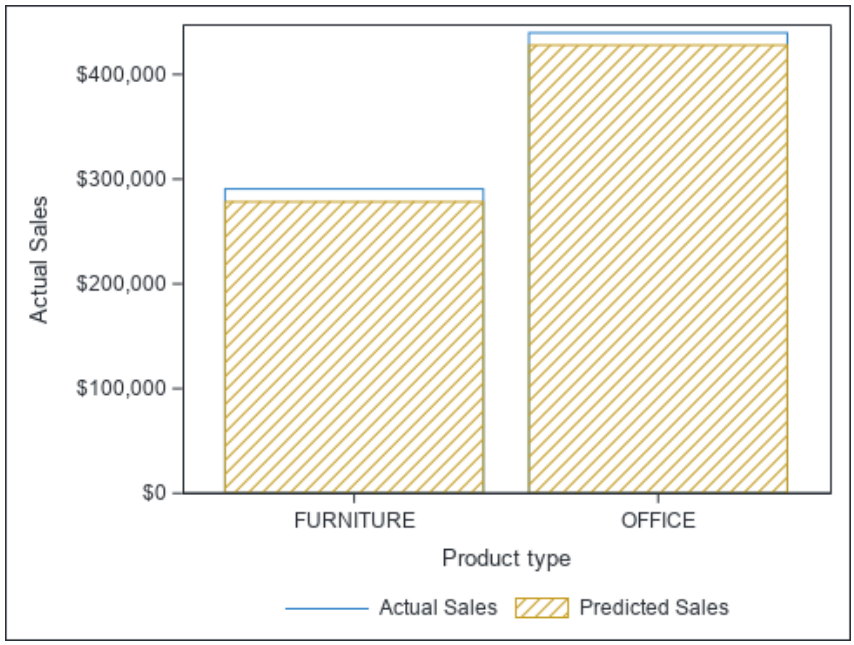
The following code was used to produce the example output.

```
ods graphics / attrpriority=none;
proc sgplot data=sashelp.prdsale;
  vbar prodtype / response=actual
     group=country groupdisplay=cluster
     fillpattern nofill;
run;
```

Specifying a Single Fill Pattern for a Plot

In addition to the FILLPATTERN option, the FILLPATTERNATTRS= enables you to specify the color and pattern for a plot's fill pattern.

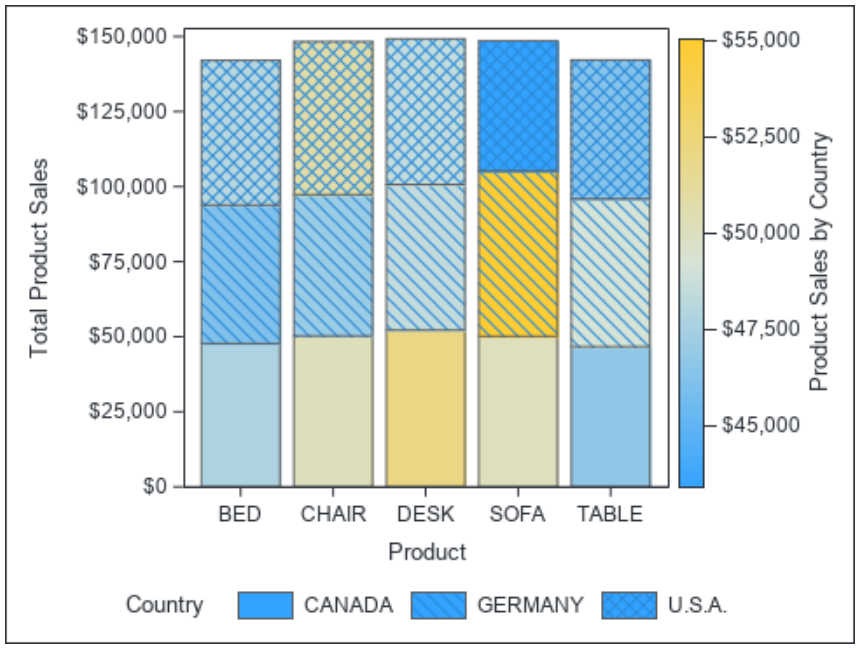
Figure 15.8 *Fill Pattern with a Specified Line Pattern*




```
proc sgplot data=sashelp.prdsale;
  vbar prodtype / response=actual
      nofill;
  vbar prodtype / response=predict
      nofill fillpattern
      fillpatternattrs=(pattern=r1);
run;
```

Using Fill Patterns and Gradient-Color Fills

In a grouped plot that uses gradient-color fills, you can add fill patterns to distinguish the group values by using the FILLPATTERN option in the plot statement. However, when the COLORRESPONSE= option is used with the GROUP= option in a plot statement, only the gradient legend is automatically generated. In that case, to show both legends, you must include a KEYLEGEND statement and a GRADLEGEND statement in the procedure step.




```
ods graphics / attrpriority=none;
proc sgplot data=sashelp.prdsale;
  vbar product / name="bar" response=actual colorresponse=actual
     group=country fillpattern;
  yaxis label="Total Product Sales";
  keylegend "bar" / scale=1.2 title="Country";
  gradlegend "bar" / title="Product Sales by Country";
run;
```

With this method, the GraphDataDefault:contrastColor attribute determines the color of the fill patterns in the plot, and the GraphDataDefault:color attribute determines the fill color of the fill pattern swatches in the discrete legend. You can use the FILLPATTERNATTRS= option to change the fill pattern color, if desired.

An alternative method is to use an overlay plot to overlay group fill patterns onto a color-gradient plot.

```
ods graphics / attrpriority=none;
proc sgplot data=sashelp.prdsale nocycleattrs;
  vbar product / name="bar" response=actual colorresponse=actual
     group=country;
  vbar product / name="fillpatterns" response=actual group=country
     nofill fillpattern fillpatternattrs=(color=black) transparency=0.7
     outlineattrs=(color=black);
  yaxis label="Total Product Sales";
  keylegend "fillpatterns" / scale=1.2 title="Country";
  gradlegend "bar" / title="Product Sales by Country";
run;
```

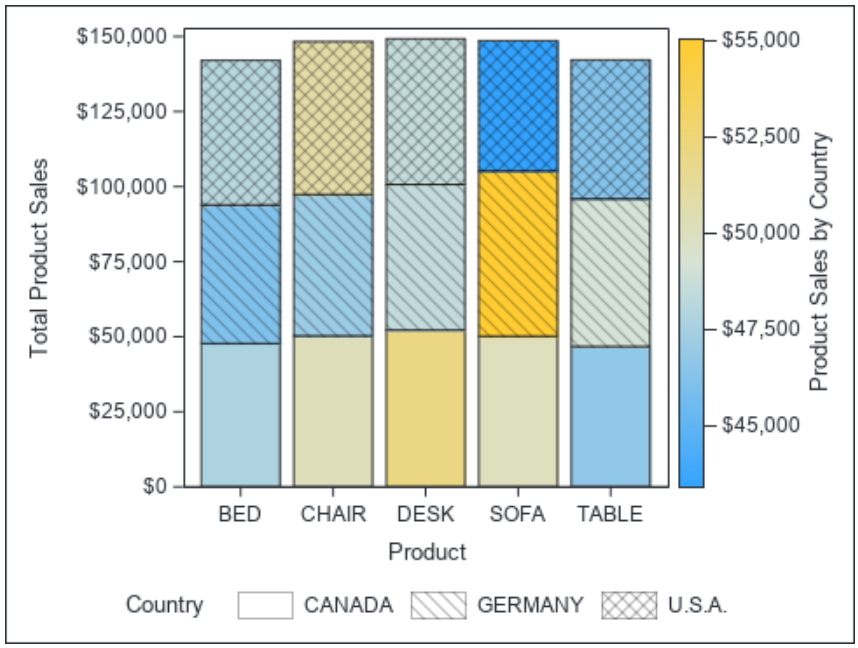
In this example, the NOCYCLEATTRS option is specified in the PROC SGPLOT statement to ensure that the overlay plot uses the same visual attributes as the primary plot when not specified in the overlay plot statement. The second VBAR statement overlays the group fill patterns onto the first VBAR plot. In the overlay plot, fill color is suppressed to enable the underlying bar colors to show through and to remove the fill color from the swatches in the discrete legend. Fill patterns are specified instead. Options specify the attributes of the fill patterns and

outlines. This method enables more control over the appearance of the fill patterns in the plot and the fill-pattern swatches in the discrete legend.

With this method, both legends are generated automatically. You do not need to include the KEYLEGEND and GRADLEGEND statements in the procedure step unless you want to customize one or both of the legends as is done in this example.

Note: If you customize only one of the legends, you must include a KEYLEGEND statement and a GRADLEGEND statement in the procedure step. If you include only one legend statement, only that legend is rendered in the graph.

Here is the result.



Using Gray-Scale Styles to Distinguish Grouped Bar Charts

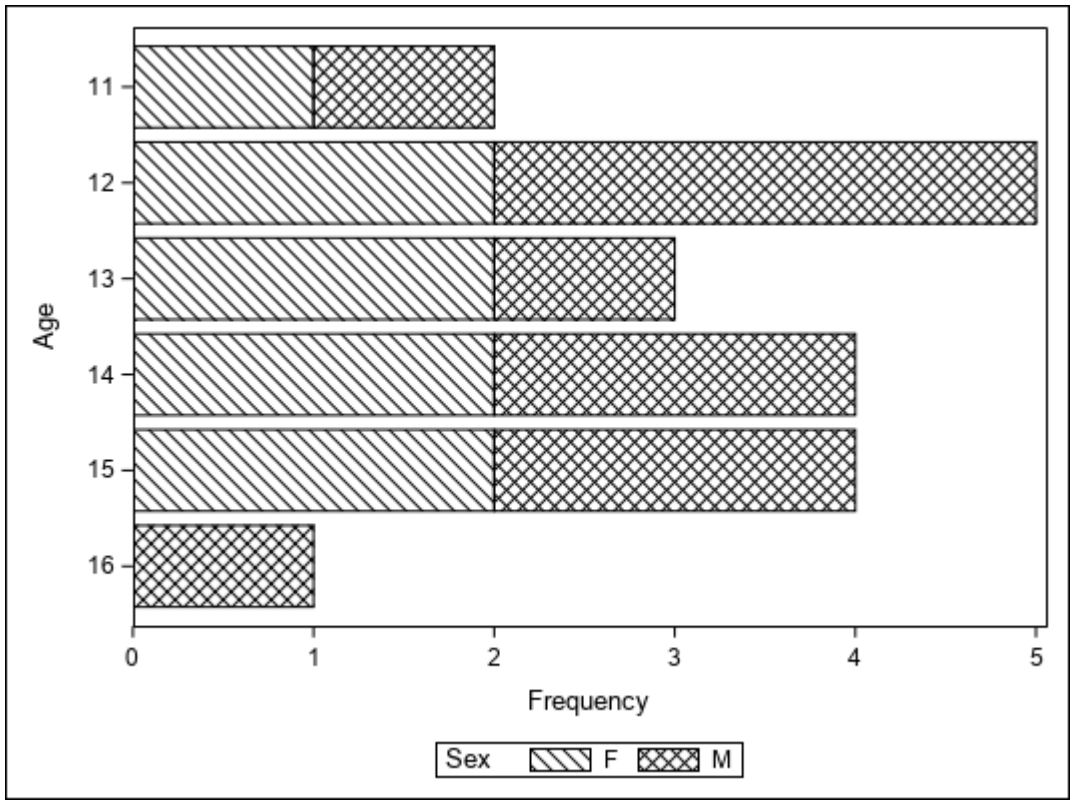
Some SAS styles automatically display fill patterns for grouped bars. Fill patterns help distinguish between different categories when you are producing or printing black and white output.

The fill patterns appear for grouped data in the following types of plots, whether generated using the SGPLOT procedure or the SGPANEL procedure:

HBAR	HBARPARM	VBARBASIC
HBARBASIC	VBAR	VBARPARM

Here is example output that was generated using the JOURNAL2 style.

Figure 15.9 *Fill Pattern for Grouped Bars Using the JOURNAL2 Style*



Here are the SAS styles that by default render grouped bars with fill patterns:

- JOURNAL2
- JOURNAL3 (uses gray and the fill pattern)

The following code was used to produce the example output.

```
proc sgplot data=sashelp.class;
  hbar age / group= sex;
run;
```

The JOURNAL2 or JOURNAL3 styles use the GRAPHBAR style element for the fill pattern. If you customize a style in the GTL, then you can add the GRAPHBAR style element to your custom style.

Style Elements for Use with ODS Graphics

Style elements affect ODS graphics and can be specified by GTL appearance options or used in styles. Attribute values can be changed with the TEMPLATE procedure.

In the ODS Graphics procedures, you can specify style elements to override the default appearance of particular graphics elements, such as data labels. Certain style elements were created to be used with specific plots. For example, the style element GraphFit2 is best used to modify secondary fit lines. The style element GraphConfidence2 was created to modify secondary confidence bands. However, many plots provide the option to override the style element with a different style element, or to specify particular attributes for a style element.

Here are examples that specify a style element and an attribute of a style element:

- GraphData2 is an example of a style element.
- GraphData2:Color is an example of a style attribute.

The tables below list each style element, the portion of the graph that it affects, and the applicable attributes.

Table 15.15 *Graph Style Elements: General Graph Appearance*

Style Element	Portion of Graph Affected	Recognized Attributes
GraphAnnoLine	Annotation lines and arrows, and outlines for closed annotation shapes such as circles and squares.	ContrastColor LineStyle LineThickness
GraphAnnoShape	Fill for closed annotation shapes such as circles and squares.	Color MarkerSize MarkerSymbol Transparency
GraphAnnoText	Annotation text	Color Font or *font-attributes*[1]
GraphAxisLines	X-, Y-, and Z-axis lines	ContrastColor LineStyle LineThickness TickDisplay
GraphBackground	Background of the graph	Color Transparency
GraphBaseMap	Appearance of the tiles in the base map that is generated by the SGMAP procedure OPENSTREETMAP statement.	TileSet[2]

Style Element	Portion of Graph Affected	Recognized Attributes
GraphBorderLines	Border line around the graph and around the graph legend	ContrastColor LineStyle LineThickness
GraphDataText	Text font and color for point and line labels	Color Font or *font-attributes*[1]
GraphFootnoteText	Text font and color for footnote(s)	Color Font or *font-attributes*[1]
GraphGridLines	Horizontal and vertical grid lines drawn at major tick marks	ContrastColor DisplayOpts LineStyle LineThickness Transparency
GraphHeaderBackground	Background color of the cell headers in a data-driven lattice or data-driven panel graph	Color Transparency
GraphLabelText	Text font and color for axis labels and legend titles	Color Font or *font-attributes*[1]
GraphLegendBackground	Background color of the legend	Color Transparency
GraphMinorGridLines	Appearance of the horizontal and vertical minor grid lines.	ContrastColor DisplayOpts LineStyle LineThickness
GraphOutlines	Outline properties for fill areas such as bars, pie slices, box plots, ellipses, and histograms	Color ContrastColor LineStyle LineThickness

Style Element	Portion of Graph Affected	Recognized Attributes
GraphReference	Horizontal and vertical reference lines and drop lines	ContrastColor LineStyle LineThickness
GraphTitleText	Text font and color for title(s)	Color Font or *font-attributes*[1]
GraphUnicodeText	Text font for Unicode values	Color Font or *font-attributes*[1]
GraphValueText	Text font and color for axis tick values and legend values	Color Font or *font-attributes*[1]
GraphWalls	Vertical wall(s) bounded by axes	Color ContrastColor FrameBorder LineStyle LineThickness

1 *Font-attributes* can be one or more of the following: FONTFAMILY=, FONTSIZE=, FONTSTYLE=, FONTWEIGHT=.

2 *TileSet* can be LIGHT, DARK, or HIGHCONTRAST. The default is LIGHT. See "Controlling the Theme of an OpenStreetMap Base Map" in *SAS/GRAPH and Base SAS: Mapping Reference* for details.

Table 15.16 *Style Elements Affecting Graphical Data Representation*

Style Element	Portion of Graph Affected	Recognized Attributes
GraphBoxMean	Marker for mean	ContrastColor MarkerSize MarkerSymbol
GraphBoxMedian	Line for median	ContrastColor LineStyle LineThickness
GraphBoxWhisker	Box whiskers and serifs	ConstrastColor LineStyle LineThickness

Style Element	Portion of Graph Affected	Recognized Attributes
GraphConfidence	Primary confidence lines and bands, colors for bands and lines	ContrastColor Color LineStyle LineThickness MarkerSize MarkerSymbol Transparency
GraphConfidence2	Secondary confidence lines and bands, color for bands, and contrast color for lines	ContrastColor Color LineStyle LineThickness MarkerSize MarkerSymbol Transparency
GraphConnectLine	Line for connecting boxes or bars	ContrastColor LineStyle LineThickness
GraphCutLine	Cutline attributes for a dendogram	Color LineStyle
GraphDataDefault	Primitives related to non-grouped data items, colors for filled areas, markers, and lines	Color ContrastColor EndColor LineStyle LineThickness NeutralColor MarkerSymbol MarkerSize StartColor
GraphError	Error line or error bar fill, ContrastColor for lines, Color for bar fill	CapStyle Color ContrastColor LineStyle Transparency

Style Element	Portion of Graph Affected	Recognized Attributes
GraphFit	Primary fit lines such as a normal density curve	Color ContrastColor LineStyle LineThickness MarkerSize MarkerSymbol
GraphFit2	Secondary fit lines such as a kernel density curve	Color ContrastColor LineStyle LineThickness MarkerSize MarkerSymbol
GraphFinal	Final data for the waterfall chart. Color applies to filled areas.	Color ContrastColor LineStyle LineThickness MarkerSize MarkerSymbol
GraphInitial	Initial data for the waterfall chart. Color applies to filled areas.	Color ContrastColor LineStyle LineThickness MarkerSize MarkerSymbol
GraphMissing	Properties for graph items representing missing values	Color ContrastColor FillPattern LineStyle LineThickness MarkerSymbol MarkerSize Transparency

Style Element	Portion of Graph Affected	Recognized Attributes
GraphOther	Other data for the graph. Color applies to filled areas.	Color ContrastColor FillPattern LineStyle LineThickness MarkerSize MarkerSymbol
GraphOverflow	Overflow data for the graph. Color applies to filled areas. ContrastColor applies to markers and lines.	Color ContrastColor LineStyle LineThickness MarkerSize MarkerSymbol
GraphOutlier	Outlier data for the graph	ContrastColor Color MarkerSize MarkerSymbol LineStyle LineThickness
GraphPrediction	Prediction lines	ContrastColor Color LineStyle LineThickness MarkerSize MarkerSymbol Transparency
GraphPredictionLimits	Fills for prediction limits	ContrastColor Color LineStyle LineThickness MarkerSize MarkerSymbol Transparency

Style Element	Portion of Graph Affected	Recognized Attributes
GraphSelection	For interactive graphs, visual properties of selected item. Color for selected fill area, ContrastColor for selected marker or line.	ContrastColor Color LineStyle LineThickness MarkerSymbol MarkerSize
GraphUnderflow	Underflow data for the graph. Color applies to filled areas. ContrastColor applies to markers and lines.	Color ContrastColor LineStyle LineThickness MarkerSize MarkerSymbol
ThreeColorAltRamp	Line contours, markers, and data labels with segmented range color response	EndColor NeutralColor StartColor
ThreeColorRamp	Gradient contours, surfaces, markers, and data labels with continuous color response	EndColor NeutralColor StartColor
TwoColorAltRamp	Line contours, markers, and data labels with segmented range color response	EndColor StartColor
TwoColorRamp	Gradient contours, surfaces, markers, and data labels with continuous color response	EndColor StartColor

Table 15.17 *Graphical Style Elements: Data Related (Grouped)*

Style Elements	Portion of Graph Affected	Attributes Defined
GraphData1 GraphData2 GraphData3 GraphData4 GraphData5 GraphData6	Primitives related to the first seven grouped data items. Color applies to filled areas. ContrastColor applies to markers and lines.	Color ContrastColor FillPattern[1] LineStyle MarkerSymbol

Style Elements	Portion of Graph Affected	Attributes Defined
GraphData7		
GraphData8 GraphData9 GraphData10 GraphData11	Primitives related to the 8th through 11th grouped data items.	Color ContrastColor FillPattern[1] LineStyle MarkerSymbol[2]
GraphData12	Primitives related to the 12th grouped data item.	Color ContrastColor FillPattern[1] MarkerSymbol[2]
GraphData13[3] GraphData14 GraphData15	Primitives related to the 13th through 15th grouped data items.	FillPattern[1] MarkerSymbol

1 Style attribute FillPattern in GraphData1–GraphData11 is also available with the DEFAULT ODS style and all styles that are derived from it.
2 Style attribute MarkerSymbol in these style elements is defined for styles JOURNAL2 and JOURNAL3 only.
3 Style elements GraphData13–GraphData15 are available only with the JOURNAL2 and JOURNAL3 styles.

Table 15.18 *Display Style Elements*

Style Element	Portion of Graph Affected	Recognized Attributes	Possible Values
GraphAltBlock	Alternate fill color for block plots	Color	GraphColors("gablock")
GraphBand	Display options for confidence bands	DisplayOpts	"Fill FillPattern Outline"
GraphBar	Display options for bar charts	DisplayOpts	"Connect Fill FillPattern Outline"
GraphBlock	Fill color for block plots	Color	GraphColors("gblock")
GraphBox	Display options for box plots	DisplayOpts	"Caps Connect Fill FillPattern Mean Median Notches Outliers"
		CapStyle	"Serif"

Style Element	Portion of Graph Affected	Recognized Attributes	Possible Values
		Connect	"Mean"
GraphContour	Display options for contours	DisplayOpts	"LabeledLineGradient"
		EndColor	GraphColors('gramp3cend')
		NeutralColor	GraphColors('gramp3cneutral')
		StartColor	GraphColors('gramp3cstart')
GraphDiscreteAxis[1, 2]	Display options for discrete axes	DisplayOpts	"Grid Label Line Ticks TickValues"
GraphEllipse	Display options for confidence ellipses	DisplayOpts	"Fill FillPattern Outline"
GraphHighLow	Display options for high-low plots	DisplayOpts	"Fill FillPattern Outline"
GraphHistogram	Display options for histograms	DisplayOpts	"Fill FillPattern Outline"
GraphLinearAxis[1, 2]	Display options for linear axes	DisplayOpts	"Grid Label Line Ticks TickValues"
GraphLogAxis[1, 2]	Display options for log axes	DisplayOpts	"Grid Label Line Ticks TickValues"
GraphPolygon	Display options for polygon plots	DisplayOpts	"Fill FillPattern Outline"
GraphSkins	Display skins	DataSkin	"Crisp" "Gloss" "Matte" "None" "Pressed"
		KpiSkin	"Basic" "Modern" "None" "Onyx"

Style Element	Portion of Graph Affected	Recognized Attributes	Possible Values
			"Satin"
GraphTimeAxis[1, 2]	Display options for time axes	DisplayOpts	"Grid Label Line Ticks TickValues"

1 Valid starting with 2025.06. This style element is defined in the updated ODS styles for 2025. You can customize any existing ODS style to define this style element. For an example, see "Axis Features in the Panel Graphs" on page 1705. For information about ODS styles, see *SAS Output Delivery System: User's Guide*.

2 This style element is not honored by all of the procedures that support ODS Graphics.

16

Managing Your Graphics with ODS

Introduction to Managing Your Graphics with ODS **1884**

Specifying ODS Destinations **1884**
See Also 1885

ODS Destination Statement Options Affecting ODS Graphics **1885**
See Also 1886

Using the ODS GRAPHICS Statement **1886**
Introduction to Using the ODS GRAPHICS Statement 1886
Sample Program That Uses the ODS GRAPHICS Statement 1887
See Also 1888

Controlling the Size of Your Graph **1888**
Overview of Controlling the Size of Your Graph 1888
Specifying Output Size with the ODS GRAPHICS Statement 1889
Retaining the Aspect Ratio 1889
Graph Scaling for User-Specified Image Size 1890
Tip for Embedding Graph Images in Documents 1890

Controlling Image Resolution **1891**
See Also 1891

Adding Data Tips and Other Features **1891**

Controlling the Image Name and Image Format **1892**
See Also 1892

Resetting ODS GRAPHICS Options **1893**

Introduction to Managing Your Graphics with ODS

The Output Delivery System (ODS) manages all output created by procedures and enables you to display the output in a variety of formats, such as HTML, PDF, and RTF. The SAS ODS Graphics procedures and many SAS Analytical procedures use ODS Graphics for creation of their graphs.

ODS provides options for controlling many relevant features using the following statements:

- ODS destination statements. For more information, see "ODS Destination Statement Options Affecting ODS Graphics" on page 1885.
- the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

Specifying ODS Destinations

For the creation of ODS graphics, a valid ODS destination must be active. ODS can send your graphs to a number of different destinations. Your IDE opens an ODS destination by default, typically ODS HTML5. If you keep the default destination open and open another, the resultant output is sent to all open destinations.

As an alternative, you can specify an ODS destination in your program. The ODS destination statement is used at the beginning and end of the program to open and close destinations.

For example, the following statements open and close an ODS POWERPOINT destination.

```
ods powerpoint file="file-specification";     /* opens the destination */
    /* procedure statements and other program elements here */
ods powerpoint close; /* closes the destination */
```

You can use ODS destination statement options to control aspects of ODS graphic output. Depending on the options available for the destination, you can specify options such as the file name or the path to an output directory.

These options are specified in ODS destination statements using the following form:

ODS *destination* <*options*;>

For example, you can use ODS destination statements to control the following:

- the style that is applied to a graph. For more information, see "Specifying Styles" on page 1847.
- the image resolution in dots per inch. For more information, see "Controlling Image Resolution" on page 1891.

See Also

"Supported Image Formats for Output Destinations" on page 127

ODS Destination Statement Options Affecting ODS Graphics

Each ODS destination has options that govern aspects of your ODS Graphics output. The following table shows the options for the most commonly used destinations.

Table 16.1 *ODS Destination Options That Affect ODS Graphics*

ODS Destination	Options for ODS Graphics	Description
PDF		Creates an embedded image or images in a PDF document. Native PDF is the default image format.
	DPI=*number*	Specifies the image resolution in dots per inch for output images. DPI=150 is the default.
	STYLE= *style-definition*	Specifies the style to use. STYLE=PEARL is the default.
RTF		Creates an embedded image or images in an RTF document. The default image format is EMF.
	IMAGE_DPI=*number*	Specifies the image resolution in dots per inch for output images. IMAGE_DPI=200 is the default.
	STYLE= *style-definition*	Specifies the style to use. STYLE=RTF is the default.

ODS Destination	Options for ODS Graphics	Description
HTML5		Creates a standalone image or images and the HTML5 page. Images are referenced in the HTML page. The default image format is SVG.
	PATH="*directory-spec*"	Indicates the directory where HTML files and graph images are created.
	IMAGE_DPI=*number*	Specifies the image resolution in dots per inch for output images. IMAGE_DPI=96 is the default.
	STYLE= *style-definition*	Specifies the style to use. The default style is HTMLEncore. **Note:** If your IDE opens ODS HTML5 by default, it might be configured to use a different ODS style.

See Also

"Understanding ODS Destinations" in *SAS Output Delivery System: User's Guide*

Using the ODS GRAPHICS Statement

Introduction to Using the ODS GRAPHICS Statement

You can use the ODS GRAPHICS statement options to control many aspects of your graphics. The settings that you specify remain in effect for all graphics in the current session until you change or reset these settings with another ODS GRAPHICS statement.

ODS GRAPHICS is ON by default.

You can use the ODS GRAPHICS statement to control many features, including the following:

- the size of the image
- the type and name of the image created
- whether features such as data tips, scaling, and anti-aliasing are used

The basic syntax for the ODS GRAPHICS statement is as follows:

ODS GRAPHICS < OFF | ON> </ *options*>;

Note: For the complete statement syntax, see "ODS GRAPHICS Statement" on page 102.

Sample Program That Uses the ODS GRAPHICS Statement

The following program uses the ODS GRAPHICS statement to illustrate several techniques for controlling your graphics.

These techniques are discussed in the following sections:

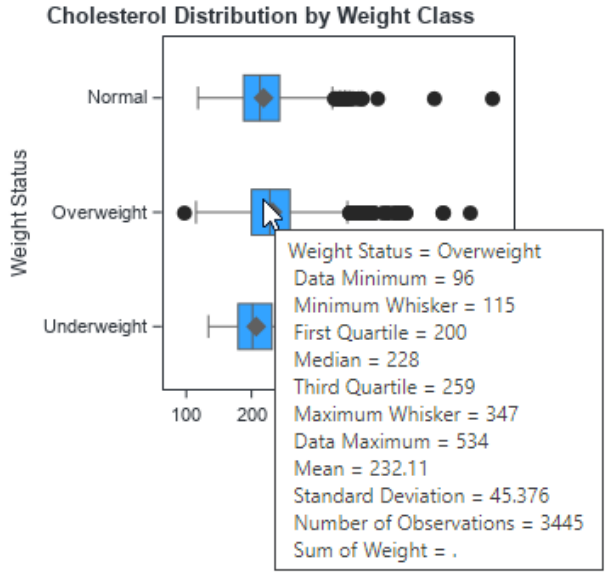
- "Controlling the Size of Your Graph" on page 1888
- "Controlling Image Resolution" on page 1891
- "Adding Data Tips and Other Features" on page 1891
- "Controlling the Image Name and Image Format" on page 1892
- "Resetting ODS GRAPHICS Options" on page 1893

Example Code 16.1 *Controlling Graphics with the ODS GRAPHICS Statement*

```
ods graphics on /
     width=3.25in
     outputfmt=png
     imagemap=on
     border=off;

proc sgplot data=sashelp.heart;
  title "Cholesterol Distribution by Weight Class";
  hbox cholesterol / category=weight_status;
run;

ods graphics on / reset=all;
```

TIP You can use the PATH= option in an ODS destination statement to specify where the output is created.

Figure 16.1 3.25 Inch Graph with Data Tips and No Border



See Also

"ODS GRAPHICS Statement" on page 102

Controlling the Size of Your Graph

Overview of Controlling the Size of Your Graph

The output size of a graph is determined by the following:

- For SGRENDER output, the design size for the graph is specified in the STATGRAPH template that you are rendering.

  The default sizes are:

  - Default design width=640px
  - Default design height=480px
  - Default aspect ratio is 4/3

  These default values are specified in the SAS Registry.
- For SGPLOT, SGPANEL, or SGSCATTER output, the preceding default sizes are used from the SAS Registry.

- DPI settings for the output destinations can be specified by the IMAGE_DPI= or DPI= options. For more information, see Table 16.1 on page 1885.
- Settings for width and height can be specified by the WIDTH= option and the HEIGHT= option for the ODS GRAPHICS statement. For the complete statement syntax, see "ODS GRAPHICS Statement" on page 102.

Specifying Output Size with the ODS GRAPHICS Statement

You can control the output size of a graph by specifying the WIDTH= or HEIGHT= options in the ODS GRAPHICS statement, as shown in the following examples:

```
ods graphics on / width=6in;
ods graphics on / height=4in;
```

The dimensions of the graph can be specified in pixels (for example, 200PX), inches (for example, 3IN), centimeters (for example, 8CM), or other units. The default dimensions of ODS Graphics are 640 pixels wide and 480 pixels high, and these values determine the default aspect ratio. The actual size of the graph in inches depends on your printer or display device. For example, if the resolution of your printer is 100 dots per inch and you want a graph that is 4 inches wide, you should set the width to 4in or 400px.

In the ODS GRAPHICS statement below, taken from Example Code 16.1 on page 1887 , the WIDTH= option changes the width of the graph to 3.25 inches:

```
ods graphics on /
      width=3.25in
      outputfmt=png
      imagemap=on
      border=off;
```

Retaining the Aspect Ratio

It is possible to specify both the width and height in the ODS GRAPHICS statement, as shown in the following example:

```
ods graphics on / width=4.5in height=3.5in;
```

However, it is highly recommended that only one of these options be specified at a time. Using one option at a time ensures that the design aspect ratio of the graph is maintained. For the aspect ratio, height=0.75 x width.

For example, when only width is specified, SAS uses the design aspect ratio of the graph to compute the appropriate height. Retaining the design aspect ratio of the graph is important in many cases as the graph might have a specific layout. For example, a plot that has multiple columns, or that has a statistics table on the side needs a wide aspect ratio. Changing the aspect ratio for this plot by specifying both width and height might produce unpredictable results.

Graph Scaling for User-Specified Image Size

When the size or resolution (DPI) of a graph is changed due to a change in output destination or user specification, the graph is scaled accordingly. For example, when you use the WIDTH= option in the ODS GRAPHICS statement, the output size of the graph is computed based on the new size and the active DPI. All graphical elements such as marker size, line thickness, and font sizes are also scaled accordingly. This is done to prevent rapid shrinking or growth of the fonts and markers.

You can suppress this scaling with the ODS GRAPHICS statement options NOSCALE or SCALE=OFF.

For example:

```
ods graphics on / scale=off;
```

If you want to shrink your graph and keep the default sizes for fonts, use the NOSCALE or SCALE=OFF option in the ODS GRAPHICS statement.

Tip for Embedding Graph Images in Documents

When including graphs in a paper or presentation, it is recommended that you create your graphs by using the exact size that is used to display the graphs in your paper. In other words, avoid generating them at one size and then expanding or shrinking them for inclusion into your document.

It is often useful to produce a graph that fits in one column of a two-column page. In this case, the column size might be 3.25 inches wide. You can place a default output image, which is 640 by 480 pixels, in this space. However, then all graphical elements scale down. This results in small markers and unreadable small fonts.

In this case, it is useful to render the graph to the exact size required, and specify the correct DPI. In the situation above, you would specify a width of 3.25 inches and use the IMAGE_DPI= destination statement option to specify a DPI of 200. Then, the graph produced can be placed in the 3.25″ wide column and the fonts and markers are readable.

Note: To get exactly the specified font sizes, use the NOSCALE option in the ODS GRAPHICS statement.

Controlling Image Resolution

All ODS destinations use a default DPI (dots per inch) setting when creating ODS Graphics output. Graphs that are rendered at higher DPI have greater resolution and larger file size. Although DPI can be set to large values such as 1200, from a practical standpoint, settings larger than 300 dpi are seldom necessary for most applications. Also, setting an unrealistically large DPI like 1200 could cause an out-of-memory condition.

Note: The ODS option for setting DPI is not the same for all destinations. For more information about the DPI for a destination, see the appropriate destination statement in the *SAS Output Delivery System: User's Guide*.

See Also

"ODS Destination Statement Options Affecting ODS Graphics" on page 1885

Adding Data Tips and Other Features

With ODS GRAPHICS statement options, you can control borders and data tips. In the ODS GRAPHICS statement below, taken from Example Code 16.1 on page 1887 , the BORDER=OFF option hides the border around the graph. The IMAGEMAP=ON option enables the generation of data tips and drill downs:

```
ods graphics on /
    width=3.25in
    outputfmt=png
    imagemap=on
    border=off;
```

When viewing HTML output, data tips appear when you position your pointer over certain features of the graph. Data tips and URL drill down functionality are available only for the HTML destination. You can also specify the maximum number of distinct data tip areas allowed before data tips are disabled by using the TOOLTIPMAX= option. There are many other ODS GRAPHICS options that you can use to control other aspects of your graphics. For the complete statement syntax, see "ODS GRAPHICS Statement" on page 102.

TIP Many plot statements include a TIP= option to specify the data tip information to be displayed when you specify IMAGEMAP=ON. In addition, the labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options. You can also specify TIP=NONE to suppress the data tip for a particular plot in a graph overlay.

Controlling the Image Name and Image Format

For ODS Graphics output, by default, the ODS object name is used as the "root" name for the image output file. You can specify names for your graphics image files and the directory in which you want to save them. The IMAGENAME= option specifies the base image file name.

The default image file format is determined by the ODS destination. You can use the OUTPUTFMT= option to specify a different image file format. The formats that are allowed depend on the specified destination.

Note: For a table that shows which formats are allowed for each ODS destination, see "Specifying the Image Format" on page 126.

In the ODS GRAPHICS statement below, taken from Example Code 16.1 on page 1887 , the OUTPUTFMT= option specifies that the file type is a **PNG**:

```
ods graphics on /
          width=3.25in
          outputfmt=png
          imagemap=on
          border=off;
```

You can use the PATH= option in an ODS destination statement to specify where the output is created.

See Also

- "ODS Destination Statement Options Affecting ODS Graphics" on page 1885
- "Supported Image Formats for Output Destinations" on page 127
- "Description of Supported Image Formats" on page 128
- "ODS HTML5 Statement" in *SAS Output Delivery System: User's Guide*
- *SAS Output Delivery System: User's Guide*

Resetting ODS GRAPHICS Options

You can specify the RESET option to change the values for these options back to their defaults. In Example Code 16.1 on page 1887 , the second ODS GRAPHICS statement uses the RESET=ALL option to set options back to their defaults for a new graph:

```
ods graphics on / reset=all;
```

You can also reset some of the ODS GRAPHICS options individually.

For example:

```
ods graphics on / reset=width;
```

For the complete statement syntax, see “ODS GRAPHICS Statement” on page 102.

PART 6

SG Attribute Maps

Chapter 17
Using Attribute Maps to Control Visual Attributes 1897

Chapter 18
Using Discrete Attribute Maps 1901

Chapter 19
Using Range Attribute Maps 1921

1896

17

Using Attribute Maps to Control Visual Attributes

Overview of Attribute Maps 1897
See Also 0
About Discrete Attribute Maps 1898
See Also 1898
About Range Attribute Maps 1898
See Also 1899

Overview of Attribute Maps

The attribute map feature provides a mechanism for controlling the visual attributes that are applied to specific data values in your graphs.

By default, many of the graphical attributes of a plot vary with the plot data. For example, when plots display grouped values, by default, the graphical attributes for each group value are selected from the GraphData1–GraphData*n* style elements in data order. Changes in the data order can significantly change the appearance of the plot.

Attribute maps enable you to assign the same graphical properties to specific values or ranges of values regardless of data order or the data range. Attribute maps enable you to ensure that particular visual attributes are applied based on the value of the data instead of the position of the data in the data set. For more information, see "Making the Appearance of Grouped Data Independent of Data Order" on page 1863.

The ODS Graphics procedures support two types of attribute maps:

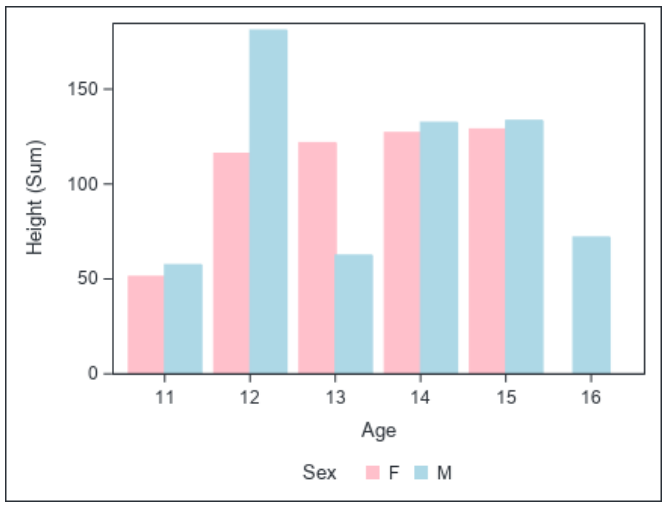
discrete attribute map
: maps discrete data values to graphical properties.

range attribute map
: maps ranges of continuous numeric values to graphical properties.

Note: The procedures do not support attribute maps that are stored in a table.

About Discrete Attribute Maps

A discrete attribute map maps discrete values to graphical properties. For example, consider the following plot of student height grouped by sex. You can use a discrete attribute map to assign pink bars to females and blue bars to males, as shown in the following figure.



Regardless of data order, the same plot colors are applied to the group values.

See Also

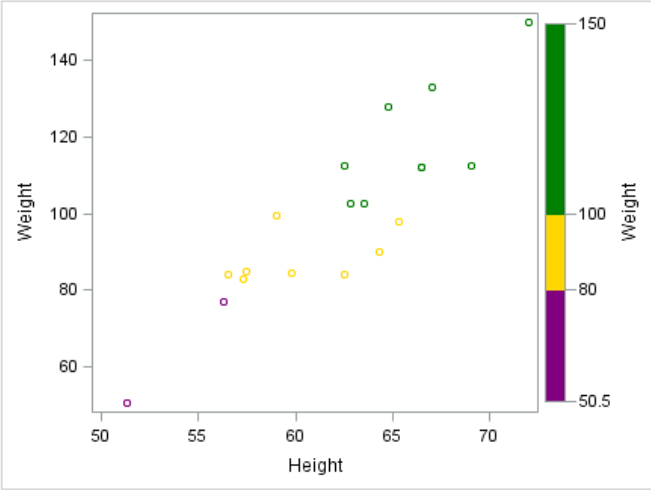
Chapter 18, “Using Discrete Attribute Maps,” on page 1901

About Range Attribute Maps

A range attribute map maps numeric values or ranges of numeric values to graphical properties.

For example, consider the following plot of height and weight distribution. You can use a range attribute map to assign colors to the markers based on specific ranges.

The first range appears as purple. The middle range appears as gold, and the last range appears as green.



See Also

Chapter 19, “Using Range Attribute Maps,” on page 1921

18

Using Discrete Attribute Maps

How Discrete Attribute Maps Work 1901
See Also 1902
Discrete Attribute Map Data Sets 1902
About the Discrete Attribute Map Data Set 1902
Data Sets That Contain Multiple Discrete Attribute Maps 1905
Reserved Discrete Attribute Map Variables 1905
Modify the Procedure to Use the Discrete Attribute Map Data Set 1911
Example: Create a Plot That Uses a Single Discrete Attribute Map 1912
Example: Combine Multiple Discrete Attribute Maps in a Graph 1914
Example: Create a Panel That Uses an Attribute Map 1916
Example: Use a Discrete Attribute Map with an Axis Table 1918
Example: Use a Discrete Attribute Map with the SGPIE Procedure 1919

How Discrete Attribute Maps Work

A discrete attribute map maps discrete values in your data to graphical properties. For example, if you create a graph that plots items sold in different countries, you can specify the display attributes for the sales data of each country. You do this by mapping the attributes to the various country names. The sales results for France might be shown in blue whereas the sales results for Japan might be shown in red.

You can use discrete attribute maps in the SGPLOT, SGPANEL, SGSCATTER, and SGPIE procedures.

IMPORTANT With the SGPLOT, SGPANEL, and SGSCATTER procedures, discrete attribute maps apply to group data. Each observation of the attribute map defines the attributes for a group value. With the SGPIE procedure, discrete attribute maps apply to category data. Each observation of the attribute map defines the attributes for a category value.

Two main tasks are required for discrete attribute mapping:

1 Create a discrete attribute map data set, which associates data values with particular visual attributes. For more information, see "Discrete Attribute Map Data Sets" on page 1902.

2 Modify the procedure and its plot statements to use the data in the discrete attribute map. For more information, see "Modify the Procedure to Use the Discrete Attribute Map Data Set" on page 1911.

See Also

"About Discrete Attribute Maps" on page 1898

Discrete Attribute Map Data Sets

About the Discrete Attribute Map Data Set

Attribute map data sets are used in the procedures to associate data values with visual attributes.

When you have determined which attributes you want to apply to the group or category data in a graph, you can create a discrete attribute map data set. Each observation defines the attributes for a particular group value. With the SGPIE procedure (preproduction), each observation defines the attributes for a particular category value.

Each observation uses reserved variable names for the attribute map identifier (ID), the group or category value (VALUE), and the attributes:

- The ID variable identifies the attribute map that you want to use. A discrete attribute map data set can contain more than one attribute map. This capability enables you to apply different attribute maps to different group variables in a graph. The ID values in the attribute map data set must be continuous (in a sorted order). For more information, see "Data Sets That Contain Multiple Discrete Attribute Maps" on page 1905.

  When used with the SGPIE procedure, the ID variable corresponds to an attribute map that maps graphical properties to category values in a graph.

- The VALUE variable associates a group or category value in your graph data with visual attributes in the attribute map.

  When used with the SGPLOT, SGPANEL, or SGSCATTER procedure, the VALUE variable contains the group values to which you assign graphical properties. For example, suppose that you are graphing fruit sales. The variable in your graph

data set that you want to use for grouping is FRUIT. Your plot statement specifies the GROUP=FRUIT option. Then in your attribute map, the first observation might have VALUE=APPLE, the second observation might have VALUE=BANANA, and so on.

When used with the SGPIE procedure, the VALUE variable contains the category values to which you can assign graphical properties.

- The attribute variables associate visual attributes with an observation in the discrete attribute map data set. Some reserved variables are FILLCOLOR=, LINECOLOR=, LINEPATTERN=, and so on. In the previous example, you might apply attributes to each observation to ensure that the graphics elements are red for apples, yellow for bananas, and so on.

You create the data set using the same methods that you use to create any SAS data set. The main distinctions are that the discrete attribute map data set uses reserved keywords for its variable names, and each observation represents the attributes for a particular data group. The most commonly used method for creating data sets is with a DATA step.

Note: Incorrect data in the attribute map data set can cause the graph to fail. For example, truncated variable values caused by the incorrect variable length being specified results in incorrect data.

Here is an example of a discrete attribute map data set called MYATTRMAP. The observations in this data set contain the attribute map identifier (ID), the group value (VALUE), and the attributes (LINECOLOR, FILLCOLOR).

Figure 18.1 *Listing of the Discrete Attribute Map Data Set MYATTRMAP*

Obs	linecolor	fillcolor	ID	value
1	pink	pink	myid	F
2	lightblue	lightblue	myid	M

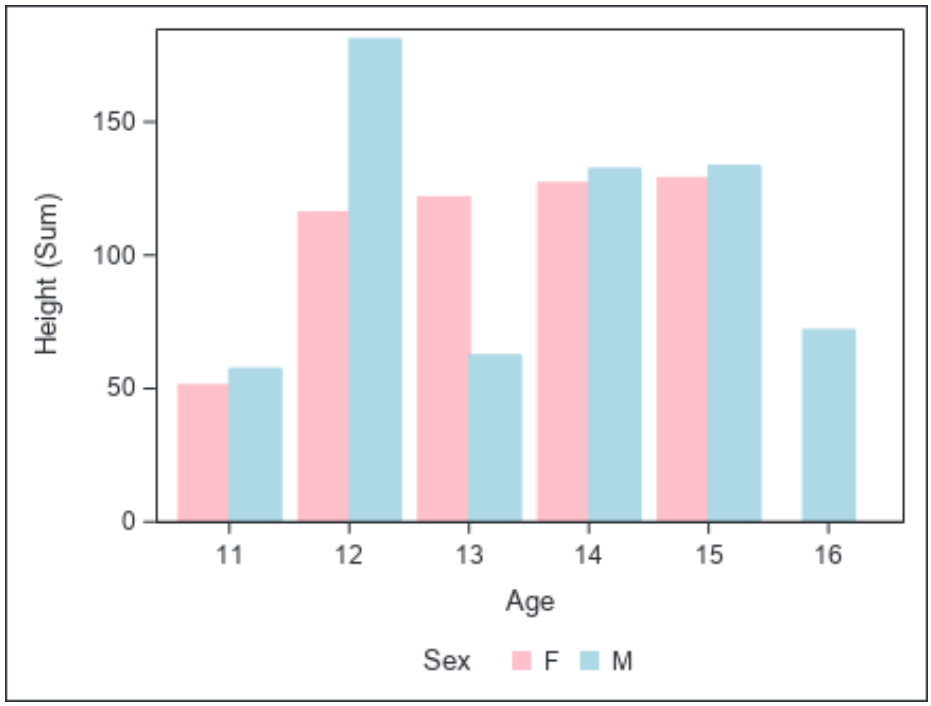
The data set has these characteristics:

- All variables listed are CHAR variables.
- The value of the ID variable, MYID in this case, is referenced in one or more plot statements within the procedure.
- The values of the VALUE variable are valid data group values. These values are case sensitive. The data group is assigned in the plot statement with the GROUP= option.
- The values for LINECOLOR= and FILLCOLOR= are valid SAS colors. You can specify colors using a number of different color-naming schemes. For more information, see “Color-Naming Schemes” on page 1808.

The following output shows a bar chart that uses the MYATTRMAP data set for its pink and blue bar colors. The FILLCOLOR= values determine the color of the bars, and the LINECOLOR= values determine the color of the border edges around the bars.

The chart was produced with the SGPLOT procedure. The name of the data set is referenced in the procedure statement.

Figure 18.2 *Example Output That Uses a Discrete Attribute Map*



This DATA step creates the discrete attribute map data set MYATTRMAP. The ID values for the attribute map are MYID.

```
data myattrmap;
length linecolor $ 9 fillcolor $ 9;
input ID $ value $ linecolor $ fillcolor $;
datalines;
myid  F pink pink
myid  M lightblue lightblue
;
run;
```

This procedure generates the graph. The SGPLOT statement references the SASHELP.CLASS data set and the MYATTRMAP attribute map data set. The plot statement references the attribute map ID (MYID). The options that reference the attribute map (data set and ID) are highlighted.

```
proc sgplot data=sashelp.class dattrmap=myattrmap;
vbar age / response=height group=sex groupdisplay=cluster attrid=myid;
run;
```

For more information about the highlighted options that are used in the procedure, see "Modify the Procedure to Use the Discrete Attribute Map Data Set" on page 1911.

You use a standard syntax to specify colors, line thickness, line patterns, and marker symbols. For more information about this syntax, see the following topics:

- "Color-Naming Schemes" on page 1808

- "Units of Measurement" on page 2079
- List of Line Patterns on page 1829
- List of Marker Symbols on page 1834

Data Sets That Contain Multiple Discrete Attribute Maps

A discrete attribute map data set can contain more than one attribute map. This capability enables you to apply different attribute maps to different group variables in a graph.

When a discrete attribute map data set contains multiple attribute maps:

- The ID variable has more than one value.
- The VALUE variable has different values that correspond to different data groups.
- The ID values in the attribute map data set must be continuous (in a sorted order). If they are not, use the SORT procedure to sort the data set by ID, in ascending or descending order.

In the procedure that generates the graph, the plot statements can specify different attribute map ID values.

Note: Plots that specify different ID values must also specify different groups. A group variable can be associated with only one attribute map ID. If a group is associated with more than one attribute map ID value, the graph produces incorrect attribute mapping and a warning is written to the SAS log.

For an example, see "Example: Combine Multiple Discrete Attribute Maps in a Graph" on page 1914.

Reserved Discrete Attribute Map Variables

About the Reserved Discrete Attribute Map Variables

When a discrete attribute map data set is processed, the procedure looks at the values of specific variables in the discrete attribute map data set. The procedure uses these values to associate visual attributes with group data. Variables in the discrete attribute map data set have predefined names. In each observation, the

procedure looks only for variables with those names. Other variables can be present, but they are ignored.

Variables That Have Style Values

For some variables, you can specify a style element or a style attribute as the value.

Here are examples:

- GraphData2 is an example of a style element.
- GraphData2:Color is an example of a style attribute.

For more information about style elements, see "Style Elements for Use with ODS Graphics" on page 1871.

For a table of the style elements and attributes that you can use with ODS Statistical Graphics, see "Style Elements Affecting Template-Based Graphics" in *SAS Output Delivery System: Procedures Guide*. This table contains each style element, the portion of the graph that it affects, and the default attribute values.

Required Reserved Variables

The ID and VALUE variables are required for every discrete attribute map data set. If they are not found, a warning is written to the SAS log, and the data set is ignored. The remaining attribute variables are used as applicable to the plot type.

Descriptions of the Reserved Variables

The following list describes each reserved variable.

FILLCOLOR="*color*"
specifies the fill color. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

FILLSTYLEELEMENT="*style-element*"
specifies the style element for fill attributes. If you specify the style element, you can also specify the FILLCOLOR variable to override the color.

Alias FILLSTYLE=

Restriction This variable applies to only the SGPLOT, SGPANEL, and SGRENDER procedures.

FILLTRANSPARENCY=*value*
specifies the degree of transparency for the fill.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

ID="*text-string*"
(required) specifies the ID of the attribute map. This value is referenced from the ATTRID option on one or more plot statements.

Though every observation must have an ID value, different observations can have different values. You might use different values in the data set if you want plots to have different attributes.

LINECOLOR="*color*"
specifies the color of the line. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

The color that you specify here also applies to the line labels, if displayed.

LINEPATTERN="*line-pattern*"
specifies a line pattern for lines and outlines. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Note This variable has no effect on bar chart outlines.

LINESTYLEELEMENT="*style-element*"
specifies the style element for line attributes. If you specify the style element, you can also specify the LINEPATTERN and LINECOLOR variables to override specific appearance attributes.

Alias LINESTYLE=

Restriction This variable applies to only the SGPLOT, SGPANEL, and SGRENDER procedures.

LINETHICKNESS=*numeric-value*
specifies the line thickness in pixels.

MARKERCOLOR="*color*"
specifies the color for the markers. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

The color that you specify here also applies to the marker labels, if displayed.

MARKERSIZE=*numeric-value*
specifies the size of the marker in pixels.

MARKERSTYLEELEMENT="*style-element*"
specifies the style element for marker attributes. If you specify the style element, you can also specify the MARKERSYMBOL, MARKERCOLOR, and MARKERSIZE variables to override specific appearance attributes.

Alias MARKERSTYLE=

Restriction This variable applies to only the SGPLOT, SGPANEL, and SGRENDER procedures.

MARKERSYMBOL="*symbol-name*"
specifies the symbol for the markers. See marker symbols on page 1834.

MARKERTRANSPARENCY=*value*
specifies the degree of transparency for the markers.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

NOCASE="TRUE" | "FALSE"
specifies whether value comparisons in the attribute map are case sensitive.

"TRUE"
The value comparisons are not case sensitive.

"FALSE"
The value comparisons are case sensitive. Each instance of a value must have the exact same capitalization.

Default "FALSE"

SHOW="ATTRMAP" | "DATA"
specifies whether values in the attribute map are displayed in the legend.

"ATTRMAP"
displays all attribute map values in the legend.

"DATA"
displays only the map values for which there is data.

Default "DATA"

TEXTCOLOR="*color*"
specifies the color of the text in an axis table. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Restriction This attribute does not apply in all cases. See "Text Attribute Variable Applicability" on page 1909.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

TEXTFAMILY="*font-family*"
specifies the font family for the text in an axis table.

Restriction This attribute does not apply in all cases. See "Text Attribute Variable Applicability" on page 1909.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

TEXTSTYLE="NORMAL" | "ITALIC"
specifies the style of the text in an axis table.

Restriction This attribute does not apply in all cases. See "Text Attribute Variable Applicability" on page 1909.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

TEXTSTYLEELEMENT="*style-element*"
specifies the style element for the text in an axis table. Attributes FontColor, FontFamily, FontStyle, and FontWeight are honored. You can also specify the TEXTCOLOR, TEXTFAMILY, TEXTSTYLE,and TEXTWEIGHT variables to override specific attributes in the specified style.

Restrictions This attribute does not apply in all cases. See "Text Attribute Variable Applicability" on page 1909.

The FontSize attribute in the specified style is ignored.

TEXTWEIGHT="NORMAL" | "BOLD"
specifies the weight of the text in an axis table.

Restriction This attribute does not apply in all cases. See "Text Attribute Variable Applicability" on page 1909.

See "Example: Use a Discrete Attribute Map with an Axis Table" on page 1918

VALUE="*text-string*"
(required) specifies the group data value that is assigned to the attributes. The value must be valid for the data group that is assigned in the plot statement with the GROUP= option. You can provide different values in the data set if your plots specify different group variables.

Notes By default, the group values are case sensitive. Each instance of a value must have the exact same capitalization. To make the values case insensitive, specify NOCASE="TRUE".

If the group value is formatted, then the *text-string* in the attribute map data set must contain the formatted value.

VALUE= also accepts the _MISSING_ and _OTHER_ keywords. The keyword _OTHER_ in the column can be used to define the attributes of any values that are not explicitly defined in the map. The _MISSING_ keyword is used to define attributes when the data contains missing values. These keywords are not case sensitive.

Text Attribute Variable Applicability

The text attribute variables cannot be applied to text for all plots. The following table lists the cases in which the text attribute variables apply.

Table 18.1 *The Text Attribute Variables and the Procedure Statements to Which They Apply*

Text Attribute Variable	Procedure	Applicability
TEXTCOLOR=	SGPANEL	applies to only COLAXISTABLE, ROWAXISTABLE, and TEXT
	SGPIE	not applicable
	SGPLOT	applies to only TEXT, XAXISTABLE, and YAXISTABLE
	SGRENDER	applies to only AXISTABLE and TEXTPLOT
	SGSCATTER	applies to COMPARE, MATRIX, and PLOT
TEXTFAMILY=, TEXTSTYLE=, and TEXTWEIGHT=	SGPANEL	applies to only COLAXISTABLE and ROWAXISTABLE
	SGPIE	not applicable
	SGPLOT	applies to only XAXISTABLE and YAXISTABLE
	SGRENDER	attributes TEXTSTYLE= and TEXTWEIGHT= apply to only AXISTABLE attribute TEXTFAMILY= is ignored
	SGSCATTER	not applicable
TEXTSTYLEELEMENT=[1]	SGPANEL	attribute Color applies to only COLAXISTABLE, ROWAXISTABLE, and TEXT attributes FontFamily=, FontStyle=, and FontWeight= apply to only COLAXISTABLE and ROWAXISTABLE
	SGPIE	not applicable
	SGPLOT	attribute Color applies to only TEXT, XAXISTABLE, and YAXISTABLE attributes FontFamily=, FontStyle=, and FontWeight= apply to only XAXISTABLE and YAXISTABLE
	SGRENDER	attribute Color applies to only AXISTABLE and TEXTPLOT

Text Attribute Variable	Procedure	Applicability
		attributes FontFamily=, FontStyle= and FontWeight= apply to only AXISTABLE
	SGSCATTER	attribute Color applies to COMPARE, MATRIX, and PLOT attributes FontFamily=, FontStyle=, and FontWeight= are ignored.

1 Applicable attributes are Color and either Font or font attributes FontFamily=, FontStyle=, and FontWeight=. The font size specification in attribute Font and the font attribute FontSize= are ignored.

Modify the Procedure to Use the Discrete Attribute Map Data Set

After you have created a discrete attribute map data set, you can modify the ODS Graphics procedure and its plot statement or statements to reference the discrete attribute map data.

To reference attribute map data in a procedure:

1. Add the DATTRMAP= option to the procedure statement and specify the name of the discrete attribute map data set.
2. If you are using the SGPLOT, SGPANEL, or SGSCATTER procedure, then perform these steps for each plot statement that maps attributes:
   - Specify the GROUP= option. The group's values must correspond to the values in the VALUE variable in the attribute map data set.
   - Specify the ATTRID= option. This option specifies the value of the ID variable that is in the discrete attribute map data set.

     If the discrete attribute map data set contains more than one ID value, then the plot statements can specify different ATTRID values. However, plots that specify different ATTRID values must also specify different groups. A group variable can be associated with only one attribute map ID. For more information, see "Data Sets That Contain Multiple Discrete Attribute Maps" on page 1905.

     If the ATTRID value that you specify does not match a value for the ID variable in the attribute map data set, a warning is written to the SAS log and the ATTRID is ignored.

3 If you are using the SGPIE procedure (preproduction), then specify the ATTRID= option in the chart statement. The ATTRID= option specifies the value of the ID variable that is in the discrete attribute map data set.

Attribute mapping is supported in the PLOT statement of the SGSCATTER procedure.

Note: WHERE expressions cannot be used to subset attribute map data in the ODS Graphics procedures. The reason is that the WHERE expression might unintentionally affect both the data and the attribute map. If you want to subset your attribute map data, set up that data prior to using the procedure.

In the SGPLOT and SGPANEL procedures, any plot statement that supports GROUP variables supports discrete attribute mapping. The following plot statements support discrete attribute mapping:

BAND	HISTOGRAM	TEXT
BLOCK	HLINE	VBAR
BUBBLE	LINEPARM	VBARBASIC
DENSITY	LOESS	VBARPARM
DOT	NEEDLE	VBOX
FRINGE	PBSPLINE	VECTOR
HBAR	POLYGON	VLINE
HBARBASIC	REG	WATERFALL *
HBARPARM	SCATTER	XAXISTABLE *
HBOX	SERIES	YAXISTABLE *
HEATMAPPARM	SPLINE	COLAXISTABLE **
HIGHLOW	STEP	ROWAXISTABLE **

*applies to the SGPLOT procedure only

**applies to the SGPANEL procedure only

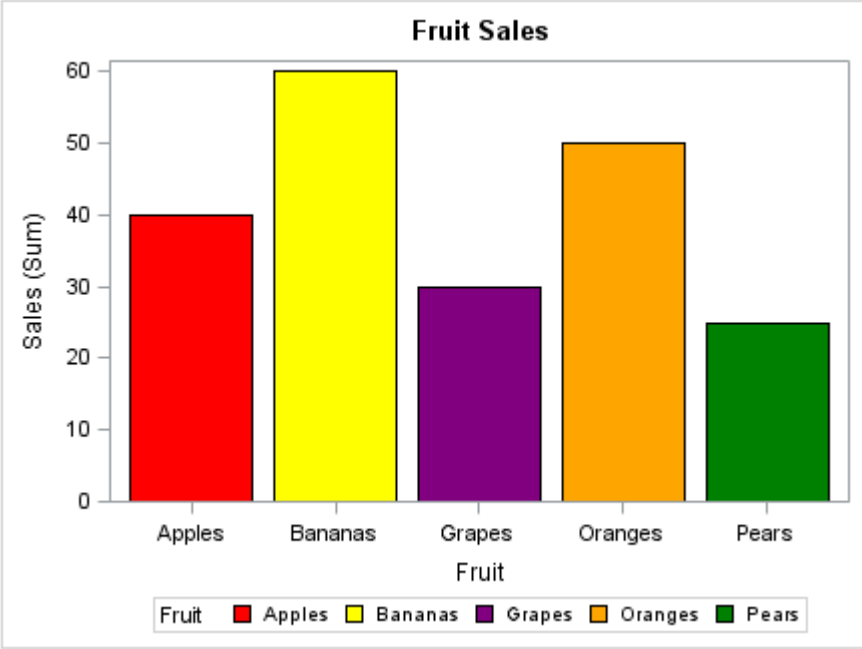
Here is an example of an SGPLOT procedure that references the discrete attribute map data set shown in the example Figure 18.1 on page 1903. The options that reference the attribute map (data set and ID) are highlighted.

```
proc sgplot data=sashelp.class dattrmap=myattrmap;
vbar age / response=height group=sex groupdisplay=cluster attrid=myid;
run;
```

Example: Create a Plot That Uses a Single Discrete Attribute Map

This example creates a simple bar chart that uses a discrete attribute map.

Figure 18.3 Bar Chart That Uses a Discrete Attribute Map



Here is the SAS code that creates the input data set, the attribute map data set, and the SGPLOT procedure output.

This DATA step creates the data that is used for the graph. The FRUIT variable is used in the procedure to group the data.

```
data fruit_sales;
input Fruit $ Sales;
datalines;
Apples   40
Pears    25
Oranges  50
Grapes   30
Bananas  60
run;
```

This DATA step creates the discrete attribute map data set. The data set contains one attribute map, which is specified by the ID variable. All ID values for the attribute map are FRUIT.

```
data attrmapfruit;
retain linecolor "black";
input id $ value $ fillcolor $;
datalines;
fruit Apples  red
fruit Pears   green
fruit Oranges orange
fruit Grapes  purple
fruit Bananas yellow
;
run;
```

This PROC step generates the graph. The SGPLOT statement references the FRUIT_SALES data set and the ATTRMAPFRUIT attribute map data set. The plot statement references the attribute map ID (FRUIT).

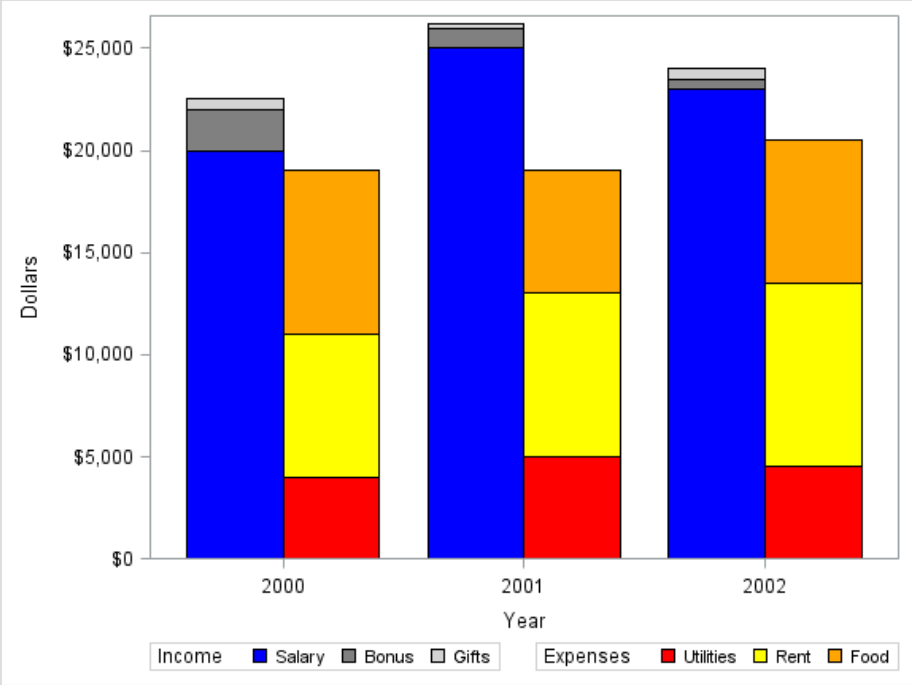
```
title  "Fruit Sales";
proc sgplot data=fruit_sales dattrmap=attrmapfruit;
vbar Fruit / response=Sales group=Fruit attrid=fruit;
run;
title;
```

Example: Combine Multiple Discrete Attribute Maps in a Graph

This example uses a discrete attribute map data set that contains two attribute maps. Here are the noteworthy characteristics of the attribute map data set and the procedure:

- The attribute map data set has these features:
  - The ID variable has two values.
  - The VALUE variable has different values that correspond to two different data groups.
  - The ID values in the attribute map data set are continuous (in a sorted order). If they were not, the example would use the SORT procedure to sort the data set by ID (in ascending or descending order).
- In the procedure, the plot statements specify different ATTRID= values in order to use the different attribute maps. The plot statements also specify different GROUP= values. (A group variable can be associated with only one attribute map ID. For more information, see “Data Sets That Contain Multiple Discrete Attribute Maps” on page 1905.)

Figure 18.4 *Output That Uses Two Discrete Attribute Maps*



Here is the SAS code that creates the attribute map data set and uses the SGPLOT procedure to generate the graph.

This DATA step creates the data that is used for the graph. The STATE and TYPE variables are used in the procedure to group the data.

```
data finances;
format income dollar8. expense dollar8.;
length expensetype $ 9;
input Year incometype $ income expensetype $ expense;
datalines;
2000 Salary 20000 Utilities 4000
2000 Bonus   2000 Rent      7000
2000 Gifts    500 Food      8000
2001 Salary 25000 Utilities 5000
2001 Bonus   1000 Rent      8000
2001 Gifts    200 Food      6000
2002 Salary 23000 Utilities 4500
2002 Bonus    500 Rent      9000
2002 Gifts    500 Food      7000
;
run;
```

This DATA step creates the discrete attribute map data set. The data set contains two attribute maps. Each attribute map is specified by an ID value: INCOME or EXPENSE.

```
data attrmap;
length value $ 9 fillcolor $ 9;
retain linecolor "black";
input id $ value $ fillcolor $;
datalines;
income  Salary    blue
```

```
income  Bonus     gray
income  Gifts     lightgray
expense Utilities red
expense Rent      yellow
expense Food      orange
;
run;
```

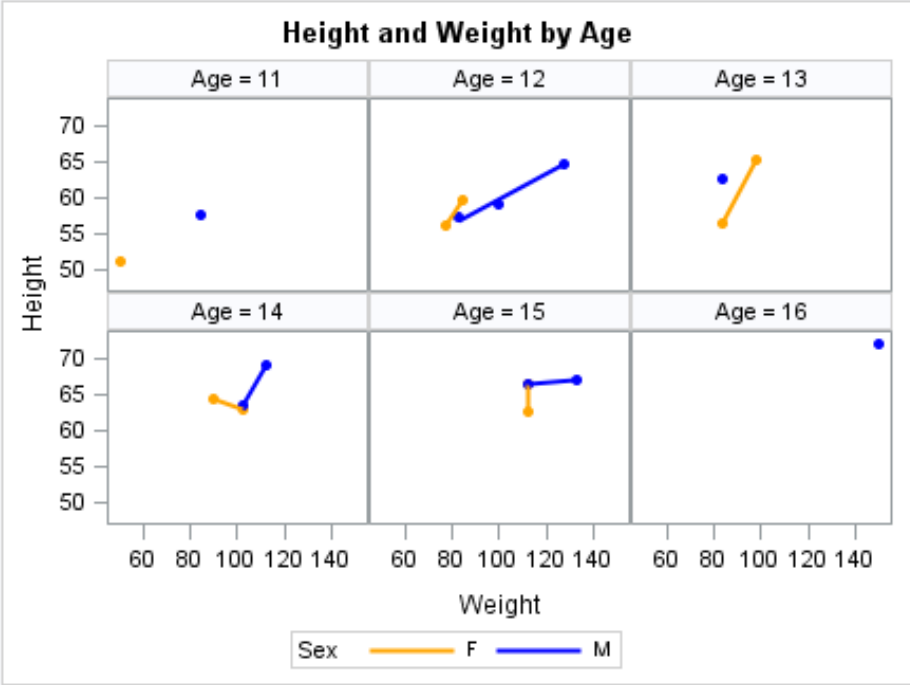
This PROC step generates the graph. The SGPLOT statement references the FINANCES data set and the ATTRMAP attribute map data set. Each plot statement references an attribute map ID. The plot statements also specify different GROUP= values.

```
proc sgplot data=finances dattrmap=attrmap;
yaxis label="Dollars";
vbarparm category=year response=income / group=incometype attrid=income
         barwidth=0.4 groupdisplay=stack
         discreteoffset=-0.2 name="income";
vbarparm category=year response=expense / group=expensetype
attrid=expense
         barwidth=0.4 groupdisplay=stack
         discreteoffset=0.2 name="expense";
keylegend "income" / position=bottomleft title="Income";
keylegend "expense" / position=bottomright title="Expenses";
run;
```

Example: Create a Panel That Uses an Attribute Map

This example uses the SGPANEL procedure to generate discrete attribute map output.

Figure 18.5 *Panel That Uses a Discrete Attribute Map*



Here is the SAS code that creates the attribute map data set and runs the SGPANEL procedure.

This DATA step creates the discrete attribute map data set. The data set contains one attribute map, which is specified by the ID variable. All ID values for the attribute map are MYID.

```
data attrmap;
retain ID "myid";
retain markersymbol "circlefilled";
input value $ markercolor $ linecolor $ linepattern $;
datalines;
F  orange  orange  solid
M  blue    blue    solid
;
run;
```

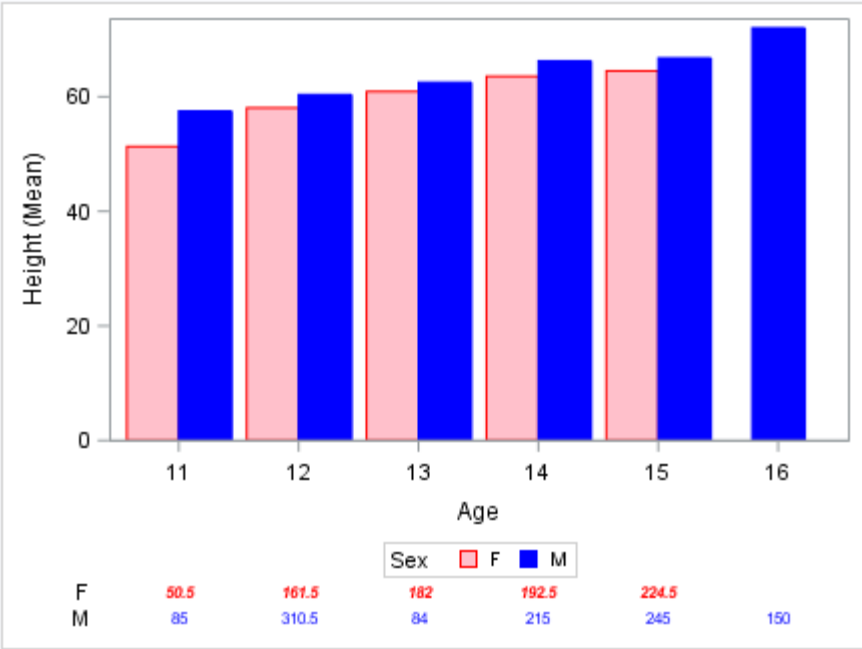
This PROC step generates the graph. The SGPANEL statement references the SASHELP.CLASS data set and the ATTRMAP attribute map data set. The plot statement references the attribute map ID (MYID).

```
title "Height and Weight by Age";
proc sgpanel data=sashelp.class dattrmap=attrmap;
panelby age;
reg x=weight y=height / group=sex attrid=myid;
run;
title;
```

Example: Use a Discrete Attribute Map with an Axis Table

This example creates a simple bar chart along with an axis table. Both the chart and the axis table use a discrete attribute map.

Figure 18.6 *Bar Chart and Axis Table That Use a Discrete Attribute Map*



Here is the SAS code that creates the attribute map data set and the SGPLOT procedure output.

This DATA step creates the discrete attribute map data set. The data set contains one attribute map, which is specified by the ID variable.

```
data myattrmap;
  id="myid"; value="M";
  linecolor="blue"; fillcolor="blue";
  textcolor="blue"; textstyle="normal";
  textweight="normal";
output;

id="myid"; value="F";
  linecolor="red"; fillcolor="pink";
  textcolor="red"; textstyle="italic";
  textweight="bold";
output;
run;
```

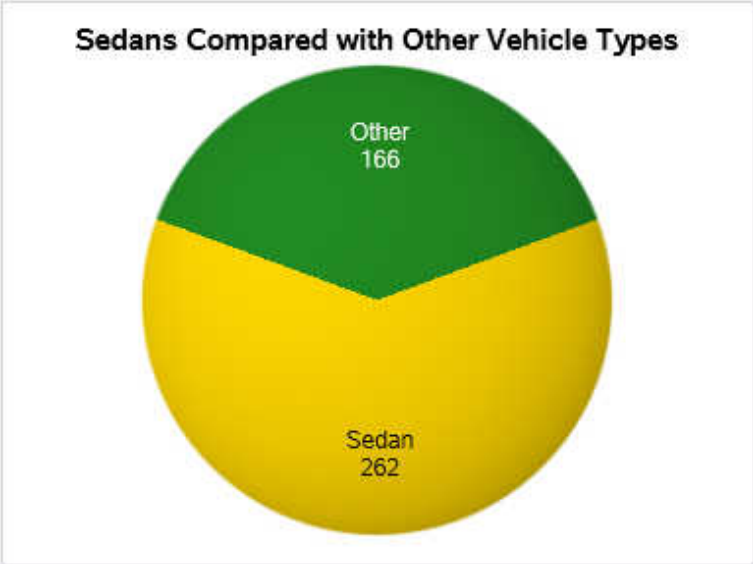
This PROC step generates the graph. The plot statement references the attribute map ID (MYID).

```
proc sgplot data=sashelp.class dattrmap=myattrmap;
  vbar age / response=height group=sex stat=mean
     groupdisplay=cluster attrid=myid;
  xaxistable weight / textgroup=sex
     textgroupid=myid;
run;
```

Example: Use a Discrete Attribute Map with the SGPIE Procedure

This example creates a simple pie chart that uses a discrete attribute map. The example first defines a format that uses the Other keyword to group all vehicle types that are not sedans.

Figure 18.7 *Pie Chart That Uses a Discrete Attribute Map*



Here is the SAS code that defines the format, creates the attribute map data set, and generates the SGPIE procedure output.

```
proc format;
  value $type
    'Sedan'='Sedan'
      other='Other';
run;

data attrmap;
length ID $ 9 FillColor $ 20;
```

```
input id $ Value $  fillcolor $;
datalines;
mytype   Sedan  gold
mytype   Other  forestgreen
;
run;

title "Sedans Compared with Other Vehicle Types";
proc sgpie data=sashelp.cars dattrmap=attrmap;
  format type $type.;
  pie type / startangle=90 attrid=mytype;
run;
title;
```

19

Using Range Attribute Maps

Main Tasks for Range Attribute Mapping **1921**
See Also 1921
Range Attribute Map Data Sets **1922**
About the Range Attribute Map Data Set 1922
Data Sets That Contain Multiple Range Attribute Maps 1924
Reserved Range Attribute Map Variables 1924
Tips for Defining Ranges 1928
Modify the Procedure to Use the Range Attribute Map Data Set **1928**
Example: Create a Panel That Uses COLORMODELn Variables **1930**

Main Tasks for Range Attribute Mapping

A range attribute map maps numeric values or ranges of numeric values to graphical properties.

Two main tasks are required for range attribute mapping:

1 Create a range attribute map data set, which associates data values with particular visual attributes. For more information, see "Range Attribute Map Data Sets" on page 1922.

2 Modify the procedure and its plot statements to use the data in the range attribute map. You can use attribute maps in the SGPLOT, SGPANEL, and SGSCATTER procedures (not all plot statements support attribute maps). For more information, see "Modify the Procedure to Use the Range Attribute Map Data Set" on page 1928.

See Also

"About Range Attribute Maps" on page 1898

Range Attribute Map Data Sets

About the Range Attribute Map Data Set

When you have determined which attributes you want to apply to the data in a graph, you can create a range attribute map data set. Each observation defines the color attributes for a particular range.

Each observation uses reserved variable names for the attribute map identifier (ID), one or two variables that define the range (MIN and MAX), and the color attributes:

- The ID variable identifies the attribute map that you want to use. A range attribute map data set can contain more than one attribute map. This capability enables you to apply different attribute maps to different plots in a graph. The ID values in the attribute map data set must be continuous (in a sorted order). For more information, see "Data Sets That Contain Multiple Range Attribute Maps" on page 1924.
- The MIN and MAX variables define the ranges in the attribute map. The MIN variable is always required. The MAX variable is sometimes required depending on the value that you provide for the MIN variable.

  In addition, two variables (EXCLUDEMIN, EXCLUDEMAX) enable you to exclude the first or last end points in a range, respectively. For example, you might define the first range as 0–100, and use EXCLUDEMIN to exclude the 0 values.
- The attribute variables associate color attributes with an observation in the range attribute map data set.

  There are three ways to assign colors to a range:

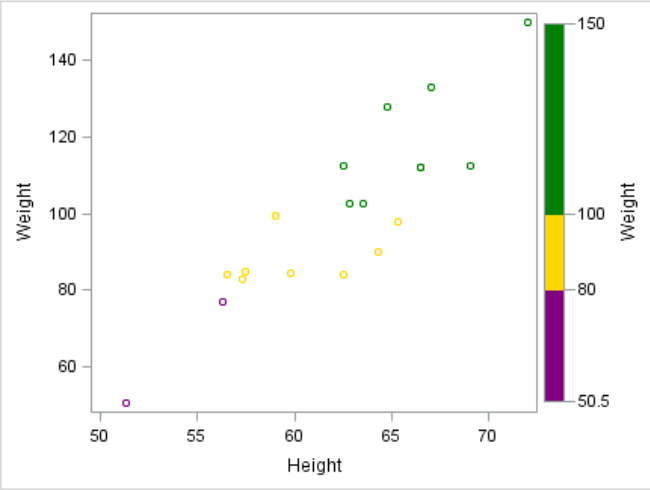
  - assign a single color to a range.
  - specify a reference to an ODS style element that contains color ramp start and end colors (with an optional neutral color).
  - use variable names COLORMODEL1 – COLORMODEL*n* to specify several consecutive colors for the range. This list of colors creates a gradient across the range. See "Example: Create a Panel That Uses COLORMODELn Variables" on page 1930.

You create the data set using the same methods that you use to create any SAS data set. The main distinctions are that a range attribute map data set uses reserved keywords for its variable names, and each observation represents the attributes for a particular data range. The most commonly used method for creating data sets is with a DATA step..

Note: Incorrect data in the attribute map data set can cause the graph to fail. For example, truncated variable values caused by the incorrect variable length being specified results in incorrect data.

The following output shows a scatter plot that uses a range attribute map to assign marker colors based on specific ranges. The first range appears as purple. The middle range appears as gold, and the last range appears as green. The chart was produced with the SGPLOT procedure.

Figure 19.1 *Example Range Attribute Map Output*



Here is the code for the example.

The DATA step creates the range attribute map data set MYRATTRMAP. The ID values for the attribute map are MYID. Three ranges are defined.

```
data myrattrmap;
retain id "myID";
length min $ 5 max $ 5;
input min $ max $ color $ altcolor $;
datalines;
_min_ 80    purple purple
80    100   gold   gold
100   _max_ green  green
;
run;
```

The following procedure generates the graph. The SGPLOT statement references the SASHELP.CLASS data set and the MYRATTRMAP attribute map data set. The plot statement references the attribute map ID (MYID). The options that reference the attribute map (data set and ID) are highlighted.

```
proc sgplot data=sashelp.class rattrmap=myrattrmap;
  scatter x=height y=weight /
    colorresponse=weight rattrid=myID;
run;
```

Note: The COLORRESPONSE= option must be specified in order for the attribute map to be applied to the plot.

Data Sets That Contain Multiple Range Attribute Maps

A range attribute map data set can contain more than one attribute map. This capability enables you to apply different attribute maps to different group variables in a graph.

When a range attribute map data set contains multiple attribute maps:

- The ID variable has more than one value.
- The ID values in the attribute map data set must be continuous (in a sorted order). If they are not, use the SORT procedure to sort the data set by ID, in ascending or descending order.

In the procedure that generates the graph, the plot statements can specify different attribute map ID values.

Reserved Range Attribute Map Variables

About the Reserved Range Attribute Map Variables

When a range attribute map data set is processed, the procedure looks at the values of specific variables in the attribute map data set. The procedure uses these values to associate visual attributes with the range data. Variables in the attribute map data set have predefined names. In each observation, the procedure looks only for variables with those names. Other variables can be present, but they are ignored.

Required Reserved Variables

Each observation in the data set requires the ID variable and one or two variables that define the range (MIN and MAX). If the required variables are not found, a warning is written to the SAS log, and the data set is ignored.

Note: For helpful information about defining ranges, see "Tips for Defining Ranges" on page 1928.

The following list describes the required variables.

ID= *"text-string"*
specifies the ID of the attribute map. This value is referenced from the RATTRID option in one or more plot statements.

Even though every observation must have an ID value, different observations can have different values. You might use different values in the data set if you want plots to have different attributes. See "Data Sets That Contain Multiple Range Attribute Maps" on page 1924.

MIN=*numeric-value* | *"keyword"*
defines the minimum value of the range specification. You can specify a numeric value, or you can specify one of several accepted keywords. The keywords are character values.

You can specify one of the following keywords as the minimum value:

MIN	indicates the minimum data value for the response variable.
NEGMAX	indicates –MAX
NEGMAXABS	indicates –max(abs(MIN) , abs(MAX))

Rather than specifying a low value, you can use one of the following keywords for the range specification.

Note: When you specify one of the following four keywords, the value in the MAX column is not required.

MISSING	indicates a mapping for missing values. The visual attributes for this setting are obtained from the GraphMissing style element. If one observation in the data set specifies this value and another observation specifies the keyword _OTHER_, then the _OTHER_ range does not include missing values.
OTHER	creates a category for all other column values not explicitly assigned to a range. The _OTHER_ values can consist of several non-contiguous ranges. The visual attributes for this setting are obtained from the GraphOther style element.
OVER	creates a range for all data between the highest mapped value and the highest actual data value. The visual attributes for this setting are obtained from the GraphOverflow style element. If one observation in the data set specifies this value and another observation specifies the keyword _OTHER_, then the _OTHER_ range does not include overflow values.
UNDER	creates a range for all data values between the lowest mapped value and the lowest actual data value. The visual attributes for this setting are obtained from the GraphUnderflow style element. If one observation in the data set specifies this value and another observation specifies the keyword _OTHER_, then the _OTHER_ range does not include underflow values.

Tip Use the EXCLUDEMIN optional variable to exclude the lowest end point in the range. For example, you might define the first range as 0–100, and use EXCLUDEMIN to exclude the 0 values.

MAX=*numeric-value* | *"keyword"*
defines the maximum value of the range specification. You can specify a numeric value, or you can specify one of two accepted keywords. The keywords are character values.

This variable is required unless one of the special four keywords is specified for the MIN variable (_MISSING_, _OTHER_, _OVER_, _UNDER_).

You can specify one of the following keywords as the maximum value:

MAX	indicates the maximum data value for the response variable.
MAXABS	indicates max(abs(MIN) , abs(MAX))

Tip Use the EXCLUDEMAX optional variable to exclude the highest end point in the range.

Optional Reserved Variables

The following list describes each optional reserved variable.

ALTCOLOR= *"color"*
assigns a single color to lines, markers, and text in a range. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

ALTCOLORMODEL1–*N*
specifies several consecutive line, marker, and text colors for the range. This list of colors creates a gradient across the range.

Requirement The columns must be contiguous (ALTCOLORMODEL1, ALTCOLORMODEL2, and so on). Any gaps prevent the remaining colors from being used.

ALTCOLORMODELSTYLE=*"style-element"*
specifies a style reference to a gradient style element.

style-element
specifies the name of a gradient style element. The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value.
NEUTRALCOLOR	specifies the color for the midpoint of the data range. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value.

Interaction This option is ignored if the ALTCOLOR= option is specified.

Tip To display the range as a gradient ramp, choose a style element such as TwoColorRamp, TwoColorAltRamp, ThreeColorRamp, or ThreeColorAltRamp.

See "Style Elements for Use with ODS Graphics" on page 1871

Example `altcolormodelstyle=TwoColorRamp`

Default The ThreeColorAltRamp style element

COLOR= "*color*"
specifies the fill color for the range. You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

COLORMODEL1–*N*
specifies several consecutive fill colors for the range. This list of colors creates a gradient across the range.

Requirement The columns must be contiguous (COLORMODEL1, COLORMODEL2, and so on). Any gaps prevent the remaining colors from being used.

See "Example: Create a Panel That Uses COLORMODELn Variables" on page 1930

COLORMODELSTYLE="*style-element*"
specifies a style reference to a gradient style element.

style-element
specifies the name of a gradient style element. The style element should contain these style attributes:

STARTCOLOR	specifies the color for the smallest data value.
NEUTRALCOLOR	specifies the color for the midpoint of the data range. This attribute is not required when you specify a two-color ramp model.
ENDCOLOR	specifies the color for the highest data value.

Interaction This option is ignored if the COLOR= option is specified.

Tip To display the range as a gradient ramp, choose a style element such as TwoColorRamp, TwoColorAltRamp, ThreeColorRamp, or ThreeColorAltRamp.

See "Style Elements for Use with ODS Graphics" on page 1871

Example `colormodelstyle=TwoColorRamp`

Default The ThreeColorAltRamp style element

EXCLUDEMAX="TRUE" | "FALSE" | 1 | 0
Specifies whether to exclude the maximum value from the range specification. This column can be character or numeric. If the column is character, you must use the values "TRUE" or "FALSE". If the column is numeric, the values must be 1 or 0.

EXCLUDEMIN="TRUE" | "FALSE" | 1 | 0
Specifies whether to exclude the minimum value from the range specification. This column can be character or numeric. If the column is character, you must use the values "TRUE" or "FALSE". If the column is numeric, the values must be 1 or 0.

Tips for Defining Ranges

Here are some helpful notes and tips for specifying ranges in the attribute map data set:

- If two ranges share a common endpoint (for example, 10–20 and 20–30) and you do not use EXCLUDEMIN or EXCLUDEMAX, then the common endpoint belongs to the lower encountered range (10–20 in this case). The order of the specification does not matter.
- If two or more ranges define colors to associate with the same numeric values or ranges, then the first range's settings are used.
- If any range overlaps another range (for example, 10–20 and 15–25), then the entire attribute map is ignored and default coloring is used.
- To set a single numeric value, specify the same value for both the low value and the high value.
- If the low value is not less than or equal to the high value, then the range specification is invalid, and the range is ignored in the attribute map.
- If a range is not defined for keyword MIN="_OTHER_", then gaps within the attribute map ranges are assigned the default color that is defined by the GraphOther:ContrastColor style reference.

Modify the Procedure to Use the Range Attribute Map Data Set

After you have created a range attribute map data set, you can modify the procedure and its plot statements to reference the attribute map data. You can use range attribute maps in the SGPLOT, SGPANEL, and SGSCATTER procedures.

To reference attribute map data in a procedure:

1 Add the RATTRMAP= option to the procedure statement and specify the name of the range attribute map data set.

2 For each plot statement that maps attributes, perform these steps:

- Specify the RATTRID= option. This option specifies the value of the ID variable in the range attribute map data set.

  If the range attribute map data set contains more than one ID value, then the plot statements can specify different RATTRID values. For more information, see "Data Sets That Contain Multiple Range Attribute Maps" on page 1924.

  If the RATTRID value that you specify does not match a value for the ID variable in the attribute map data set, a warning is written to the SAS log and the RATTRID option is ignored.

- Specify the COLORRESPONSE= option.

Attribute mapping is supported in the PLOT statement of the SGSCATTER procedure.

Note: WHERE expressions cannot be used to subset attribute map data in the ODS Graphics procedures. The reason is that the WHERE expression might unintentionally affect both the data and the attribute map. If you want to subset your attribute map data, set up that data prior to using the procedure.

In the SGPLOT and SGPANEL procedures, the following plot statements support range attribute mapping:

BUBBLE	HEATMAPPARM	TEXT
DOT	HIGHLOW	VBAR
HBAR	POLYGON	VBARBASIC
HBARBASIC	SCATTER	VBARPARM
HBARPARM	SERIES	VECTOR
HEATMAP	SPLINE	WATERFALL*

*applies to the SGPLOT procedure only

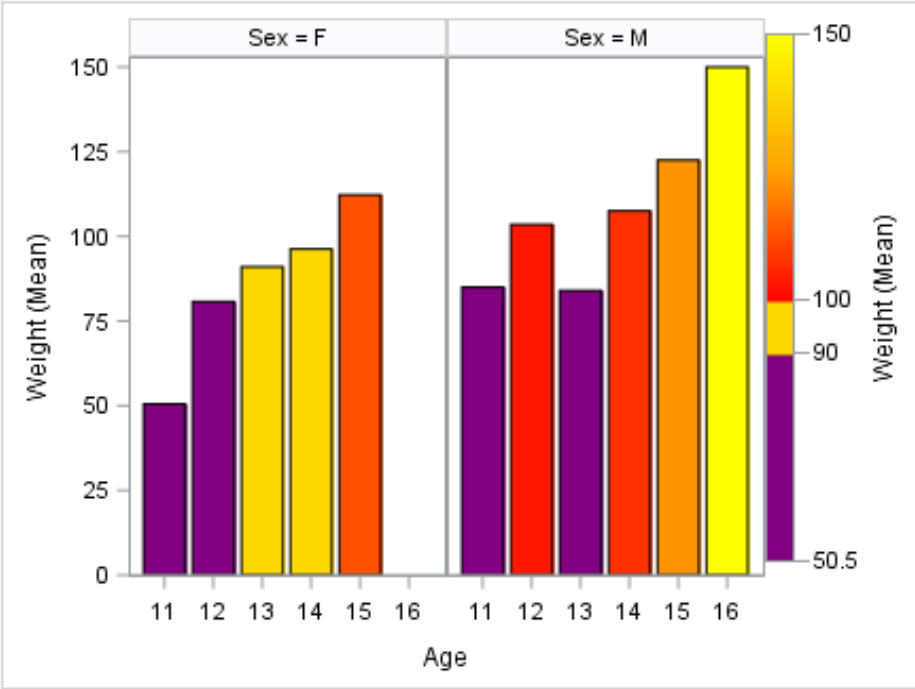
Here is an example of an SGPLOT procedure that references the range attribute map data set shown in Figure 19.1 on page 1923. The options that reference the attribute map (data set and ID) are highlighted.

```
proc sgplot data=sashelp.class rattrmap=myrattrmap;
  scatter x=height y=weight /
    colorresponse=weight rattrid=myID;
run;
```

Example: Create a Panel That Uses COLORMODEL*n* Variables

This example uses the SGPANEL procedure to generate range attribute map output. The graph uses a range attribute map to assign fill colors based on specific ranges. The first range appears as purple. The middle range appears as gold, and the last range appears as a gradient of red, orange, and yellow.

Figure 19.2 *Panel That Uses a Range Attribute Map*



Here is the SAS code that creates the attribute map data set and runs the SGPANEL procedure.

A DATA step creates the range attribute map data set. The data set contains one attribute map, which is specified by the ID variable. The data set defines three ranges. The third range uses the COLORMODEL1 – COLORMODEL3 variables to specify a custom gradient of three colors.

```
data clrresp;
retain id "myid";
length min $ 5 max $ 5;
input min $ max $ color $ altcolor $ colormodel1 $ colormodel2 $
colormodel3 $;
datalines;
_min_ 90    purple purple .     .       .
```

```
90    100   gold   gold   .    .      .
100   _max_ .      .      red  orange yellow
;
run;
```

This PROC step generates the graph. The SGPANEL statement references the SASHELP.CLASS data set and the CLRRESP attribute map data set. The plot statement references the attribute map ID (MYID).

```
proc sgpanel data=sashelp.class rattrmap=clrresp;
  panelby sex;
  vbar age / colorresponse=weight rattrid=myid
    response=weight stat=mean colorstat=mean;
run;
```

PART 7

SG Annotation

Chapter 20
Annotating ODS Graphics 1935

Chapter 21
SG Annotation Function Dictionary 1945

Chapter 22
SG Annotation Macro Dictionary 2007

1934

20

Annotating ODS Graphics

Overview of SG Annotation **1935**

SG Annotation Data Sets **1936**
About the SG Annotation Data Set 1936
SG Annotation Variables 1938
SG Annotation Functions 1939
Using Variables as Arguments 1940

Specifying an SG Annotation Data Set in an SG Procedure Statement **1940**

Controlling the Drawing Space **1941**
About the Drawing Space 1941
Drawing Space Variables 1943
Drawing Space Values 1944

Overview of SG Annotation

The ODS Statistical Graphics (SG) procedures provide a mechanism for adding shapes, images, and other annotations to graph output.

You can add the following annotation elements to a graph:

- text labels
- lines and arrows
- ovals (including circles)
- rectangles (including squares)
- polygons
- images

There are two main tasks required to add annotation elements to a graph:

1 Create an SG annotation data set, which contains the commands for creating the annotation elements. For more information, see "About the SG Annotation Data Set " on page 1936 .

Note: Annotation data sets that were created for SAS/GRAPH cannot be used with the ODS Graphics Annotate facility. The variables that are required for the ODS Graphics Annotate Facility differ from those that are required for SAS/GRAPH.

2 Specify the SG annotation data set in the SG procedure statement. You can use annotation in the SGPLOT, SGPANEL, and SGSCATTER procedures. For more information, see "Specifying an SG Annotation Data Set in an SG Procedure Statement" on page 1940 .

SG Annotation Data Sets

About the SG Annotation Data Set

Once you have determined which annotation you want and how you want it to appear in the output, you can create the SG annotation data set. This data set contains the commands for creating one or more annotation elements. The annotations drawn by these commands can be added to SG procedure output.

Each observation represents a command to draw an annotation element or to continue an element. Reserved keywords are used for functions that specify the drawing operation.

Note: The procedures do not support annotation commands that are stored in a table.

You create the data set using the same methods that you use to create any SAS data set. The main distinctions are that the SG annotation data set uses reserved keywords for its variable names, and each observation represents a command to draw an annotation element.

TIP The most commonly used method for creating data sets is with a DATA step.

Here is an example of an SG annotation data set named LINE. The observations in this data set contain the commands that create a blue text label and a blue line.

Figure 20.1 *LINE SG Annotation Data Set*

Obs	label	textcolor	linecolor	drawspace	function	x1	y1	x2	y2
1	Average Height 62 Inches	blue		datavalue	text	60	64	.	.
2			blue	datavalue	line	50	62	150	62

A blank denotes a missing value for a character variable. A '.' denotes a missing value for a numeric variable.

Each observation in this data set contains complete instructions for drawing an annotation. The value of the FUNCTION variable determines what the observation does. Other variables control how the function is performed.

This list describes each observation in LINE and the task that it performs:

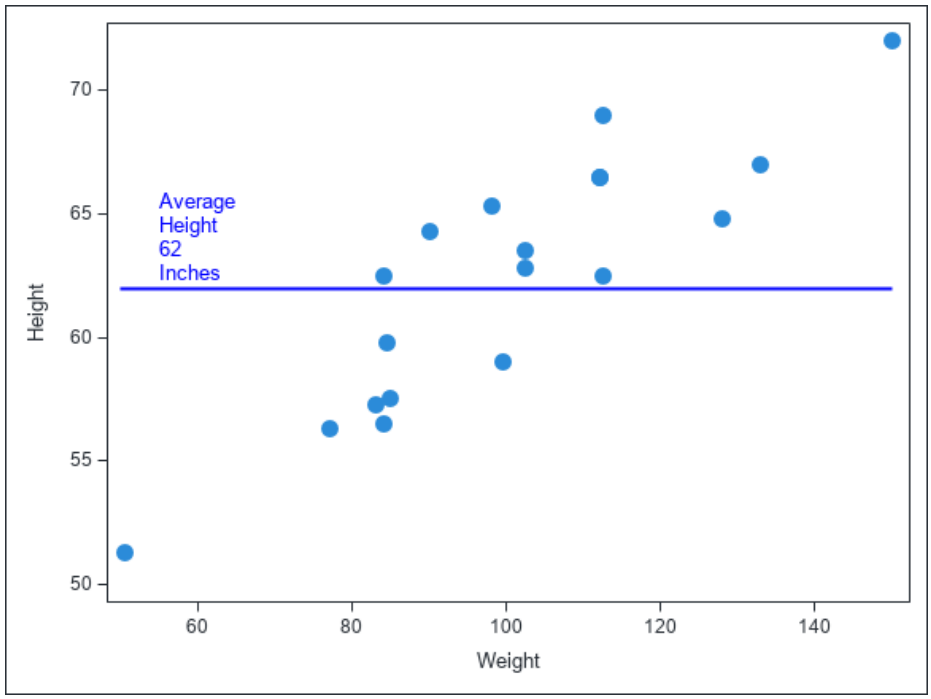
1 This instruction writes a blue text inset at position (60, 64). The value of the FUNCTION variable (TEXT) tells the program what to do. The values of the coordinate variables X1 and Y1 tell where to do it. The value of the attribute variable TEXTCOLOR specifies the color of the text inset.

Note: The default drawing space for annotations is the layout area. In this example, the drawing space has been changed to DATAVALUE, which binds the annotation to a data location along the axis. For more information about drawing spaces, see "Controlling the Drawing Space" on page 1941.

2 This instruction draws a blue line that begins at position (50, 62) and ends at (150, 62). The value of the FUNCTION variable (LINE) tells the program what to do. The values of the coordinate variables X1, Y1, X2, and Y2 tell where to do it. The value of the attribute variable LINECOLOR specifies the color of the line.

The following figure shows the blue text and line annotations created by the LINE data set displayed with an SGPLOT procedure that specifies a scatter plot.

Figure 20.2 LINE SG Annotation Output



Here is the SAS code step that creates the LINE data set and generates the graph.

```
ods graphics / reset;

data Line;
  infile datalines dlm='#';
  length label $ 27
         textcolor $ 9
         linecolor $ 9
         drawspace $ 9;
  retain drawspace "datavalue";
  width=12;
  input function $ x1 y1 label x2 y2 textcolor linecolor;
datalines;
text # 60 # 64 # Average Height 62 Inches # . # . # blue #
line # 50 # 62 #   # 150 # 62 #   # blue
;
run;

proc sgplot data=sashelp.class sganno=Line;
 scatter x=weight y=height;
run;
```

SG Annotation Variables

When an SG annotation data set is processed, the SG procedure looks at the values of specific variables in order to draw annotation elements. Variables in the SG annotation data set have predefined names. In each observation, the procedure

looks only for variables with those names. Other variables can be present, but they are ignored.

The variables can be categorized into the following main types:

- A FUNCTION variable specifies which annotation element to draw. For more information, see "SG Annotation Functions" on page 1939.
- Position variables specify the point or points at which to draw the graphics element.
- Coordinate system variables specify the drawing space to use for the annotation. For more information, see "Controlling the Drawing Space" on page 1941.
- Attribute variables specify the characteristics of the graphics element (for example, color, size, line style, text font).

All variables are described in the context of the functions and macros that support the variables.

SG Annotation Functions

In an SG annotation data set, the FUNCTION variable accepts a set of predefined values (functions) that insert annotation elements into the graph. The value of the FUNCTION variable specifies what drawing action the observation performs. These functions act in conjunction with other variables that determine where and how to perform the action.

The following table summarizes the functions:

Table 20.1 *Summary of SG Annotation Functions*

Function	Description
ARROW	Draws an arrow annotation. For more information, see "ARROW Function" on page 1945.
IMAGE	Specifies a graphic file to use for an image annotation. For more information, see "IMAGE Function" on page 1953.
LINE	Draws a line annotation. For more information, see "LINE Function" on page 1959.
OVAL	Draws an oval or circle annotation. For more information, see "OVAL Function" on page 1965.
POLYCONT	Continues drawing a polygon that was begun with the POLYGON function, or a line that was begun

Function	Description
	with the POLYLINE function. For more information, see “POLYCONT Function” on page 1971.
POLYGON	Specifies the beginning point of a polygon. For more information, see “POLYGON Function” on page 1974.
POLYLINE	Specifies the beginning point of a polyline, which is a connected series of line segments. For more information, see “POLYLINE Function” on page 1979.
RECTANGLE	Draws a rectangle or square annotation. For more information, see “RECTANGLE Function” on page 1984.
TEXT	Places text in the graph output. For more information, see “TEXT Function” on page 1991.
TEXTCONT	Continues a text string. For more information, see “TEXTCONT Function” on page 1998.

Using Variables as Arguments

The function parameters are either numeric or character. Numeric parameters can be numeric constants or numeric column names associated with your data. Similarly, character parameters can be character column names.

When you specify variable column names for parameters, the DATA step for the annotation data set must read the observations from your data. This enables you to plot a series of labels, arrows, or other annotation based on the values in the data. See “Example 1: Create Bar Labels” on page 2070.

Specifying an SG Annotation Data Set in an SG Procedure Statement

In order to use an SG annotation data set with an SG procedure, you must add the following option in the SG procedure statement:

SGANNO= *annotation-data-set*
specifies the SG annotation data set that you want to use. For more information about SG annotation data sets, see "SG Annotation Data Sets" on page 1936.

You can also create margins around the graph by using the PAD= option.

PAD=*dimension<units>* | (*pad-options*)
specifies the amount of extra space that is reserved along the edges of the graph. This option creates margins around the graph for company logos, annotated notes, and so on. You can also specify the unit of measurement. The default unit is pixels. For a list of measurement units that are supported, see "Units of Measurement" on page 1836.

Use pad options to create non-uniform padding. Edges that are not assigned padding are padded with the default amount.

pad-options can be one or more of the following:

LEFT= *dimension<units>*
specifies the amount of extra space to add to the left edge.

RIGHT= *dimension<units>*
specifies the amount of extra space to add to the right edge.

TOP= *dimension<units>*
specifies the amount of extra space to add to the top edge.

BOTTOM= *dimension<units>*
specifies the amount of extra space to add to the bottom edge.

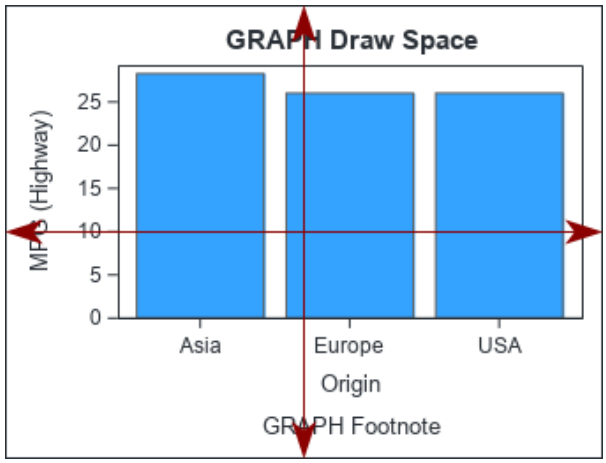
Controlling the Drawing Space

About the Drawing Space

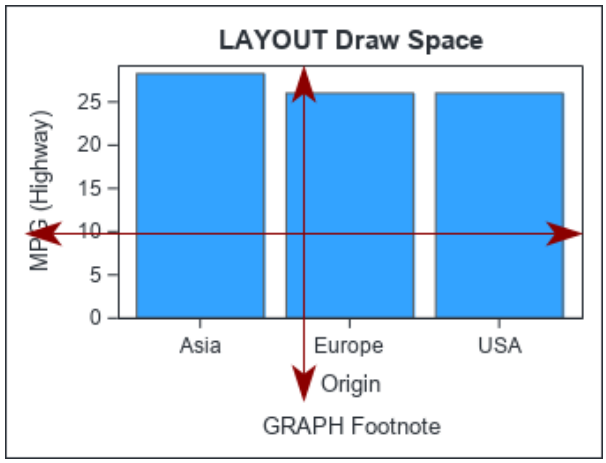
You can control the position and scaling of your annotations by specifying the drawing space and units in the SG annotation data set. You have the option to specify the drawing space for the X or Y axes individually, or for both axes.

All annotations are rendered in one of four drawing areas. In the figures, arrows indicate the extent of the horizontal and vertical drawing space:

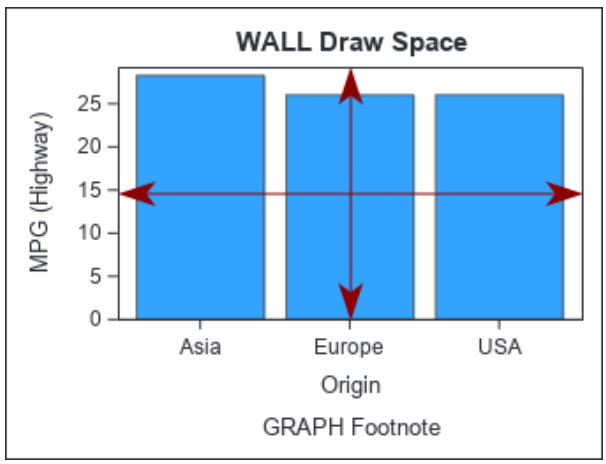
graph area
the entire region of the graph image. The graph area contains the axes, titles, footnotes, legends, and one or more cells.



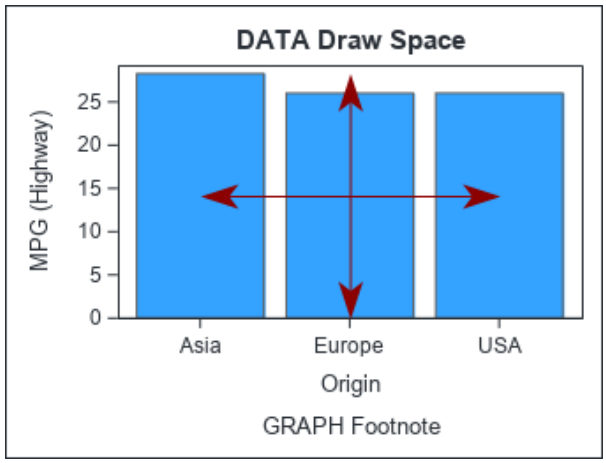
layout area
: the area of the layout, not including any titles and footnotes. The layout area contains the axes, legends, and one or more cells.



wall area
: the area within the axes, including the axis offsets. (This area is not available when using the SGPANEL and SGSCATTER procedures.)



data area
: the area within the axes, **not including** the axis offsets. (This area is not available when using the SGPANEL and SGSCATTER procedures.)



Annotations can be rendered in the drawing areas using one of three units:

- percentage
- pixels
- data value (data area only)

Drawing Space Variables

You can specify the drawing space values for the following variables in the SG annotation data set:

Table 20.2 *Data Set Variables That Are Used to Specify Drawing Space*

Variable	Description	Supported Functions
DRAWSPACE	specifies the drawing space for the annotation	all except POLYCONT and TEXTCONT
X1SPACE	specifies the drawing space of the annotation's X coordinate.	all except TEXTCONT
X2SPACE	specifies the drawing space of the annotation's second X coordinate.	ARROW, LINE
Y1SPACE	specifies the drawing space of the annotation's Y coordinate.	all except TEXTCONT
Y2SPACE	specifies the drawing space of the annotation's second Y coordinate.	ARROW, LINE

Drawing Space Values

Here are the values that are used in the SG annotation data set to control the drawing space.

Note: For more information about the data, graph, layout, or wall areas, see "About the Drawing Space" on page 1941.

DATAPERCENT
positioned and scaled as a percentage with respect to the data area.

DATAPIXEL
positioned and scaled as pixels with respect to the data area.

DATAVALUE
positioned and scaled with respect to the data values.

GRAPHPERCENT
positioned and scaled as a percentage of the graph area.

GRAPHPIXEL
positioned and scaled as pixels with respect to the graph area.

LAYOUTPERCENT
positioned and scaled as a percentage of the layout area.

Note: LAYOUTPERCENT is the default value.

LAYOUTPIXEL
positioned and scaled as pixels with respect to the layout area.

WALLPERCENT
positioned and scaled as a percentage of the wall area.

WALLPIXEL
positioned and scaled as pixels with respect to the wall area.

Note: For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

21

SG Annotation Function Dictionary

Dictionary **1945**
ARROW Function 1945
IMAGE Function 1953
LINE Function 1959
OVAL Function 1965
POLYCONT Function 1971
POLYGON Function 1974
POLYLINE Function 1979
RECTANGLE Function 1984
TEXT Function 1991
TEXTCONT Function 1998

Examples **2000**
Create Labeled Arrows in a Paneled Graph 2000
Create Annotations for Data Points on a Graph 2000
Create Custom Bar Labels 2000

Dictionary

ARROW Function

Draws an arrow annotation from the specified (X1, Y1) coordinates to the specified (X2,Y2) coordinates.

Syntax

FUNCTION= “ARROW”;

Required Variables

X1= *numeric-value* | XC1= "*text-string*"
specifies the first X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

X2= *numeric-value* | XC2= "*text-string*"
specifies the second X coordinate of the annotation.

Use one of the following arguments:

X2= *numeric-value*
specifies the X coordinate for numeric data.

XC2= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X2= and XC2= are specified, then X2= takes precedence, and XC2= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the first Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Y2= *numeric-value* | YC2= "*text-string*"
specifies the second Y coordinate of the annotation. Use one of the following arguments:

Y2= *numeric-value*
specifies the Y coordinate for numeric data.

YC2= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y2= and YC2= are specified, then Y2= takes precedence, and YC2= is ignored.

Optional Variables

DIRECTION= "BOTH" | "IN" | "OUT"
specifies the direction for arrows.

"BOTH"
Draws two arrowheads, one at the starting point in the IN direction and one at the ending point in the OUT direction.

"IN"
Draws a single arrowhead at the starting point (X1,Y1) pointing inward, toward the starting point.

"OUT"
Draws a single arrowhead at the ending point (X2,Y2) pointing outward, away from the starting point (X1,Y1).

Default OUT

DISCRETEOFFSET= *numeric-value*
specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

DRAWSPACE= "*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

LAYER= "BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also

available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR= "color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN= "*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT= "*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS= *n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

SCALE= *numeric-value*
specifies a scale factor for the arrowheads. Specify a positive number.

Default determined by the procedure

SHAPE= "BARBED" | "CLOSED" | "FILLED" | "OPEN"
specifies the shape of the arrowheads. Specify one of the following:

"BARBED"
a solid triangle with an indent at the base.

"CLOSED"
an outline of a triangle.

"FILLED"
a solid triangle.

"OPEN"
a triangle that resembles the letter "V".

Default OPEN

TRANSPARENCY= *numeric-value*
specifies the degree of transparency for the annotation.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies a character variable that contains URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

X1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

X2SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's second X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X1SPACE= value, then the first X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS= "X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's first Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

Y2SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y1SPACE= value, then the first Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS= "Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

IMAGE Function

Specifies a graphic file to use for an image annotation. If no (X1, Y1) coordinates are provided, the image appears in the center of the graph.

Syntax

FUNCTION= "IMAGE";

Required Variable

IMAGE= "*file-name*"
specifies the image file to be displayed in the graphics output. Include the complete path and file name. The syntax of external file specifications varies across operating environments.

For a list of image formats that can be used, see "Supported Image Formats for Output Destinations" on page 127.

Optional Variables

ANCHOR= "TOPLEFT" | "TOP" | "TOPRIGHT" | "RIGHT" | "BOTTOMRIGHT" | "BOTTOM" | "BOTTOMLEFT" | "LEFT" | "CENTER"
specifies the anchor position of the annotation. This point is placed on the specified X1 and Y1 positions.

Default CENTER

BORDER= "TRUE" | "FALSE"
turns the border on and off.

Default FALSE

DISCRETEOFFSET= *numeric-value*
specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

DRAWSPACE= "*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

HEIGHT= *numeric-value*
specifies the height of the annotation. Specify a positive number greater than zero. You can use the HEIGHTUNIT variable to specify the unit of measurement.

Default Determined by the system

HEIGHTUNIT= "DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the HEIGHT value.

Default PERCENT

IMAGESCALE= "FIT" | "FITHEIGHT" | "FITWIDTH" | "TILE"
specifies how the image is scaled within the width and height. You can specify the WIDTH and HEIGHT variables.

Defaults FITWIDTH or FITHEIGHT if no size or if one size (either the width or the height) is specified (to preserve aspect)

FIT if width and height are both specified

LAYER= "BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR= "color" | "*style-attribute*"
specifies a color of the border around the image, if displayed. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN= "*line-pattern*"
specifies a line pattern of the border around the image, if displayed. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT= "*style-element*"
specifies a style element for the line. Here is an example of a style element:

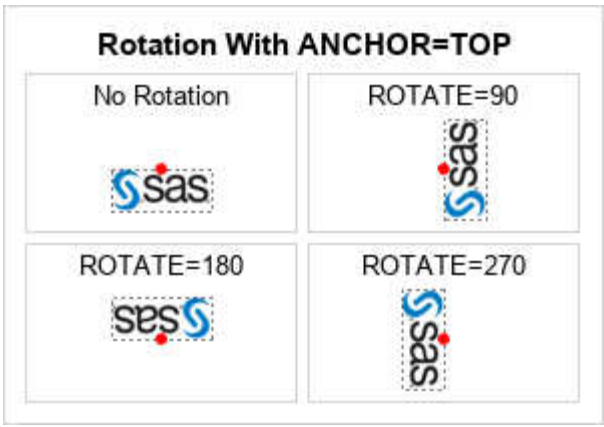
GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS= *n*
specifies the thickness of the border around the image, if displayed. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness).

ROTATE= *degrees*
rotates the image, measured in degrees. The angle is measured as if a horizontal line extended to the right through the image anchor point as shown in the following figure.



Positive angles rotate the image counterclockwise, and negative angles rotate the image clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0

Note When this option is used with a vector graph, the graph is rendered as an image.

TRANSPARENCY= *numeric-value*
specifies the degree of transparency for the annotation.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies a character variable that contains URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

WIDTH= *numeric-value*
specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default Determined by the system

WIDTHUNIT= "DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the WIDTH value.

Default PERCENT

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

X1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS= "X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation. Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

Default 50

YC1= "*text-string*"
specifies the Y coordinate for character data.

Default none

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Y1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS= "Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

LINE Function

Draws a line annotation from the specified (X1, Y1) coordinates to the specified (X2,Y2) coordinates.

Syntax

FUNCTION= "LINE";

Required Variable

X1= *numeric-value* | XC1= "*text-string*"
specifies the first X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

X2= *numeric-value* | XC2= "*text-string*"
specifies the second X coordinate of the annotation.

Use one of the following arguments:

X2= *numeric-value*
specifies the X coordinate for numeric data.

XC2= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X2= and XC2= are specified, then X2= takes precedence, and XC2= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the first Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Y2= *numeric-value* | YC2= "*text-string*"
specifies the second Y coordinate of the annotation. Use one of the following arguments:

Y2= *numeric-value*
specifies the Y coordinate for numeric data.

YC2= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y2= and YC2= are specified, then Y2= takes precedence, and YC2= is ignored.

Optional Variables

DISCRETEOFFSET= *numeric-value*
specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

DRAWSPACE= "*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

LAYER= "BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR= "color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN= "*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT= "*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS= *n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

TRANSPARENCY= *numeric-value*
specifies the degree of transparency for the annotation.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies a character variable that contains URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

X1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

X2SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's second X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X1SPACE= value, then the first X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS= "X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's first Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

Y2SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y1SPACE= value, then the first Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS= "Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

OVAL Function

Draws an oval annotation around the specified (X1, Y1) coordinates using the specified height and width.

Tip: An easy way to draw a circle is to specify PIXEL for the HEIGHTUNIT and WIDTHUNIT variables and then provide the same value for the HEIGHT and WIDTH.

Syntax

FUNCTION= "OVAL";

Required Variables

HEIGHT= *numeric-value*
specifies the height of the annotation. Specify a positive number greater than zero. You can use the HEIGHTUNIT variable to specify the unit of measurement.

Default none

WIDTH= *numeric-value*
specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default none

X1= *numeric-value* | XC1= *"text-string"*
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= *"text-string"*
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

Y1= *numeric-value* | YC1= *"text-string"*
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= *"text-string"*
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Optional Variables

ANCHOR= "TOPLEFT" | "TOP" | "TOPRIGHT" | "RIGHT" | "BOTTOMRIGHT" | "BOTTOM" | "BOTTOMLEFT" | "LEFT" | "CENTER"
specifies the anchor position of the annotation. This point is placed on the specified X1 and Y1 positions.

Default CENTER

DISCRETEOFFSET= *numeric-value*
specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

DISPLAY= "ALL" | "FILL" | "OUTLINE"
specifies the fill and outline properties. You can specify that the annotation be filled (FILL), have an outline (OUTLINE), or both (ALL).

Default OUTLINE

DRAWSPACE= "*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

FILLCOLOR= "color" | "*style-attribute*"
specifies a fill color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default Color attribute of the GraphAnnoShape style element in the current style (GraphAnnoShape:Color)

Interaction This option takes effect only if the DISPLAY option is specified as FILLED or ALL.

FILLSTYLEELEMENT= "*style-element*"
specifies a style element for the fill. Here is an example of a style element:

GraphData2

Note Only the COLOR attribute of the style element applies to the fill.

See "Style Elements for Use with ODS Graphics" on page 1871

FILLTRANSPARENCY= *numeric-value*
specifies the transparency for the fill portion of an annotation.

Default 0.0

Range 0.0 (completely opaque) to 1.0 (completely transparent)

Tip To set both the fill and outline transparency, if specified, use the TRANSPARENCY= option.

HEIGHTUNIT= "DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the HEIGHT value.

Default PERCENT

LAYER= "BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR= "color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN= "*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT= "*style-element*"
specifies a style element for the line. Here is an example of a style element:

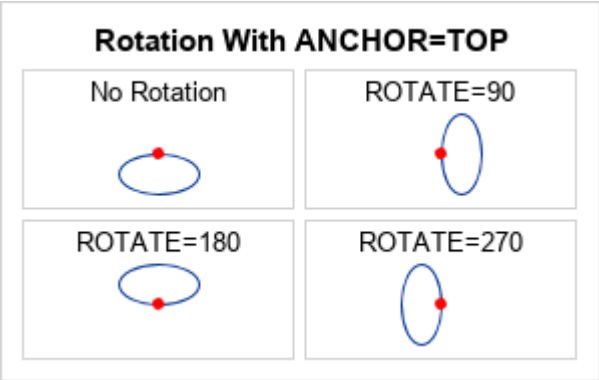
GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS= *n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

ROTATE= *degrees*
rotates the annotation, measured in degrees. The angle is measured as if a horizontal line extended to the right through the oval anchor point as shown in the following figure.



Positive angles rotate the annotation counterclockwise, and negative angles rotate the annotation clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0 (no rotation)

TRANSPARENCY= *numeric-value*
specifies the degree of transparency for the annotation. This option affects both the outline and the fill, if specified.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Tip FILLTRANSPARENCY= overrides the fill transparency specified by this option.

URL=*character-variable*
specifies a character variable that contains URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

WIDTHUNIT= "DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the WIDTH value.

Default PERCENT

X1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS= "X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS= "Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

POLYCONT Function

Continues drawing a polygon that was begun with the POLYGON function, or a line that was begun with the POLYLINE function. POLYCONT specifies each successive point in the polygon or polyline.

Requirements: If you are drawing a polyline, one or more POLYCONT function observations must immediately follow the POLYLINE observation in the SG annotation data set. The POLYLINE and subsequent POLYCONT observations must be contiguous.

If you are drawing a polygon, two or more POLYCONT function observations must immediately follow the POLYGON observation in the SG annotation data set. The POLYGON and subsequent POLYCONT observations must be contiguous. A polygon is similar to a polyline except that the segments of the polygon must form a closed shape.

Syntax

FUNCTION= “POLYCONT”;

Required Variables

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Optional Variables

X1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

Y1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT

- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

POLYGON Function

Specifies the beginning point of a polygon. A polygon is a connected series of lines that form a closed shape. This function is used in conjunction with two or more POLYCONT functions to define vertex points for the polygon. Associated variables can define the fill pattern and color, as well as the line type that outlines the polygon.

Requirement: The SG annotation data set must also specify at least two instances of the POLYCONT function immediately after the POLYGON function.

Syntax

FUNCTION= "POLYGON";

Required Variable

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Optional Variables

DISCRETEOFFSET= *numeric-value*
specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

DISPLAY= "ALL" | "FILL" | "OUTLINE"
specifies the fill and outline properties. You can specify that the annotation be filled (FILL), have an outline (OUTLINE), or both (ALL).

Default OUTLINE

DRAWSPACE= "*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

FILLCOLOR= "color" | "*style-attribute*"
specifies a fill color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default Color attribute of the GraphAnnoShape style element in the current style (GraphAnnoShape:Color)

Interaction This option takes effect only if the DISPLAY option is specified as FILLED or ALL.

FILLSTYLEELEMENT= "*style-element*"
specifies a style element for the fill. Here is an example of a style element:

GraphData2

Note Only the COLOR attribute of the style element applies to the fill.

See "Style Elements for Use with ODS Graphics" on page 1871

FILLTRANSPARENCY= *numeric-value*
specifies the transparency for the fill portion of an annotation.

Default 0.0

Range 0.0 (completely opaque) to 1.0 (completely transparent)

Tip To set both the fill and outline transparency, if specified, use the TRANSPARENCY= option.

LAYER= "BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR= "color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN= "*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT= "*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS= *n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

TRANSPARENCY= *numeric-value*
specifies the degree of transparency for the annotation. This option affects both the outline and the fill, if specified.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Tip FILLTRANSPARENCY= overrides the fill transparency specified by this option.

URL=*character-variable*
specifies a character variable that contains URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

X1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS= "X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT

- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS= "Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

POLYLINE Function

Specifies the beginning point of a polyline, which is a connected series of line segments. This function is used in conjunction with one or more POLYCONT functions to draw a series of connected straight lines.

Requirement: The SG annotation data set must also specify at least one instance of the POLYCONT function immediately after the POLYLINE function.

Note: The POLYLINE object is similar to a POLYGON, except that the POLYLINE object does not need to be a closed shape.

Syntax

FUNCTION= "POLYLINE";

Required Variables

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Optional Variables

DISCRETEOFFSET= *numeric-value*
specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

DRAWSPACE= "*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT

- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

LAYER= "BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR= "color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN= "*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT= "*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS= *n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

TRANSPARENCY= *numeric-value*
specifies the degree of transparency for the annotation.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies a character variable that contains URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

X1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT

- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS= "X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS= "Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

RECTANGLE Function

Draws a rectangle annotation around the specified (X1, Y1) coordinates using the specified height and width. By default, the (X1, Y1) coordinates constitute the center of the rectangle, though you can change this behavior using the ANCHOR option.

Tip: An easy way to draw a square is to specify PIXEL for the HEIGHTUNIT and WIDTHUNIT variables and then provide the same value for the HEIGHT and WIDTH.

Syntax

FUNCTION= “RECTANGLE”;

Required Variables

HEIGHT= *numeric-value*
specifies the height of the annotation. Specify a positive number greater than zero. You can use the HEIGHTUNIT variable to specify the unit of measurement.

Default none

WIDTH= *numeric-value*
specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default none

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Optional Variables

ANCHOR= "TOPLEFT" | "TOP" | "TOPRIGHT" | "RIGHT" | "BOTTOMRIGHT" | "BOTTOM" | "BOTTOMLEFT" | "LEFT" | "CENTER"
specifies the anchor position of the annotation. This point is placed on the specified X1 and Y1 positions.

Default CENTER

CORNERRADIUS= *numeric-value*
specifies the roundness of the corners of rectangles when used with SG procedures other than SGSCATTER.

Specify a value from 0.0 (completely rectangular) to 1.0 (oval).

Figure 21.1 *CORNERRADIUS= Examples*

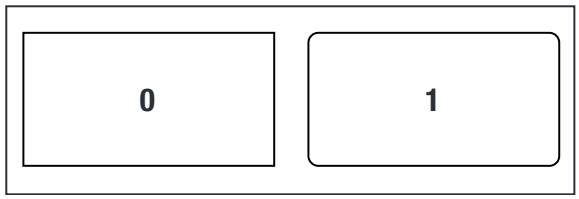
Default 0.0

Restriction This syntax does not apply to the SGSCATTER procedure. For the SGSCATTER procedure, use 0 | 1 instead.

CORNERRADIUS= 0 | 1
specifies whether corners are square (0) or rounded (1) when used with the SGSCATTER procedure.

For rounded corners, the radius is fixed.

Figure 21.2 *CORNERRADIUS= Examples*



Default 0

Restriction This syntax applies to the SGSCATTER procedure only. For the SGPLOT, SGPANEL, and SGRENDER procedures, use *numeric-value* instead.

Note Any value that is greater than 0 and less than or equal to 1 is interpreted as 1.

DISCRETEOFFSET= ***numeric-value***
specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

DISPLAY= "ALL" | "FILL" | "OUTLINE"
specifies the fill and outline properties. You can specify that the annotation be filled (FILL), have an outline (OUTLINE), or both (ALL).

Default OUTLINE

DRAWSPACE= "*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT

- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

FILLCOLOR= "color" | "*style-attribute*"
specifies a fill color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default Color attribute of the GraphAnnoShape style element in the current style (GraphAnnoShape:Color)

Interaction This option takes effect only if the DISPLAY option is specified as FILLED or ALL.

FILLSTYLEELEMENT= "*style-element*"
specifies a style element for the fill. Here is an example of a style element:

GraphData2

Note Only the COLOR attribute of the style element applies to the fill.

See "Style Elements for Use with ODS Graphics" on page 1871

FILLTRANSPARENCY= *numeric-value*
specifies the transparency for the fill portion of an annotation.

Default 0.0

Range 0.0 (completely opaque) to 1.0 (completely transparent)

Tip To set both the fill and outline transparency, if specified, use the TRANSPARENCY= option.

HEIGHTUNIT= "DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the HEIGHT value.

Default PERCENT

LAYER= "BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR= "color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN= "*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT= "*style-element*"
specifies a style element for the line. Here is an example of a style element:

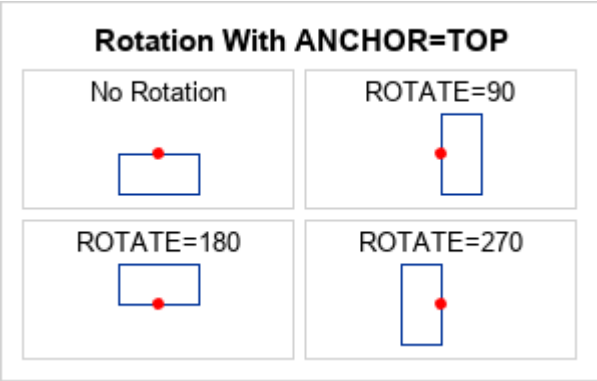
GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS= *n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

ROTATE= *degrees*
rotates the annotation, measured in degrees. The angle is measured as if a horizontal line extended to the right through the rectangle anchor point as shown in the following figure.



Positive angles rotate the annotation counterclockwise, and negative angles rotate the annotation clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0 (no rotation)

TRANSPARENCY= *numeric-value*
specifies the degree of transparency for the annotation. This option affects both the outline and the fill, if specified.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Tip FILLTRANSPARENCY= overrides the fill transparency specified by this option.

URL=*character-variable*
specifies a character variable that contains URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

WIDTHUNIT= "DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the WIDTH value.

Default PERCENT

X1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS= "X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS= "Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

TEXT Function

Places text in the graph output. Associated variables can control the color, size, font, base angle, and rotation of the characters displayed. If you do not supply the (X1, Y1) coordinates, the text is placed in the center of the graph.

Syntax

FUNCTION= "TEXT";

Required Variable

LABEL= "*text-string*"
specifies the text label. The *text-string* can contain superscripts, subscripts, and Unicode characters by using ODS escapement notation. Here is a simple example:

```
label="This label is rich(*ESC*){sup '2'}"
```

You can invoke a macro function that returns a string. For example, consider the following macro, %LABELTEXT:

```
%macro labeltext;
   This label is rich(*ESC*){sup '2'}
%mend labeltext;
```

When invoked, this macro returns the following string:

```
This label is rich(*ESC*){sup '2'}
```

Here is an example of how you invoke this macro in your annotate DATA step code:

```
label="%labeltext";
```

You can add arguments to your macro that enable you to pass in information that is included in the label text. For more information about SAS macros, see *SAS Macro Language: Reference*.

Some combinations of characters can result in warning messages written to the log. The actual warning varies with the types of characters. For example, `label="95&Conf Interval"` and `label="95%Conf Interval"` both produce a warning, although the annotation is drawn in both cases. There are several ways to avoid the warning message:

- add a space after the % or & character.
- escape the % character with a second % character. For example: `label="95% %Conf Interval"`
- use the %NRQUOTE macro function. For example: `label="95%nrquote(&)Conf Interval"`

Default none

Optional Variables

ANCHOR= "TOPLEFT" | "TOP" | "TOPRIGHT" | "RIGHT" | "BOTTOMRIGHT" | "BOTTOM" | "BOTTOMLEFT" | "LEFT" | "CENTER"
specifies the anchor position of the annotation. This point is placed on the specified X1 and Y1 positions.

Default CENTER

BORDER= "TRUE" | "FALSE"
turns the border on and off.

Default FALSE

DISCRETEOFFSET= *numeric-value*
specifies an amount to offset the annotation from a discrete value in data space. Specify a value from -0.5 (left offset) to +0.5 (right offset).

Default 0.0 (no offset)

DRAWSPACE= "*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify X1SPACE and Y1SPACE individually.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT

- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

FILLCOLOR= "color" | "*style-attribute*"
specifies the background color for the text annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The Color attribute of the GraphAnnoShape style element in the current style (GraphAnnoShape:Color)

FILLTRANSPARENCY= *numeric-value*
specifies the transparency for the fill portion of an annotation.

Default 0.0

Range 0.0 (completely opaque) to 1.0 (completely transparent)

Tip To set both the fill and border transparency, if specified, use the TRANSPARENCY= option.

JUSTIFY= "CENTER" | "LEFT" | "RIGHT"
specifies the text justification.

Default LEFT

LAYER= "BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR= "color" | "*style-attribute*"
specifies a line color for the border of the text annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

Interaction This option takes effect only if BORDER= "TRUE".

LINEPATTERN= "*line-pattern*"
specifies a line pattern for the border of the text annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

Interaction This option takes effect only if BORDER= "TRUE".

LINESTYLEELEMENT= "*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

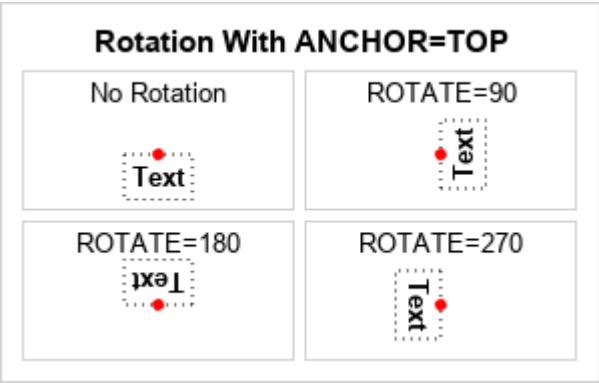
See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS= *n*
specifies the line thickness of the border of the text annotation. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

Interaction This option takes effect only if BORDER= "TRUE".

ROTATE= *degrees*
rotates the annotation, measured in degrees. The angle is measured as if a horizontal line extended to the right through the text box anchor point as shown in the following figure.



Positive angles rotate the annotation counterclockwise, and negative angles rotate the annotation clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0 (no rotation)

TEXTCOLOR= "color" | "*style-attribute*"
specifies the text color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default COLOR attribute of the GraphAnnoText style element in the current style (GraphAnnoText:Color)

TEXTFONT= "*font-family*"
specifies the font family for the annotation. The SAS ODS styles use TrueType system fonts. For more information, see "Fonts" on page 1819.

Default FontFamily attribute of the GraphAnnoText style element in the current style.

TEXTSIZE= *n*
specifies the font size of the annotation.

Default FontSize attribute of the GraphAnnoText style element in the current style.

TEXTSTYLE= "ITALIC" | "NORMAL"
specifies whether the annotation characters are italic (ITALIC) or normal (NORMAL).

Default FontStyle attribute of the GraphAnnoText style element in the current style.

TEXTSTYLEELEMENT= "*style-element*"
specifies a style element for the text. Here is an example of a style element:

GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

TEXTWEIGHT= "BOLD" | "NORMAL"
specifies whether the annotation characters are bold (BOLD) or normal (NORMAL).

Default FontWeight attribute of the GraphAnnoText style element in the current style.

TRANSPARENCY= *numeric-value*
specifies the degree of transparency for the annotation.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*character-variable*
specifies a character variable that contains URLs for web pages to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

WIDTH= *numeric-value*
specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default Determined by the system

WIDTHUNIT= "DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the WIDTH value.

Default PERCENT

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

X1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS= "X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation. Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

Default 50

YC1= "*text-string*"
specifies the Y coordinate for character data.

Default none

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Y1SPACE= "*draw-space-value*"
specifies the drawing space of the annotation's Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT

- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS= "Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

TEXTCONT Function

Continues a text string that was started using the TEXT function. You can set the continued text apart from the main text by specifying different text attributes.

Requirement: To continue the text, immediately follow the initial TEXT observation with one or more TEXTCONT observations. The TEXT and subsequent TEXTCONT observations must be contiguous.

Syntax

FUNCTION= "TEXTCONT";

Required Variable

LABEL= "*text-string*"
specifies the text label. The *text-string* can contain superscripts, subscripts, and Unicode characters by using ODS escapement notation. Here is a simple example:

```
label="This label is rich(*ESC*){sup '2'}"
```

You can invoke a macro function that returns a string. For example, consider the following macro, %LABELTEXT:

```
%macro labeltext;
   This label is rich(*ESC*){sup '2'}
%mend labeltext;
```

When invoked, this macro returns the following string:

```
This label is rich(*ESC*){sup '2'}
```

Here is an example of how you invoke this macro in your annotate DATA step code:

```
label="%labeltext";
```

You can add arguments to your macro that enable you to pass in information that is included in the label text. For more information about SAS macros, see *SAS Macro Language: Reference*.

Some combinations of characters can result in warning messages written to the log. The actual warning varies with the types of characters. For example, `label="95&Conf Interval"` and `label="95%Conf Interval"` both produce a warning, although the annotation is drawn in both cases. There are several ways to avoid the warning message:

- add a space after the % or & character.
- escape the % character with a second % character. For example: `label="95% %Conf Interval"`
- use the %NRQUOTE macro function. For example: `label="95%nrquote(&)Conf Interval"`

Default none

Optional Variables

TEXTCOLOR= "color" | "*style-attribute*"
specifies the text color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default COLOR attribute of the GraphAnnoText style element in the current style (GraphAnnoText:Color)

TEXTFONT= "*font-family*"
specifies the font family for the annotation. The SAS ODS styles use TrueType system fonts. For more information, see "Fonts" on page 1819.

Default FontFamily attribute of the GraphAnnoText style element in the current style.

TEXTSIZE= *n*
specifies the font size of the annotation.

Default FontSize attribute of the GraphAnnoText style element in the current style.

TEXTSTYLE= "ITALIC" | "NORMAL"
specifies whether the annotation characters are italic (ITALIC) or normal (NORMAL).

Default FontStyle attribute of the GraphAnnoText style element in the current style.

TEXTSTYLEELEMENT= "*style-element*"
specifies a style element for the text. Here is an example of a style element:

GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

TEXTWEIGHT= "BOLD" | "NORMAL"
specifies whether the annotation characters are bold (BOLD) or normal (NORMAL).

Default FontWeight attribute of the GraphAnnoText style element in the current style.

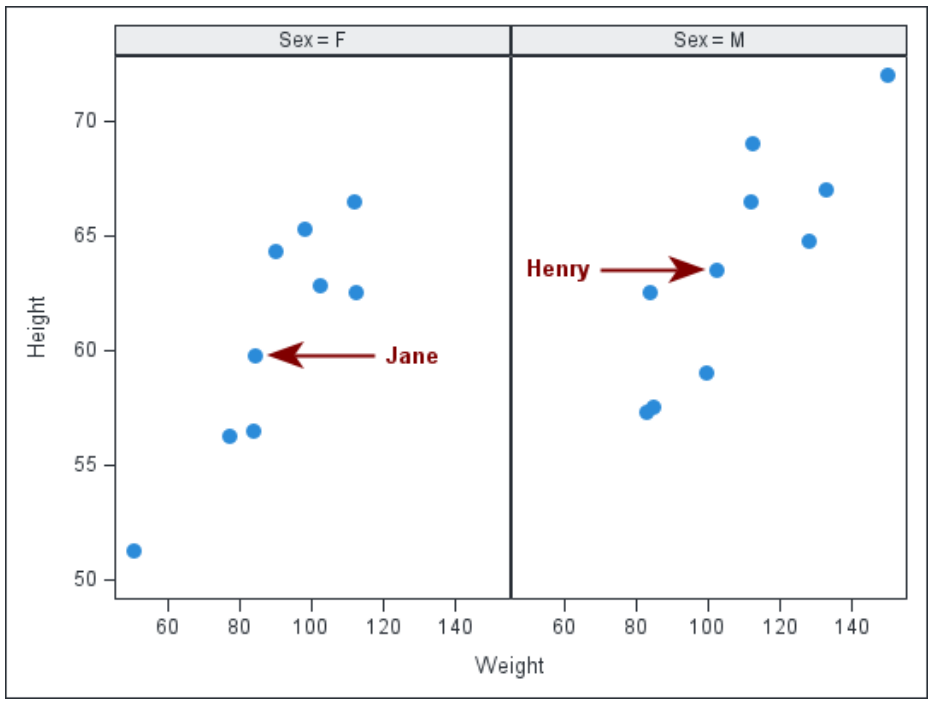
Examples

Example 1: Create Labeled Arrows in a Paneled Graph

Features: ARROW
TEXT

This example shows you how to use the ODS Graphics Annotate facility to draw labeled arrows in a panel of scatter plots. The panel is created by using the SGPANEL procedure to panel variables Height and Weight in data set Sashelp.Class by variable Sex. A labeled arrow is used to annotate a specific point in each scatter plot in the panel. The label for each arrow indicates the name that is associated with that data point.

Output



Program

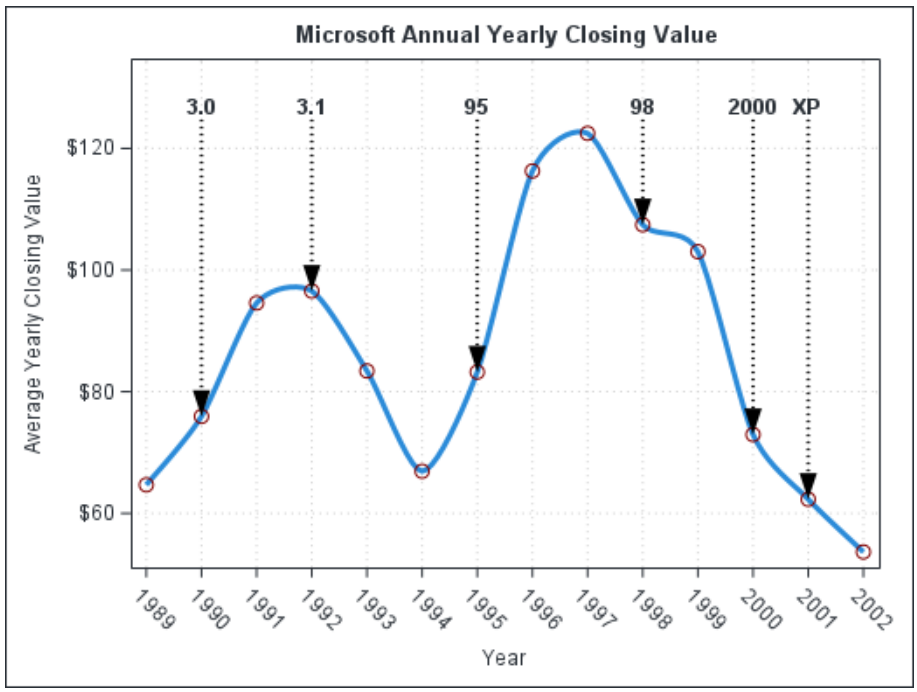
```
data annoArrow;                                          /* 1 */
   retain drawspace "layoutpercent" linecolor "maroon"
      textcolor "maroon" textweight "bold";
   input function $1-5 x1 y1 x2 y2 shape $25-30 direction $32-34
      anchor $36-40 label $42-60;
datalines;
arrow 26   49.5 38 49.5 barbed in
text  39   49.5 .  .               left  Jane
arrow 65   62.7 77 62.7 barbed out
text  64.5 62.7 .  .               right Henry
;
run;
proc sgpanel data=sashelp.class sganno=annoArrow;    /* 2 */
   panelby sex;
   scatter x=weight y=height;
run;
```

1 Data set AnnoArrow stores the annotation data. The DRAWSPACE variable specifies LAYOUTPERCENT. All annotation coordinates are expressed as a percentage of the layout space. Function ARROW is used to draw the arrows, and function TEXT is used to draw the label for each arrow.

2 The SGPANEL procedure generates the graph. Option SGANNO= in the PROC SGPANEL statement specifies the annotation data set.

Example 2: Create Annotations for Data Points on a Graph

This example shows you how to use the ODS Graphics annotation facility to annotate data points on a series plot. The series plot shows the average yearly closing value of Microsoft stock from 1989 to 2002. Annotations identify the average closing price in the year in which each major version of Microsoft Windows was released. Here is the output for this example.



Example Program

```
data MSreleases;                    /* 1 */
  input Year Release $5.;
datalines;
1990 3.0
1992 3.1
1995 95
1998 98
```

```
2000 2000
2001 XP
;

data MSstock;                          /* 2 */
   set sashelp.stocks(keep=date stock close);
   year=year(date);
   where stock eq "Microsoft" and year(date) between 1989 and 2002;
   drop stock;
run;

proc summary data=MSstock nway;    /* 3 */
  class year;
  var close;
  output out=MSMeanClose mean=MeanClose;
run;

proc sql;                              /* 4 */
   create table _MSAnno as
   select Year, Release, MeanClose
   from MSreleases(rename=(Year=RelYear)), MSMeanClose
   where RelYear eq Year;
run;
quit;

data MSAnno;                           /* 5 */
   id="anno"; drawspace="layoutpercent"; x1space="DATAVALUE";
   x2space="DATAVALUE"; y2space="DATAVALUE"; linecolor="black";
   linepattern=34; textweight="bold"; shape="filled"; scale=0.5; y1=90;
   set _MSAnno(rename=(Year=x1 MeanClose=y2 Release=Label));
   function="arrow"; x2=x1;
   output;
   y1=92; function="text";
   output;
run;

                                       /* 6 */
title "Microsoft Annual Yearly Closing Value";
proc sgplot data=MSMeanClose
   sganno=MSAnno;
   series x=year y=MeanClose / markers smoothconnect
      markerattrs=(symbol=circle size=7pt color=maroon);
   yaxis label="Average Yearly Closing Value" offsetmax=0.15
      grid gridattrs=(color=lightgray pattern=dot);
   xaxis label="Year" type=discrete grid
      gridattrs=(color=lightgray pattern=dot);
run;
```

1 Data set MSReleases stores the Microsoft Windows release information.

2 Data set MSStock stores the Microsoft stock closing values for 1989 through 2002.

3 The SUMMARY procedure summarizes the closing values for each year and stores the result in data set MSMeanClose.

4 The SQL procedure adds the closing value from data set MSMeanClose to the data in data set MSReleases and stores the result in data set _MSAnno.

5 The MSAnno DATA step adds the required annotate columns to the data in data set _MSAnno.

6 The SGPLOT procedure generates the graph. Option SGANNO= specifies the annotation data set, MSAnno. The SERIES statement generates the series plot.

Example 3: Create Custom Bar Labels

Overview

This example demonstrates how to create custom category labels for a horizontal bar chart. The bar chart in this example shows the average quarterly closing value of IBM stock for 2001 and 2002. It also shows the change in value from the previous quarter for the second and subsequent quarters. The quarter is the category, and the average closing value is the response. The category values are formatted by using format yyq, which displays the date values as *yyyy*Q*n*. A user-defined format could be created to provide more readable date values, but it requires a significant amount of space to fit the category axis tick values. To conserve space, more readable annotations are placed on the bars to replace the category axis tick values. Here is the output for this example.

Program

```
proc sort data=sashelp.stocks out=stocks;            /* 1 */
```

```
   where year(date) in (2001 2002) and stock eq "IBM";
   by date stock;
run;

proc summary data=stocks;
   var close;
   by date;
   format date yyq.;
   output out=summary mean=avg_close;
run;

data close_summary;                                          /* 2 */
   retain prev_close;
   set summary;
   length change $8 changepct $8 changetype $10;
   label avg_close = "Close" change = "Change"
      changepct = "Percent" changetype = "Change Type";
   format date yyq. avg_close dollar6.1;
   if _N_ = 1 then do;
      _change = 0;
      prev_close = avg_close;
      change = "---";   /* Not applicable */
      changepct = "---";
      changetype = "INITIAL";
   end;
   else do;
      _change = avg_close - prev_close;;
      change = strip(put(_change, dollar6.1));
      changepct = strip(put(_change / prev_close, percentn8.1));
      select;
         when(_change > 0) changetype = "UP";
         when(_change < 0) changetype = "DOWN";
         otherwise do;
            changetype = "UNCH";
            change = "$0";
            changepct = "---";
         end;
      end;
      prev_close = avg_close;
   end;
   drop prev_close _change;
run;

data chgattrmap;                                             /* 3 */
   length ID value $8 textcolor $ 12;
   input ID $ value $ textcolor $;
datalines;
chg  UP      darkgreen
chg  DOWN    darkred
chg  UNCH    darkblue
chg  INITIAL black
;
run;

data anno;                                                   /* 4 */
retain function "text" x1 2 x1space "wallpercent" y1space "datavalue"
```

```
   anchor "left" textweight "bold" textcolor "darkslategray"
   textfont "Arial" textsize 10 width 200;
length id $12;
id = "BAR";
input yc1 $ 1-6 label $ 8-26;
datalines;
2001Q1 First Quarter 2001
2001Q2 Second Quarter 2001
2001Q3 Third Quarter 2001
2001Q4 Fourth Quarter 2001
2002Q1 First Quarter 2002
2002Q2 Second Quarter 2002
2002Q3 Third Quarter 2002
2002Q4 Fourth Quarter 2002
;
run;
                                                        /* 5 */
title "Average Quarterly IBM Stock Value in 2001 and 2002";
proc sgplot data=close_summary dattrmap=chgattrmap sganno=anno;
   hbarparm category=date response=avg_close / dataskin=matte
               fillattrs=(color=LightSteelBlue transparency=0.3);
   yaxistable avg_close change changepct / attrid=chg
                  location=inside separator nomissingchar
                  valueattrs=(size=10pt) valuejustify=right
                  colorgroup=changetype;
   xaxis type=linear values = (20 to 120 by 20);
   yaxis type=discrete display=none colorbands=even
      colorbandsattrs=(color=cxf0f0f0);
run;
title;
```

1 The SORT procedure sorts the data for 2001 and 2002 by date and stock. The SUMMARY procedure summarizes the mean closing price by date. Format YYQ is applied to the date column. The result is stored in data set Avg_Close.

2 This DATA step adds columns Change and ChangeType to the summarized data. Column Change stores the amount by which the stock value changed from the last quarter for the second and subsequent quarters. Column ChangeType stores the type of change as INITIAL, UP, DOWN, or UNCH. The ChangeType values are used later to determine the color of the change values.

3 This DATA step creates an attribute map for the change type values.

4 This DATA step creates the annotation data set.The annotation data set provides more readable labels for the quarterly dates. The YC1 column specifies the formatted category values. The values in column YC1 must exactly match the formatted date values in column Date. If a value in YC1 does not match any of the formatted values in column Date, that annotation is dropped.

5 The SGPLOT generates the bar chart.Data set ChgAttrMap is specified as the attribute data set, and data set Anno is specified as the annotation data set. Because the data is summarized, the HBARPARM statement is used to the generate the bar chart. The YAXISTABLE statement displays the change values in a table to the right of the bars. Column ChangeType specifies the color groups, and attribute map Chg provides the group colors.

22

SG Annotation Macro Dictionary

About the SG Annotation Macros **2007**

Using the SG Annotation Macros **2008**
- Macro Structure 2008
- Making the Macros Available 2008
- Example 2008
- Resetting Parameter Values 2008
- SG Annotation Macro Summary 2008
- Data-Driven Macro Parameters 2008

Dictionary **2011**
- %SGANNO_HELP Annotate Macro 2011
- %SGARROW Annotate Macro 2012
- %SGIMAGE Annotate Macro 2020
- %SGLINE Annotate Macro 2026
- %SGOVAL Annotate Macro 2033
- %SGPOLYCONT Annotate Macro 2039
- %SGPOLYGON Annotate Macro 2042
- %SGPOLYLINE Annotate Macro 2048
- %SGRECTANGLE Annotate Macro 2053
- %SGTEXT Annotate Macro 2060
- %SGTEXTCONT Annotate Macro 2068

Examples **2070**
- Create Bar Labels 2070
- Adding Information to a Graph 2070

About the SG Annotation Macros

You can use SG annotation macros within a SAS DATA step to simplify the process of creating annotation observations. With a macro, you specify a function and assign variable values in one step without having to write explicit variable

assignment statements. You can mix assignment statements and macro calls in the same DATA step.

The following sections describe all of the annotation macros and include information about accessing and using macros.

Using the SG Annotation Macros

Macro Structure

The general form of an SG annotation macro is

```
%MACRO (parameters);
```

In general, the macro name represents a function and the parameters contain the values for the variables that can be used with the function.

The parameters can be specified in any order.

Each of the parameters in the SGANNO macros corresponds to a variable in the SG annotation data set.

The parameters are either numeric or character. Numeric parameters can be numeric constants or numeric variable names that have been initialized to the appropriate value. Character parameters must be enclosed in quotation marks or specified as character variable names. For more information about specifying variable names, see “Data-Driven Macro Parameters” on page 2011.

The SG annotation facility assigns the parameter values to the corresponding annotation variables. Therefore, the observations in an annotation data set that is created with macros look the same as the ones that you create with assignment statements. For example, the following two statements are equivalent:

```
%sgtext (x1=10, y1=15, label="My Text", justify="center");

function="text"; x1=10; y1=15; label="My Text"; justify="center"; output;
```

Making the Macros Available

To use the SG annotation macros, you must compile the macros by issuing the %SGANNO macro:

```
%SGANNO
```

The %SGANNO macro must be run before any other SG annotation macros are used in a SAS session. A message appears in the SAS log indicating that the SG annotation macros are available.

Example

The following sample program identifies the basic elements of an annotation program.

```
%sganno;                                                    /* 1 */
data sgannodata;                                              /* 2*/
  %sgrectangle (x1=50, y1=50, height=30, width=40);          /* 3*/
  %sgarrow     (x1=51, x2=30, y2=35, linecolor="red");
proc sgplot data=sashelp.class sganno=sgannodata;           /* 4*/
  scatter x=age y=weight;
run;
```

1 The %SGANNO macro compiles all of the SG annotation macros and makes them available.

2 The DATA step creates the annotation data.

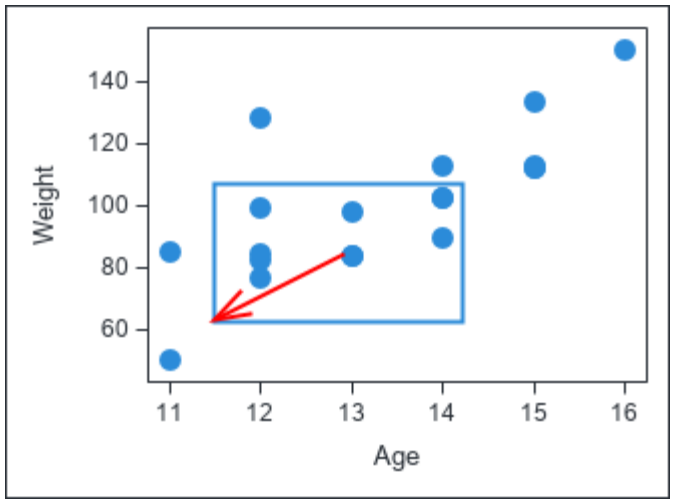
3 The %SGRECTANGLE macro draws a 30-by-40 rectangle, starting from point (50,50).

The %SGARROW macro draws a red arrow from point (51, 50) to point (30, 35). The X1 value overrides the X1 value that was specified in the call to %SGRECTANGLE. The Y1 parameter is not specified, so the arrow uses the Y1 value that was specified in the call to %SGRECTANGLE.

By default, the rectangle and the arrow are positioned and scaled as a percentage of the layout area.

4 The SGPLOT procedure statement uses the SGANNO= option to reference the annotation data set.

Here is the graphics output:



Here is the SGANNODATA data set:

Obs	FUNCTION	LINECOLOR	X1	Y1	HEIGHT	WIDTH	X2	Y2
1	RECTANGLE		50	50	30	40	.	.
2	ARROW	red	51	50	30	40	30	35

Resetting Parameter Values

By default, a macro call retains all variable values that were set in previous macro calls.

You have two options for resetting the values:

- To reset individual variable values on a subsequent macro call, set new values for the corresponding parameters in that macro call.
- To re-initialize all variable values on any macro call, specify RESET="ALL" among the macro's parameters.

For example, the %SGARROW macro draws an arrow between two points and requires values for both points: (x1,y1) and (x2,y2). To draw two arrows from the same starting point but to different end points, you can omit the X1 and Y1 parameters in the second macro call.

```
%sgarrow(x1=34, y1=30, x2=62, y2=64, linecolor="red");
%sgarrow(x2=45, y2=40, linecolor="green");
%sgarrow(reset="all", x1=60, y1=83, x2=79, y2=83);
```

In the previous example:

- The first macro call draws a red arrow from point (34, 30) to point (62, 64).
- The second macro call draws a green arrow from point (34, 30) to point (45, 40).
- The third macro call resets all macro-variable values and draws a default-colored arrow from point (60, 83) to point (79, 83).

SG Annotation Macro Summary

The following table summarizes the tasks performed by the Annotate macros.

Macro	Description
%SGANNO_HELP	displays help for using the SG annotation macros
%SGARROW	draws an arrow

Macro	Description
%SGIMAGE	draws an image
%SGLINE	draws a line from one point to another
%SGOVAL	draws an oval or a circle
%SGPOLYCONT	continues drawing a polygon or a polyline
%SGPOLYGON	draws a polygon
%SGPOLYLINE	draws a polyline
%SGRECTANGLE	draws a rectangle or a square
%SGTEXT	begins drawing text
%SGTEXTCONT	continues drawing text

Data-Driven Macro Parameters

The macro parameters are either numeric or character. Numeric parameters can be numeric constants or numeric variable column names associated with your data. Similarly, character parameters can be character column names.

When you specify variable names for macro parameters, the DATA step for the annotation data set must read the observations from your data. This enables you to plot a series of labels, arrows, or other annotation based on the values in the second data set. For an example of using variable names, see "Example 1: Create Bar Labels" on page 2070.

Dictionary

%SGANNO_HELP Annotate Macro

displays help for using the SG annotation macros.

Syntax

%SGANNO_HELP(*macro-name*) | (ALL)

Required Argument

(*macro-name*) | (ALL)

- To request help for a specific macro, use the form %SGANNO_HELP(*macro-name*), where *macro-name* is the name of the desired macro. For example:

```
%sganno_help(sgrectangle);
```

- To request help for all of the SG annotation macros, use the form %SGANNO_HELP(ALL).

%SGARROW Annotate Macro

Draws an arrow from (X1, Y1) to (X2,Y2).

Requirement: You must run the %SGANNO macro before using any other SG annotation macros. For more information, see "Making the Macros Available" on page 2008.

Syntax

%SGARROW (X1=*numeric-value* | XC1="*text-string*",
Y1=*numeric-value* | YC1="*text-string*",
X2=*numeric-value* | XC2="*text-string*",
Y2=*numeric-value* | YC2="*text-string*",
<*options*>)

Required Arguments

X1= *numeric-value* | XC1= "*text-string*"
specifies the first X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

X2= *numeric-value* | XC2= "*text-string*"
specifies the second X coordinate of the annotation. Use one of the following arguments:

X2= *numeric-value*
specifies the X coordinate for numeric data.

XC2= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X2= and XC2= are specified, then X2= takes precedence, and XC2= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the first Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Y2= *numeric-value* | YC2= "*text-string*"
specifies the second Y coordinate of the annotation.

Use one of the following arguments:

Y2= *numeric-value*
specifies the Y coordinate for numeric data.

YC2= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y2= and YC2= are specified, then Y2= takes precedence, and YC2= is ignored.

Optional Arguments

DIRECTION="BOTH" | "IN" | "OUT"
specifies the direction for arrows.

"BOTH"
Draws two arrowheads, one at the starting point in the IN direction and one at the ending point in the OUT direction.

"IN"
Draws a single arrowhead at the starting point (X1,Y1) pointing inward, toward the starting point.

"OUT"
Draws a single arrowhead at the ending point (X2,Y2) pointing outward, away from the starting point (X1,Y1).

Default OUT

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset the annotation from a discrete value in data space.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset)

DRAWSPACE="*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

ID="*annotation-identifier*"
Defines an ID for the annotation. The ID contains a unique character value that identifies the subset to which each annotation belongs. All annotations in the annotation data set with an ID variable value that matches the specified annotation identifier are drawn. If the annotation data set does not contain an ID variable or if no ID variable value matches the specified identifier, no annotations are drawn.

Default All of the annotations in the SGANNO data set are drawn.

LAYER="BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR="color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN="*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT="*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS=*n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

RESET="ALL"
Re-initializes all variables.

SCALE=*numeric-value*
specifies a scale factor for the arrowheads. Specify a positive number.

Default determined by the procedure

SHAPE="BARBED" | "CLOSED" | "FILLED" | "OPEN"
specifies the shape of the arrowheads. Specify one of the following:

"BARBED"
a solid triangle with an indent at the base.

"CLOSED"
an outline of a triangle.

"FILLED"
a solid triangle.

"OPEN"
a triangle that resembles the letter "V".

Default OPEN

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the annotation.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL="*text-string*"
specifies a URL for a web page to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

X1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

X2SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL

- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X1SPACE= value, then the first X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS="X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT

- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

Y2SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y1SPACE= value, then the first Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS="Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

%SGIMAGE Annotate Macro

displays an image.

Requirement: You must run the %SGANNO macro before using any other SG annotation macros. For more information, see "Making the Macros Available" on page 2008.

Syntax

%SGIMAGE (IMAGE=*"file-specification"* <*x1*, *y1*, options>)

Required Argument

IMAGE="*file-specification*"
specifies the image file to be displayed in the graphics output. Include the complete path and file name. The syntax of external file specifications varies across operating environments.

For a list of image formats that can be used, see "Supported Image Formats for Output Destinations" on page 127.

Optional Arguments

ANCHOR="TOPLEFT" | "TOP" | "TOPRIGHT" | "RIGHT" | "BOTTOMRIGHT" | "BOTTOM" | "BOTTOMLEFT" | "LEFT" | "CENTER"
specifies the anchor position of the annotation. This point is placed on the specified X1 and Y1 positions.

Default CENTER

BORDER="TRUE" | "FALSE"
turns the border on and off.

Default FALSE

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset the annotation from a discrete value in data space.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset)

DRAWSPACE="*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

HEIGHT=*numeric-value*
specifies the height of the annotation. Specify a positive number greater than zero. You can use the HEIGHTUNIT variable to specify the unit of measurement.

Default Determined by the system

HEIGHTUNIT="DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the HEIGHT value.

Default PERCENT

ID="*annotation-identifier*"
Defines an ID for the annotation. The ID contains a unique character value that identifies the subset to which each annotation belongs. All annotations in the annotation data set with an ID variable value that matches the specified annotation identifier are drawn. If the annotation data set does not contain an ID variable or if no ID variable value matches the specified identifier, no annotations are drawn.

Default All of the annotations in the SGANNO data set are drawn.

IMAGESCALE="FIT" | "FITHEIGHT" | "FITWIDTH" | "TILE"
specifies how the image is scaled within the width and height. You can specify the WIDTH and HEIGHT variables.

Defaults FITWIDTH or FITHEIGHT if no size or if one size (either the width or the height) is specified (to preserve aspect)

FIT if width and height are both specified

LAYER="BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR= "color" | "*style-attribute*"
specifies a color of the border around the image, if displayed. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN="*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT="*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

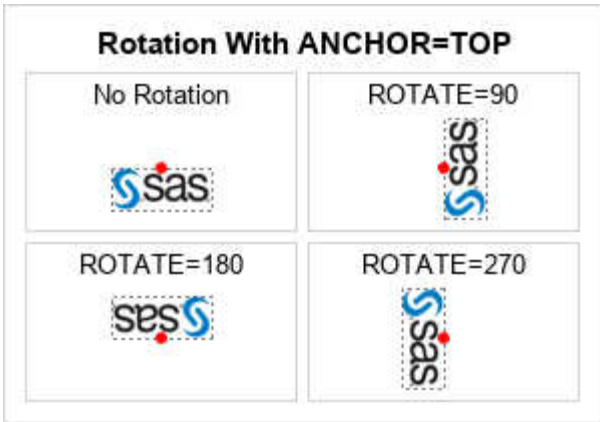
See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS=*n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

RESET="ALL"
Re-initializes all variables.

ROTATE= *degrees*
rotates the image, measured in degrees. The angle is measured as if a horizontal line extended to the right through the image anchor point as shown in the following figure.



Positive angles rotate the image counterclockwise, and negative angles rotate the image clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0

Note When this option is used with a vector graph, the graph is rendered as an image.

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the annotation.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*"text-string"*
specifies a URL for a web page to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

WIDTH=*numeric-value*
specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default Determined by the system

WIDTHUNIT="DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the WIDTH value.

Default PERCENT

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

X1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS="X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Y1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS="Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

%SGLINE Annotate Macro

Draws a line from (X1, Y1) to (X2,Y2).

Requirement: You must run the %SGANNO macro before using any other SG annotation macros. For more information, see "Making the Macros Available" on page 2008.

Syntax

%SGLINE (X1=*numeric-value* | XC1="*text-string*",
Y1=*numeric-value* | YC1="*text-string*",
X2=*numeric-value* | XC2="*text-string*",
Y2=*numeric-value* | YC2="*text-string*",
<*options*>)

Required Arguments

X1= *numeric-value* | XC1= "*text-string*"
specifies the first X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

X2= *numeric-value* | XC2= "*text-string*"
specifies the second X coordinate of the annotation. Use one of the following arguments:

X2= *numeric-value*
specifies the X coordinate for numeric data.

XC2= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X2= and XC2= are specified, then X2= takes precedence, and XC2= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the first Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Y2= *numeric-value* | YC2= "*text-string*"
specifies the second Y coordinate of the annotation.

Use one of the following arguments:

Y2= *numeric-value*
specifies the Y coordinate for numeric data.

YC2= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y2= and YC2= are specified, then Y2= takes precedence, and YC2= is ignored.

Optional Arguments

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset the annotation from a discrete value in data space.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset)

DRAWSPACE="*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

ID="*annotation-identifier*"
Defines an ID for the annotation. The ID contains a unique character value that identifies the subset to which each annotation belongs. All annotations in the annotation data set with an ID variable value that matches the specified

annotation identifier are drawn. If the annotation data set does not contain an ID variable or if no ID variable value matches the specified identifier, no annotations are drawn.

Default All of the annotations in the SGANNO data set are drawn.

LAYER="BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR="color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN="*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT="*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS=*n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

RESET="ALL"
Re-initializes all variables.

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the annotation.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*"text-string"*
specifies a URL for a web page to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

X1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

X2SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X1SPACE= value, then the first X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS="X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

Y2SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y1SPACE= value, then the first Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS="Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

%SGOVAL Annotate Macro

Draws an oval annotation around the specified (X1, Y1) coordinates using the specified height and width.

Requirement: You must run the %SGANNO macro before using any other SG annotation macros. For more information, see "Making the Macros Available" on page 2008.

Syntax

%SGOVAL (HEIGHT=*numeric-value*, WIDTH=*numeric-value*,
X1=*numeric-value* | XC1="*text-string*",
Y1=*numeric-value* | YC1="*text-string*", <*options*>)

Required Arguments

HEIGHT=*numeric-value*
specifies the height of the annotation. Specify a positive number greater than zero. You can use the HEIGHTUNIT variable to specify the unit of measurement.

Default none

WIDTH=*numeric-value*
specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default none

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Optional Arguments

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset the annotation from a discrete value in data space.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset)

DISPLAY="ALL" | "FILL" | "OUTLINE"
specifies the fill and outline properties. You can specify that the annotation be filled (FILL), have an outline (OUTLINE), or both (ALL).

Default OUTLINE

DRAWSPACE="*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

FILLCOLOR="color" | "*style-attribute*"
specifies a fill color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default Color attribute of the GraphAnnoShape style element in the current style (GraphAnnoShape:Color)

Interaction This option takes effect only if the DISPLAY option is specified as FILLED or ALL.

FILLSTYLEELEMENT="*style-element*"
specifies a style element for the fill. Here is an example of a style element:

GraphData2

Note Only the COLOR attribute of the style element applies to the fill.

See "Style Elements for Use with ODS Graphics" on page 1871

FILLTRANSPARENCY=*numeric-value*
specifies the transparency for the fill portion of an annotation.

Default 0.0

Range 0.0 (completely opaque) to 1.0 (completely transparent)

Tip To set both the fill and outline transparency, if specified, use the TRANSPARENCY= option.

HEIGHTUNIT="DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the HEIGHT value.

Default PERCENT

ID="*annotation-identifier*"
Defines an ID for the annotation. The ID contains a unique character value that identifies the subset to which each annotation belongs. All annotations in the annotation data set with an ID variable value that matches the specified annotation identifier are drawn. If the annotation data set does not contain an ID variable or if no ID variable value matches the specified identifier, no annotations are drawn.

Default All of the annotations in the SGANNO data set are drawn.

LAYER="BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR="color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN="*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT="*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

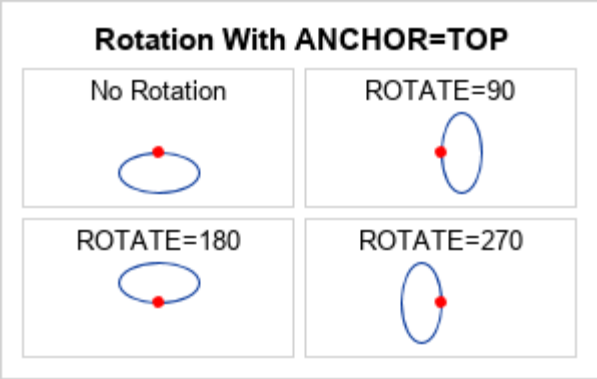
See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS=*n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

RESET="ALL"
Re-initializes all variables.

ROTATE= *degrees*
rotates the annotation, measured in degrees. The angle is measured as if a horizontal line extended to the right through the oval anchor point as shown in the following figure.



Positive angles rotate the annotation counterclockwise, and negative angles rotate the annotation clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0 (no rotation)

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the annotation. This option affects both the outline and the fill, if specified.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Tip FILLTRANSPARENCY= overrides the fill transparency specified by this option.

URL=*"text-string"*
specifies a URL for a web page to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

WIDTHUNIT="DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the WIDTH value.

Default PERCENT

X1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS="X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS="Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

%SGPOLYCONT Annotate Macro

Continues drawing a polygon that was begun with the %SGPOLYGON macro, or a line that was begun with the %SGPOLYLINE macro. %SGPOLYCONT specifies each successive point in the polygon or polyline.

Requirements: You must run the %SGANNO macro before using any other SG annotation macros. For more information, see "Making the Macros Available" on page 2008.

If you are drawing a polyline, one or more %SGPOLYCONT macros must immediately follow the %SGPOLYLINE macro to specify the segments of the polyline. The %SGPOLYLINE and subsequent %SGPOLYCONT macros must be contiguous.

If you are drawing a polygon, two or more %SGPOLYCONT macros must immediately follow the %SGPOLYGON macro to specify the segments of the polygon. The %SGPOLYGON and subsequent %SGPOLYCONT macros must be contiguous. A polygon is similar to a polyline except that the segments of the polygon must form a closed shape.

Syntax

%SGPOLYCONT (X1=*numeric-value* | XC1="*text-string*", Y1=*numeric-value* | YC1="*text-string*", <*options*>)

Required Arguments

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Optional Arguments

ID="*annotation-identifier*"
Defines an ID for the annotation. The ID contains a unique character value that identifies the subset to which each annotation belongs. All annotations in the annotation data set with an ID variable value that matches the specified annotation identifier are drawn. If the annotation data set does not contain an

ID variable or if no ID variable value matches the specified identifier, no annotations are drawn.

Default All of the annotations in the SGANNO data set are drawn.

RESET="ALL"
Re-initializes all variables.

X1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

Y1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

%SGPOLYGON Annotate Macro

Specifies the beginning point of a polygon. A polygon is a connected series of lines that form a closed shape. This function is used in conjunction with two or more %SGPOLYCONT macros to define vertex points for the polygon. Associated variables can define the fill pattern and color, as well as the line type that outlines the polygon.

Requirements: You must run the %SGANNO macro before using any other SG annotation macros. For more information, see "Making the Macros Available" on page 2008.

The SG annotation data set must also specify at least two instances of the %SGPOLYCONT macro immediately after the %SGPOLYGON macro.

Syntax

%SGPOLYGON (X1=*numeric-value* | XC1="*text-string*", Y1=*numeric-value* | YC1="*text-string*", <*options*>)

Required Arguments

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= *"text-string"*
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

Y1= *numeric-value* | YC1= *"text-string"*
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= *"text-string"*
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Optional Arguments

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset the annotation from a discrete value in data space.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset)

DISPLAY="ALL" | "FILL" | "OUTLINE"
specifies the fill and outline properties. You can specify that the annotation be filled (FILL), have an outline (OUTLINE), or both (ALL).

Default OUTLINE

DRAWSPACE="*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

FILLCOLOR="color" | "*style-attribute*"
specifies a fill color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default Color attribute of the GraphAnnoShape style element in the current style (GraphAnnoShape:Color)

Interaction This option takes effect only if the DISPLAY option is specified as FILLED or ALL.

FILLSTYLEELEMENT="*style-element*"
specifies a style element for the fill. Here is an example of a style element:

GraphData2

Note Only the COLOR attribute of the style element applies to the fill.

See "Style Elements for Use with ODS Graphics" on page 1871

FILLTRANSPARENCY=*numeric-value*
specifies the transparency for the fill portion of an annotation.

Default 0.0

Range 0.0 (completely opaque) to 1.0 (completely transparent)

Tip To set both the fill and outline transparency, if specified, use the TRANSPARENCY= option.

ID="*annotation-identifier*"
Defines an ID for the annotation. The ID contains a unique character value that identifies the subset to which each annotation belongs. All annotations in the annotation data set with an ID variable value that matches the specified annotation identifier are drawn. If the annotation data set does not contain an ID variable or if no ID variable value matches the specified identifier, no annotations are drawn.

Default All of the annotations in the SGANNO data set are drawn.

LAYER="BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR="color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN="*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT="*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS=*n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

RESET="ALL"
Re-initializes all variables.

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the annotation. This option affects both the outline and the fill, if specified.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Tip FILLTRANSPARENCY= overrides the fill transparency specified by this option.

URL="*text-string*"
specifies a URL for a web page to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

X1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT

- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS="X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS="Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

%SGPOLYLINE Annotate Macro

Specifies the beginning point of a polyline, which is a connected series of line segments. This function is used in conjunction with one or more %SGPOLYCONT functions to draw a series of connected straight lines.

Requirements: You must run the %SGANNO macro before using any other SG annotation macros. For more information, see "Making the Macros Available" on page 2008.

The SG annotation data set must also specify at least one instance of the %SGPOLYCONT macro immediately after the %SGPOLYLINE macro.

Syntax

%SGPOLYLINE (X1=*numeric-value* | XC1="*text-string*", Y1=*numeric-value* | YC1="*text-string*", <*options*>)

Required Arguments

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Optional Arguments

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset the annotation from a discrete value in data space.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset)

DRAWSPACE="*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

ID="*annotation-identifier*"
Defines an ID for the annotation. The ID contains a unique character value that identifies the subset to which each annotation belongs. All annotations in the annotation data set with an ID variable value that matches the specified annotation identifier are drawn. If the annotation data set does not contain an ID variable or if no ID variable value matches the specified identifier, no annotations are drawn.

Default All of the annotations in the SGANNO data set are drawn.

LAYER="BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR="color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN="*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT="*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS=*n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

RESET="ALL"
Re-initializes all variables.

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the annotation.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*"text-string"*
specifies a URL for a web page to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

X1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS="X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS="Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

%SGRECTANGLE Annotate Macro

Draws a rectangle annotation around the specified (X1, Y1) coordinates using the specified height and width. By default, the (X1, Y1) coordinates constitute the center of the rectangle, though you can change this behavior using the ANCHOR option.

Requirement: You must run the %SGANNO macro before using any other SG annotation macros. For more information, see "Making the Macros Available" on page 2008.

Tip: An easy way to draw a square is to specify PIXEL for the HEIGHTUNIT and WIDTHUNIT variables and then provide the same value for the HEIGHT and WIDTH.

Syntax

%SGRECTANGLE (HEIGHT=*numeric-value*, WIDTH=*numeric-value*,
X1=*numeric-value* | XC1="*text-string*",
Y1=*numeric-value* | YC1="*text-string*", <*options*>)

Required Arguments

HEIGHT=*numeric-value*
specifies the height of the annotation. Specify a positive number greater than zero. You can use the HEIGHTUNIT variable to specify the unit of measurement.

Default none

WIDTH=*numeric-value*
specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default none

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Optional Arguments

ANCHOR="TOPLEFT" | "TOP" | "TOPRIGHT" | "RIGHT" | "BOTTOMRIGHT" | "BOTTOM" | "BOTTOMLEFT" | "LEFT" | "CENTER"
specifies the anchor position of the annotation. This point is placed on the specified X1 and Y1 positions.

Default CENTER

CORNERRADIUS= *numeric-value*
specifies the roundness of the corners of rectangles when used with SG procedures other than SGSCATTER.

Specify a value from 0.0 (completely rectangular) to 1.0 (oval).

Figure 22.1 *CORNERRADIUS= Examples*

Default 0.0

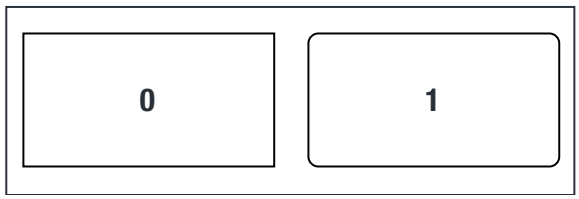
Restriction This syntax does not apply to the SGSCATTER procedure. For the SGSCATTER procedure, use 0 | 1 instead.

CORNERRADIUS= 0 | 1

specifies whether corners are square (0) or rounded (1) when used with the SGSCATTER procedure.

For rounded corners, the radius is fixed.

Figure 22.2 *CORNERRADIUS= Examples*



Default 0

Restriction This syntax applies to the SGSCATTER procedure only. For the SGPLOT, SGPANEL, and SGRENDER procedures, use *numeric-value* instead.

Note Any value that is greater than 0 and less than or equal to 1 is interpreted as 1.

DISCRETEOFFSET=*numeric-value*

specifies an amount to offset the annotation from a discrete value in data space.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset)

DISPLAY="ALL" | "FILL" | "OUTLINE"

specifies the fill and outline properties. You can specify that the annotation be filled (FILL), have an outline (OUTLINE), or both (ALL).

Default OUTLINE

DRAWSPACE="*draw-space-value*"

specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL

- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

FILLCOLOR="color" | "*style-attribute*"
specifies a fill color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default Color attribute of the GraphAnnoShape style element in the current style (GraphAnnoShape:Color)

Interaction This option takes effect only if the DISPLAY option is specified as FILLED or ALL.

FILLSTYLEELEMENT="*style-element*"
specifies a style element for the fill. Here is an example of a style element:

GraphData2

Note Only the COLOR attribute of the style element applies to the fill.

See "Style Elements for Use with ODS Graphics" on page 1871

FILLTRANSPARENCY=*numeric-value*
specifies the transparency for the fill portion of an annotation.

Default 0.0

Range 0.0 (completely opaque) to 1.0 (completely transparent)

Tip To set both the fill and outline transparency, if specified, use the TRANSPARENCY= option.

HEIGHTUNIT="DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the HEIGHT value.

Default PERCENT

ID="*annotation-identifier*"
Defines an ID for the annotation. The ID contains a unique character value that identifies the subset to which each annotation belongs. All annotations in the annotation data set with an ID variable value that matches the specified annotation identifier are drawn. If the annotation data set does not contain an ID variable or if no ID variable value matches the specified identifier, no annotations are drawn.

Default All of the annotations in the SGANNO data set are drawn.

LAYER="BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR="color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN="*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT="*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

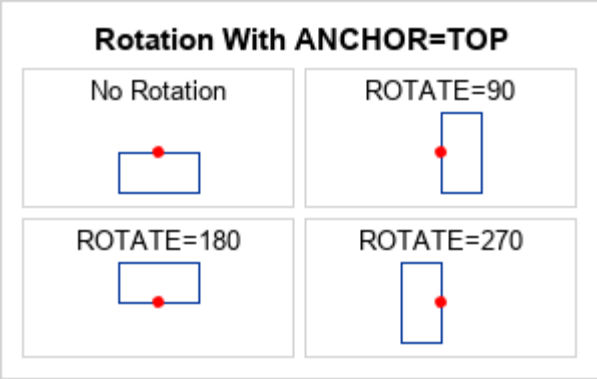
See "Style Elements for Use with ODS Graphics" on page 1871

LINETHICKNESS=*n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

RESET="ALL"
Re-initializes all variables.

ROTATE= *degrees*
rotates the annotation, measured in degrees. The angle is measured as if a horizontal line extended to the right through the rectangle anchor point as shown in the following figure.



Positive angles rotate the annotation counterclockwise, and negative angles rotate the annotation clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0 (no rotation)

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the annotation. This option affects both the outline and the fill, if specified.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

Tip FILLTRANSPARENCY= overrides the fill transparency specified by this option.

URL=*"text-string"*
specifies a URL for a web page to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

WIDTHUNIT="DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the WIDTH value.

Default PERCENT

X1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS="X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS="Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

%SGTEXT Annotate Macro

Places text in the graph output. Associated variables can control the color, size, font, base angle, and rotation of the characters displayed. If you do not supply the (X1, Y1) coordinates, the text is placed in the center of the graph.

Requirement: You must run the %SGANNO macro before using any other SG annotation macros. For more information, see "Making the Macros Available" on page 2008.

Syntax

%SGTEXT (LABEL=*"text-string"*, <*options*>)

Required Argument

LABEL="*text-string*"
specifies the text label. The *text-string* can contain superscripts, subscripts, and Unicode characters by using ODS escapement notation. For example: `(*ESC*){sup '2'}`

You can invoke a macro that returns a string. For example: `label="%mylabel"`, where *mylabel* is the name of the macro.

Note: Some combinations of characters can result in warning messages written to the log. The actual warning varies with the types of characters. For example, `label="95&Conf Interval"` and `label="95%Conf Interval"` both produce a warning, although the annotation is drawn in both cases. There are several ways to avoid the warning message:

- add a space after the % or & character.
- escape the % character with a second % character. For example: `label="95% %Conf Interval"`
- use the %NRQUOTE macro function. For example: `label="95%nrquote(&)Conf Interval"`

Default none

Optional Arguments

ANCHOR="TOPLEFT" | "TOP" | "TOPRIGHT" | "RIGHT" | "BOTTOMRIGHT" | "BOTTOM" | "BOTTOMLEFT" | "LEFT" | "CENTER"
specifies the anchor position of the annotation. This point is placed on the specified X1 and Y1 positions.

Default CENTER

BORDER="TRUE" | "FALSE"
turns the border on and off.

Default FALSE

DISCRETEOFFSET=*numeric-value*
specifies an amount to offset the annotation from a discrete value in data space.

Default 0.0 (no offset)

Range -0.5 (left offset) to +0.5 (right offset)

DRAWSPACE="*draw-space-value*"
specifies the drawing space and units for the annotation. DRAWSPACE can be used rather than specify individual values for X1SPACE, X2SPACE, Y1SPACE, and Y2SPACE.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT (unless overridden by a coordinate draw space, such as X1SPACE)

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

See For more information about these values, see "Drawing Space Values" on page 1944.

FILLCOLOR="color" | "*style-attribute*"
specifies a fill color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default The Color attribute of the GraphAnnoShape style element in the current style (GraphAnnoShape:Color)

FILLSTYLEELEMENT="*style-element*"
specifies a style element for the fill. Here is an example of a style element:

GraphData2

Note Only the COLOR attribute of the style element applies to the fill.

See "Style Elements for Use with ODS Graphics" on page 1871

FILLTRANSPARENCY=*numeric-value*
specifies the transparency for the fill portion of an annotation.

Default 0.0

Range 0.0 (completely opaque) to 1.0 (completely transparent)

Tip To set both the fill and border transparency, if specified, use the TRANSPARENCY= option.

ID="*annotation-identifier*"
Defines an ID for the annotation. The ID contains a unique character value that identifies the subset to which each annotation belongs. All annotations in the annotation data set with an ID variable value that matches the specified annotation identifier are drawn. If the annotation data set does not contain an ID variable or if no ID variable value matches the specified identifier, no annotations are drawn.

Default All of the annotations in the SGANNO data set are drawn.

JUSTIFY= "CENTER" | "LEFT" | "RIGHT"
specifies the text justification.

Default LEFT

LAYER="BACK" | "FRONT"
specifies whether the annotation is drawn in front of (FRONT) or behind (BACK) the graph.

Default FRONT

Interaction For the annotation to appear behind the graph, you might need to disable the wall display. You can use the NOWALL option in the PROC SGPLOT statement (SGPLOT procedure) or in the PANELBY statement (SGPANEL procedure). The NOWALL option is also available in the PLOT and COMPARE statements of the SGSCATTER procedure.

LINECOLOR="color" | "*style-attribute*"
specifies a line color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute:

GraphData2:ContrastColor

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default ContrastColor attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:ContrastColor).

LINEPATTERN="*line-pattern*"
specifies a line pattern for the annotation. You can reference SAS patterns by number or by name. See the list of line patterns on page 1829.

Default LineStyle attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine:lineStyle)

LINESTYLEELEMENT="*style-element*"
specifies a style element for the line. Here is an example of a style element:

GraphData2

See "Style Elements for Use with ODS Graphics" on page 1871

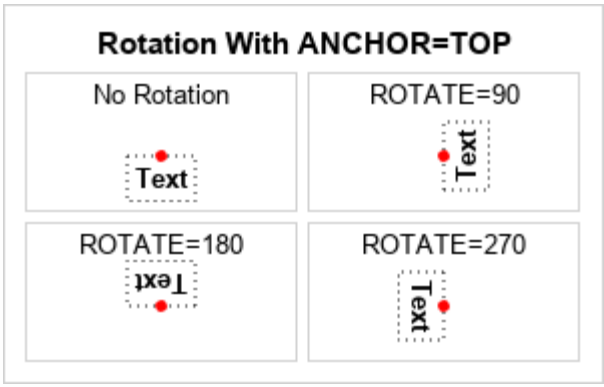
LINETHICKNESS=*n*
specifies the thickness of the line. You cannot specify the unit of measure. The default unit of pixels is always used.

Default LineThickness attribute of the GraphAnnoLine style element in the current style (GraphAnnoLine: LineThickness)

Interaction This option takes effect only if BORDER= "TRUE".

RESET="ALL"
Re-initializes all variables.

ROTATE= *degrees*
rotates the annotation, measured in degrees. The angle is measured as if a horizontal line extended to the right through the text box anchor point as shown in the following figure.



Positive angles rotate the annotation counterclockwise, and negative angles rotate the annotation clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0 (no rotation)

TEXTCOLOR="color" | "*style-attribute*"
specifies the text color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default COLOR attribute of the GraphAnnoText style element in the current style (GraphAnnoText:Color)

TEXTFONT="*font-family*"
specifies the font family for the annotation. The SAS ODS styles use TrueType system fonts. For more information, see "Fonts" on page 1819.

Default FontFamily attribute of the GraphAnnoText style element in the current style.

TEXTSIZE=*n*
specifies the font size of the annotation.

Default FontSize attribute of the GraphAnnoText style element in the current style.

TEXTSTYLE="ITALIC" | "NORMAL"
specifies whether the annotation characters are italic (ITALIC) or normal (NORMAL).

Default FontStyle attribute of the GraphAnnoText style element in the current style.

TEXTWEIGHT="BOLD" | "NORMAL"
specifies whether the annotation characters are bold (BOLD) or normal (NORMAL).

Default FontWeight attribute of the GraphAnnoText style element in the current style.

TRANSPARENCY=*numeric-value*
specifies the degree of transparency for the annotation.

Default 0.0

Range 0 (completely opaque) to 1 (completely transparent)

URL=*"text-string"*
specifies a URL for a web page to be displayed when parts of the plot are selected within an HTML page.

Default By default, no HTML links are created.

Interactions This option affects HTML output only.

This option has no effect unless you also specify IMAGEMAP in the ODS GRAPHICS statement. For more information, see "Using the ODS GRAPHICS Statement" on page 1886.

WIDTH=*numeric-value*
specifies the width of the annotation. Specify a positive number greater than zero. You can use the WIDTHUNIT variable to specify the unit of measurement.

Default Determined by the system

WIDTHUNIT="DATA" | "PERCENT" | "PIXEL"
specifies the dimension unit to use for the WIDTH value.

Default PERCENT

X1= *numeric-value* | XC1= "*text-string*"
specifies the X coordinate of the annotation.

Use one of the following arguments:

X1= *numeric-value*
specifies the X coordinate for numeric data.

XC1= "*text-string*"
specifies the X coordinate for character data.

Note If the X-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both X1= and XC1= are specified, then X1= takes precedence, and XC1= is ignored.

X1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's first X coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the X2SPACE= value, then the second X coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

XAXIS="X" | "X2"
specifies which X axis to use for data space annotations.

Default X

Y1= *numeric-value* | YC1= "*text-string*"
specifies the Y coordinate of the annotation.

Use one of the following arguments:

Y1= *numeric-value*
specifies the Y coordinate for numeric data.

YC1= "*text-string*"
specifies the Y coordinate for character data.

Note If the Y-axis character variable is formatted, then *text-string* must match a formatted value. If it does not match any of the formatted values, the annotation is dropped.

Default none

Note If both Y1= and YC1= are specified, then Y1= takes precedence, and YC1= is ignored.

Y1SPACE="*draw-space-value*"
specifies the drawing space of the annotation's second Y coordinate.

draw-space-value can be one of the following:

- DATAPERCENT
- DATAPIXEL
- DATAVALUE

  Note: When a DATAVALUE annotation is associated with a discrete axis, the annotation value must be a formatted value on the axis. Use the XC1 or YC1 columns for those values.

- GRAPHPERCENT
- GRAPHPIXEL
- LAYOUTPERCENT
- LAYOUTPIXEL
- WALLPERCENT
- WALLPIXEL

Default LAYOUTPERCENT

Restriction For the SGPANEL and SGSCATTER procedures, only GRAPHPERCENT, GRAPHPIXEL, LAYOUTPERCENT, and LAYOUTPIXEL values are valid.

Note If you do not also specify the Y2SPACE= value, then the second Y coordinate uses the default drawing space.

See For more information about these values, see "Drawing Space Values" on page 1944.

YAXIS="Y" | "Y2"
specifies which Y axis to use for data space annotations.

Default Y

%SGTEXTCONT Annotate Macro

Continues a text string that was begun with the TEXT macro. You can set the continued text apart from the main text by specifying different text attributes.

Requirements: You must run the %SGANNO macro before using any other SG annotation macros. For more information, see "Making the Macros Available" on page 2008.

To continue the text, immediately follow the initial %SGTEXT macro with one or more %SGTEXTCONT macros. The %SGTEXT and subsequent %SGTEXTCONT macros must be contiguous.

Syntax

%SGTEXTCONT (LABEL=*"text-string"*, <*options*>)

Required Argument

LABEL="*text-string*"
specifies the text label. The *text-string* can contain superscripts, subscripts, and Unicode characters by using ODS escapement notation. For example: `(*ESC*){sup '2'}`

You can invoke a macro that returns a string. For example: `label="%mylabel"`, where *mylabel* is the name of the macro.

Note: Some combinations of characters can result in warning messages written to the log. The actual warning varies with the types of characters. For example, `label="95&Conf Interval"` and `label="95%Conf Interval"` both produce a warning, although the annotation is drawn in both cases. There are several ways to avoid the warning message:

- add a space after the % or & character.
- escape the % character with a second % character. For example: `label="95% %Conf Interval"`
- use the %NRQUOTE macro function. For example: `label="95%nrquote(&)Conf Interval"`

Default none

Optional Arguments

ID="*annotation-identifier*"
Defines an ID for the annotation. The ID contains a unique character value that identifies the subset to which each annotation belongs. All annotations in the

annotation data set with an ID variable value that matches the specified annotation identifier are drawn. If the annotation data set does not contain an ID variable or if no ID variable value matches the specified identifier, no annotations are drawn.

Default All of the annotations in the SGANNO data set are drawn.

RESET="ALL"
Re-initializes all variables.

TEXTCOLOR="color" | "*style-attribute*"
specifies the text color for the annotation. You can specify a color or a style element attribute. Here is an example of a style attribute: GraphData2:Color

You can specify colors using a number of different color-naming schemes. For more information, see "Color-Naming Schemes" on page 1808.

Default COLOR attribute of the GraphAnnoText style element in the current style (GraphAnnoText:Color)

TEXTFONT="*font-family*"
specifies the font family for the annotation. The SAS ODS styles use TrueType system fonts. For more information, see "Fonts" on page 1819.

Default FontFamily attribute of the GraphAnnoText style element in the current style.

TEXTSIZE=*n*
specifies the font size of the annotation.

Default FontSize attribute of the GraphAnnoText style element in the current style.

TEXTSTYLE="ITALIC" | "NORMAL"
specifies whether the annotation characters are italic (ITALIC) or normal (NORMAL).

Default FontStyle attribute of the GraphAnnoText style element in the current style.

TEXTWEIGHT="BOLD" | "NORMAL"
specifies whether the annotation characters are bold (BOLD) or normal (NORMAL).

Default FontWeight attribute of the GraphAnnoText style element in the current style.

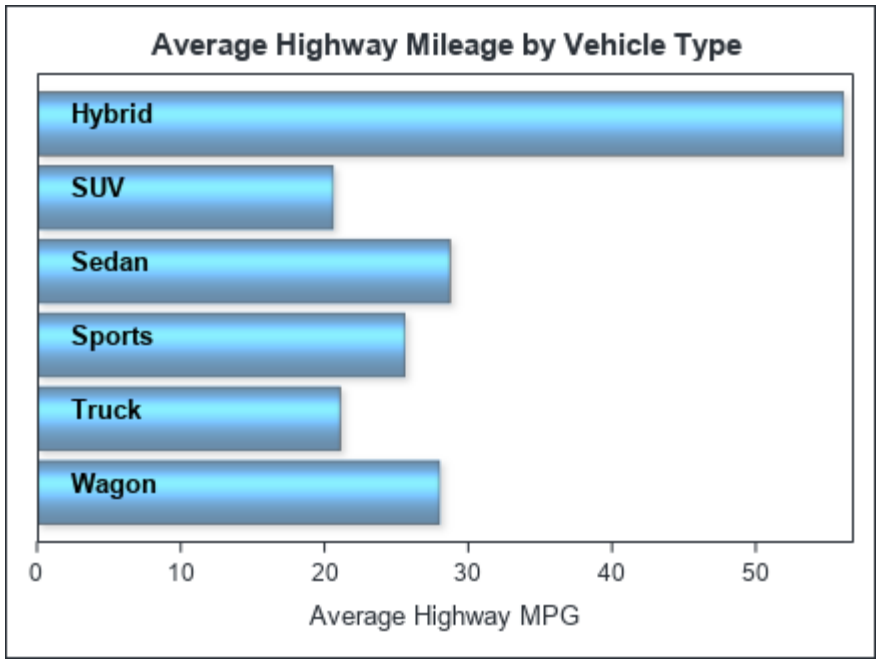
Examples

Example 1: Create Bar Labels

Overview

This example demonstrates how to create custom category labels for a horizontal bar chart. The bar chart plots the average highway mileage (response) by vehicle type (category). Ordinarily, the category labels for a horizontal bar chart appear on the Y axis to the left of each bar. This example demonstrates how to print the labels on the left end of each bar instead. This example also uses the sheen data skin on the bars. Because of the reflection on the sheen data skin, the labels are raised slightly to center the label in the reflection on each bar. Finally, the label text color uses the contrast color.

The following figure shows the final graph.



Program

```
/* Summarize the highway mileage data in SASHELP.CARS. */
proc summary data=sashelp.cars nway;
   class type;
   var mpg_highway;
   output out=mileage mean(mpg_highway) = mpg_highway;
run;

%sganno; /* Compile the annotation macros */
/* Create the annotation data set. */
data anno;
   set mileage;
   %sgText(label=type,
      x1=2,yc1=type,drawspace="datavalue",
      textfont="Arial",textweight="bold",
      textcolor="black",width=100,
      widthunit="pixel",anchor="left",discreteoffset=0.1);
run;


/* Create the plot. */
title "Average Highway Mileage by Vehicle Type";
proc sgplot data=mileage sganno=anno;
  hbarparm category=type response=mpg_highway /
              dataskin=sheen fillattrs=(transparency=0.35);
  xaxis label="Average Highway MPG";
  yaxis display=none;
run;
title;
```

Program Description

Summarize the highway mileage data in Sashelp.Cars. Because a label is needed for each unique value of vehicle type, the data in Sashelp.Cars is first summarized for the Mpg_Highway column using the Type column as the class variable. This step generates a data set that contains one observation for each unique value of Type.

```
/* Summarize the highway mileage data in SASHELP.CARS. */
proc summary data=sashelp.cars nway;
   class type;
   var mpg_highway;
   output out=mileage mean(mpg_highway) = mpg_highway;
run;
```

Create the annotation data set. The Mileage data set is used to create the annotation data set Anno. The DATA step in the Anno data set reads the observations from the Mileage data set. The Type column is used to set the Label column. The remaining columns from the Mileage data set are then dropped. The X1 column is added and set to 2 in order to position the labels on the left end of each bar. The DiscreteOffset column is added and set to 0.1 in order to center the labels in the sheen data skin reflection on each bar. Additional columns are added to specify other attributes of the labels.

```
%sganno; /* Compile the annotation macros */
```

```
/* Create the annotation data set. */
data anno;
   set mileage;
   %sgText(label=type,
      x1=2,yc1=type,drawspace="datavalue",
      textfont="Arial",textweight="bold",
      textcolor="black",width=100,
      widthunit="pixel",anchor="left",discreteoffset=0.1);
run;
```

Create the plot. The HBARPARM statement is used to generate the horizontal bar chart from the summarized mileage data.

```
/* Create the plot. */
title "Average Highway Mileage by Vehicle Type";
proc sgplot data=mileage sganno=anno;
  hbarparm category=type response=mpg_highway /
              dataskin=sheen fillattrs=(transparency=0.35);
  xaxis label="Average Highway MPG";
  yaxis display=none;
run;
title;
```

Listing of the Mileage Data Set

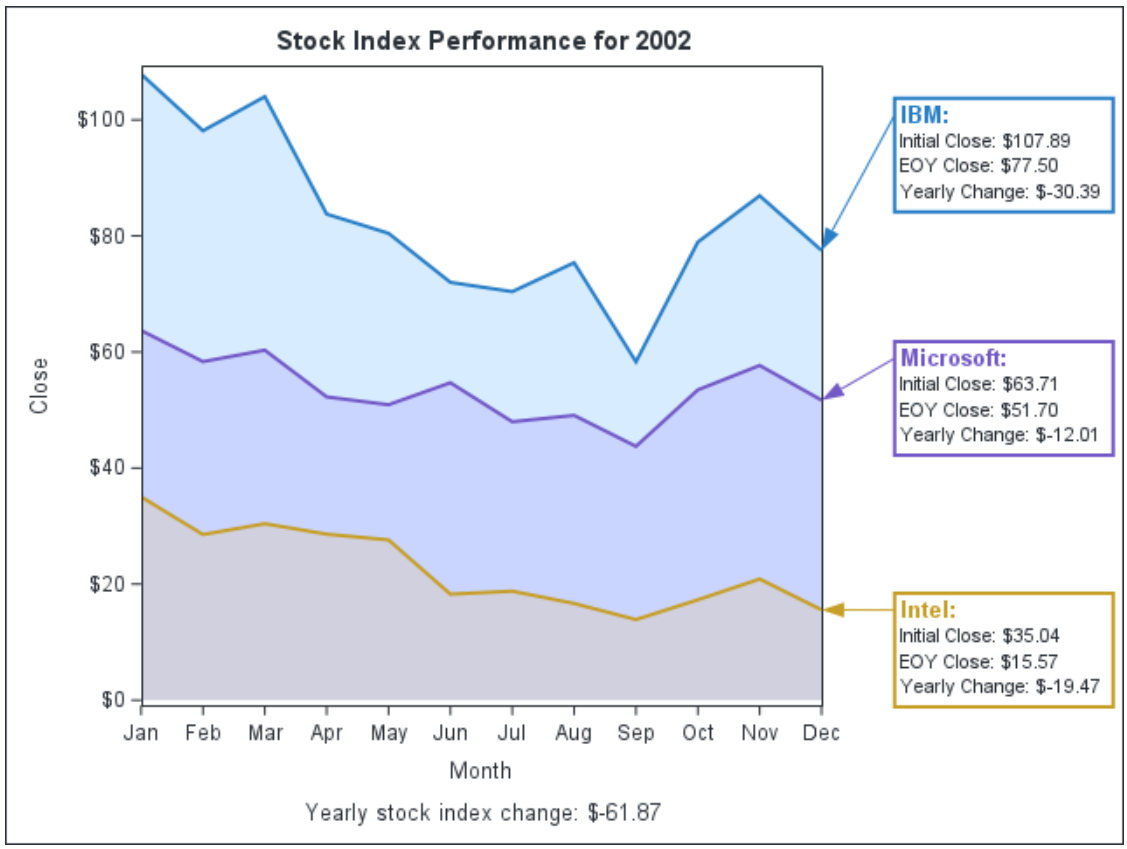
Here is a listing of the Mileage data set.

Obs	Type	_TYPE_	_FREQ_	mpg_city	mpg_ highway
1	Hybrid	1	3	55.0000	56.0000
2	SUV	1	60	16.1000	20.5000
3	Sedan	1	262	21.0840	28.6298
4	Sports	1	49	18.4082	25.4898
5	Truck	1	24	16.5000	21.0000
6	Wagon	1	30	21.1000	27.9000

Example 2: Adding Information to a Graph

You can use the annotation macros to add information to a graph. This example shows you how to use the %SGTEXT, %SGRECTANGLE, and %SGARROW macros to generate annotation data that adds calculated performance information for 2002 to a stock index series plot. The stock index consists of IBM, Intel, and Microsoft stock For each stock, a text box shows the initial closing value of 2002, the end-of-year closing value of 2002, and the index change over the entire year. This example also shows you how to write your own macro for generating annotation data.

Here is the output for this example. As shown in the output, the text boxes are arranged vertically in the order of closing price along the Y axis.



Example Program

```
%let year=2002;                                                    /* 1 */

data stocks_&year;                                                 /* 2 */
   set sashelp.stocks;
   where year(date) eq &year;
   month=month(date);
   monname=put(date,monname3.);
   keep month monname stock open close;
   label monname="Month";
   format date monname3.;
run;

proc sort data=stocks_&year;
   by month stock;
run;

data _null_;                                                       /* 3 */
   set stocks_&year end=last;
   if (month eq 1) then do;
      select (stock);
         when("IBM") call symput("init_IBM_close",strip(close));
```

```
            when("Microsoft") call symput("init_MS_close",strip(close));
            when("Intel") call symput("init_INT_close",strip(close));
            otherwise;
         end;
      end;
      if (month eq 12) then do;
         select (stock);
            when("IBM") call symput("eoy_IBM_close",strip(close));
            when("Microsoft") call symput("eoy_MS_close",strip(close));
            when("Intel") call symput("eoy_INT_close",strip(close));
            otherwise;
         end;
      end;
   run;

   %macro format(value,format=dollar8.2);                          /* 4 */
      %left(%trim(%qsysfunc(putn(&value,&format))))
   %mend format;
                                                                   /* 5 */
   %let yearly_IBM_change=%sysevalf(&eoy_IBM_close - &init_IBM_close);
   %let yearly_MS_change=%sysevalf(&eoy_MS_close - &init_MS_close);
   %let yearly_INT_change=%sysevalf(&eoy_INT_close - &init_INT_close);
   %let yearly_index_change=%sysevalf(&yearly_IBM_change +
      &yearly_MS_change + &yearly_INT_change);

   %macro drawTextBox(x1=, y1=, xc2=, y2=,                         /* 6 */
      width=130, height=65, drawspace=graphpercent,
      widthunit=pixel, heightunit=pixel, anchor=left,
      styleelement="GraphDataDefault", font=Albany AMT,
      linespace=3.05);
      %sgRectangle(x1=&x1, y1=%sysevalf(&y1+2),                    /* 7 */
         height=&height, width=&width, drawspace="&drawspace",
         heightunit="pixel", widthunit="pixel", anchor="topleft",
         linestyleelement="&styleelement");
      %sgText(x1=&x1,y1=&y1,                                       /* 8 */
         width=140,widthunit="&widthunit",anchor="&anchor",
         textsize=10,textweight="bold", textfont="&font",
         textcolor="&styleelement.:contrastColor", label=text1);
      %sgText(x1=&x1,y1=%sysevalf(&y1-&linespace),textweight="normal",
         textsize=8, textcolor="GraphAnnoText:color",label=text2);
      %sgText(x1=&x1,y1=%sysevalf(&y1-&linespace*2),
         label=text3);
      %sgText(x1=&x1,y1=%sysevalf(&y1-&linespace*3),
         label=text4);
      %sgArrow(x1=&x1, y1=&y1, xc2="&xc2", y2=&y2,                 /* 9 */
         x2space="datavalue", y2space="datavalue",
         shape="filled", scale=0.3,
         linestyleelement="&styleelement");
   %mend drawTextBox;

   %sganno;                                                        /* 10 */

   data anno;                                                      /* 11 */
      length text1 $12 text2 text3 text4 $30;
      text1 = "IBM:";                                              /* 12 */
      text2 = "Initial Close: %format(&init_IBM_close)";
```

```
   text3 = "EOY Close: %format(&eoy_IBM_close)";
   text4 = "Yearly Change: %format(&yearly_IBM_change)";
   %drawTextBox(x1=79, y1=87, xc2=Dec, y2=&eoy_IBM_close,    /* 13 */
      styleelement=GraphData1);
   text1 = "Microsoft:";
   text2 = "Initial Close: %format(&init_MS_close)";
   text3 = "EOY Close: %format(&eoy_MS_close)";
   text4 = "Yearly Change: %format(&yearly_MS_change)";
   %drawTextBox(x1=79, y1=58, xc2=Dec, y2=&eoy_MS_close,
      styleelement=GraphData3);
   text1 = "Intel:";
   text2 = "Initial Close: %format(&init_INT_close)";
   text3 = "EOY Close: %format(&eoy_INT_close)";
   text4 = "Yearly Change: %format(&yearly_INT_change)";
   %drawTextBox(x1=79, y1=28, xc2=Dec, y2=&eoy_INT_close,
      styleelement=GraphData2);
   drop text1 text2 text3 text4;
run;

title "Stock Index Performance for &year";                   /* 14 */
footnote "Yearly stock index change:"
   " %format(&yearly_index_change)";

proc sgplot data=stocks_&year sganno=anno pad=(right=160)
   noautolegend;
   band x=monname lower=0 upper=close / modelname="series"
      group=stock fillattrs=(transparency=0.8);
   series x=monname y=close / name="series"
      group=stock lineattrs=(thickness=2px);
   xaxis offsetmin=0 offsetmax=0;
run;
```

1 Macro variable Year specifies the stock data year.

2 This DATA step creates data set stocks_*year* from data set Sashelp.Stocks. Columns Month and MonName are added to the data and are derived from the Date column. Month stores the month number, and MonName stores the 3-character month name. The SORT procedure sorts the data by Month and Stock.

3 This DATA step extracts the initial closing values and end-of-year closing values for each stock into macro variables that are used later.

4 Macro function %Format formats a specified value by using the DOLLAR8.2 format as the default format and returns the result. It is used later to provide formatted values in the DRAWTEXT statements.

5 The %LET statements create macro variables in which the yearly change for each stock and the yearly index change are stored. The %SYSEVALF macro function computes the change for each stock from the initial close and end-of-year close macro variables that were created earlier. It computes the change for the index by summing the stock change values.

6 Macro %DRAWTEXTBOX generates the observations that are needed to draw a text box and arrow by using the %SGTEXT, %SGRECTANGLE, and %SGARROW macros. The %DRAWTEXTBOX macro arguments provide values for the commonly used %SGTEXT, %SGRECTANGLE, and %SGARROW macro arguments.

Because macros can provide default values for the SG annotation macros, they can help reduce the amount of code in your graph templates. They can be used for multiple graphs and can also be shared with others. For more information about SAS macros, see *SAS Macro Language: Reference*.

7 The %SGRECTANGLE macro call generates an annotate observation that draws the outline of the text box.

8 The %SGTEXT macro calls generate annotate observations that draw the text inside the text box. The text is provided by DATA step variables Text1–4. This %SGTEXT macro call draws the text stored in variable Text1 in 10-point bold font. The remaining %SGTEXT macro calls draw the performance information in variables Text2–4 in 8-point normal font.

9 The %SGARROW macro call generates an observation that draws an arrow from the IBM text box to the IBM line in the graph. Because the X2= and Y2= options specify data values, the arrows move with the data. The GraphData1 style element provides the visual attributes for the text box and arrow to match the attributes for IBM line in the graph.

10 Autocall macro %SGANNO compiles the SG annotation macros. You must run this macro before you can use the SG annotation macros.

11 This DATA step creates data set Anno, which stores the annotation instructions for drawing the text boxes.

12 Variables Text1–4 store the IBM stock performance details that are displayed in the text box for IBM.

13 The %DRAWTEXTBOX macro call generates the annotate observations that are needed to draw the text boxes and arrows for IBM, The remaining %DRAWTEXTBOX macro calls generate the annotation instructions for drawing the Microsoft and Intel text boxes.

14 The SGPLOT procedure renders the graph.

PART 8

Appendix

Appendix 1
Units of Measurement 2079

Appendix 2
Reserved Keywords and Unicode Values 2081

Appendix 3
ODS Graphics Software 2085

Appendix 1

Units of Measurement

Units of Measurement 2079

Units of Measurement

Some options enable you to specify the unit of measurement as part of the value.

For example, when using the LINEATTRS= option, you can specify the measurement unit for line thickness. When using the DATALABELATTRS= option, you can specify the font size unit for your data labels.

The following table contains the units that are available:

Table A9.1 *Measurement Units*

Unit	Description
CM	centimeters
IN	inches
MM	millimeters
PCT or %	percentage
PT	point size, calculated at 72 dots per inch
PX	pixels

Appendix 2

Reserved Keywords and Unicode Values

Reserved Keywords and Unicode Values 2081
Overview 2081
Lowercase Greek Letters 2082
Uppercase Greek Letters 2083
Special Characters 2084

Reserved Keywords and Unicode Values

Overview

The tables in this section show some of the reserved keywords and Unicode values that can be used with the UNICODE text command. For information about rendering Unicode characters, see "Managing the String on Text Statements" in *SAS Graph Template Language: User's Guide*.

Note the following:

- Keywords and Unicode values are not case-sensitive: "03B1"x is the same code point as "03b1"x.
- The word `blank` is the keyword for a blank space.

Lowercase Greek Letters

Keyword	Glyph	Unicode	Description
alpha	α	03B1	lowercase alpha
beta	β	03B2	lowercase beta
gamma	γ	03B3	lowercase gamma
delta	δ	03B4	lowercase delta
epsilon	ε	03B5	lowercase epsilon
zeta	ζ	03B6	lowercase zeta
eta	η	03B7	lowercase eta
theta	θ	03B8	lowercase theta
iota	ι	03B9	lowercase iota
kappa	κ	03BA	lowercase kappa
lambda	λ	03BB	lowercase lambda
mu	μ	03BC	lowercase mu
nu	ν	03BD	lowercase nu
xi	ξ	03BE	lowercase xi
omicron	ο	03BF	lowercase omicron
pi	π	03C0	lowercase pi
rho	ρ	03C1	lowercase rho
sigma	σ	03C3	lowercase sigma
tau	τ	03C4	lowercase tau
upsilon	υ	03C5	lowercase upsilon
phi	φ	03C6	lowercase phi

Keyword	Glyph	Unicode	Description
chi	χ	03C7	lowercase chi
psi	ψ	03C8	lowercase psi
omega	ω	03C9	lowercase omega

Uppercase Greek Letters

Table A10.1 *Uppercase Greek Letters*

Keyword	Glyph	Unicode	Description
alpha_u	Α	0391	uppercase alpha
beta_u	Β	0392	uppercase beta
gamma_u	Γ	0393	uppercase gamma
delta_u	Δ	0394	uppercase delta
epsilon_u	Ε	0395	uppercase epsilon
zeta_u	Ζ	0396	uppercase zeta
eta_u	Η	0397	uppercase eta
theta_u	Θ	0398	uppercase theta
iota_u	Ι	0399	uppercase iota
kappa_u	Κ	039A	uppercase kappa
lambda_u	Λ	039B	uppercase lambda
mu_u	Μ	039C	uppercase mu
nu_u	Ν	039D	uppercase nu
xi_u	Ξ	039E	uppercase xi
omicron_u	Ο	039F	uppercase omicron

Keyword	Glyph	Unicode	Description
pi_u	Π	03A0	uppercase pi
rho_u	Ρ	03A1	uppercase rho
sigma_u	Σ	03A3	uppercase sigma
tau_u	Τ	03A4	uppercase theta
upsilon_u	Υ	03A5	uppercase upsilon
phi_u	Φ	03A6	uppercase phi
chi_u	Χ	03A7	uppercase chi
psi_u	Ψ	03A8	uppercase psi
omega_u	Ω	03A9	uppercase omega

Special Characters

Keyword	Glyph	Unicode	Description
prime	′	00B4	single prime sign
bar	¯	0305	combining overline[1]
bar2	‾	033F	combining double overline[1]
tilde	˜	0303	combining tilde[1]
hat	ˆ	0302	combining circumflex accent[1]

1 This is an overstriking character that requires a Unicode font to render properly.

Appendix 3

ODS Graphics Software

Overview of ODS Graphics Software 2085

Overview of ODS Graphics Software

SAS ODS Graphics, sometimes called SAS ODS Statistical Graphics, is an extension of the SAS Output Delivery System (ODS). ODS manages all output created by procedures and enables you to display the output in a variety of forms, such as HTML, PDF, and others. ODS uses a template-based system, providing a variety of templates such as table templates that define the structure of tabular output.

For graphical output, ODS Graphics uses the Graph Template Language (GTL) syntax. The GTL is a comprehensive language for defining statistical graphics. It provides the power and flexibility to create many types of complex graphs. You can customize an existing graph template, or create your own using the GTL. Either way, the result is a template useful for making persistent changes to your graphs.

You can use ODS Graphics software to generate commonly used graphs independent of SAS analytical procedures. Although ODS Graphics software was initially designed to facilitate the production of statistical graphs, its capabilities are also well suited for the production of non-statistical, business graphs. These graphs can be used for preliminary exploration of data and the construction of specialized displays for analyses. These situations require general-purpose graphical tools for the creation of stand-alone plots.

The ODS Graphics system contains the following software:

ODS GRAPHICS statement
: adds graphics capabilities to many SAS analytical and Base procedures. The statement enables or disables ODS Graphics processing and sets graphics environment options. ODS Graphics is enabled by default.

  You can use the ODS GRAPHICS statement options to control many aspects of your graphics. For example, you can use options to specify the size and format

of your output images. The settings that you specify remain in effect for all graphics until you change or reset these settings with another ODS GRAPHICS statement.

SAS ODS Graphics procedures
: provide a concise syntax for creating effective statistical graphs. The *SGPLOT, SGPANEL, SGPIE, and SGSCATTER procedures* provide a traditional SAS procedure interface for the most commonly used features of the SAS Graph Template Language. These procedures enable you to create single-cell plots and charts, multi-page classification panels, comparative scatter plots, and scatter plot matrices. The procedures provide many options for customizing the appearance of your graphs.

  ODS Graphics software includes the capability to render maps. The SGMAP procedure is based on the functionality of ODS Graphics. Use the SGMAP procedure to create maps and then overlay plots such as text, scatter, or bubble plots.

SAS Graph Template Language (GTL)
: provides the power and flexibility to create advanced statistical graphics. Advanced users can create customized layouts and graphs with Graph Template Language.