

SAS/STAT 15.2[®]

User's Guide

The STDIZE Procedure

This document is an individual chapter from *SAS/STAT® 15.2 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2020. *SAS/STAT® 15.2 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 15.2 User's Guide

Copyright © 2020, 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

November 2020

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.

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to <http://support.sas.com/thirdpartylicenses>.

Chapter 114

The STDIZE Procedure

Contents

Overview: STDIZE Procedure	9426
Getting Started: STDIZE Procedure	9426
Syntax: STDIZE Procedure	9433
PROC STDIZE Statement	9433
BY Statement	9438
FREQ Statement	9439
LOCATION Statement	9440
SCALE Statement	9440
VAR Statement	9440
WEIGHT Statement	9440
Details: STDIZE Procedure	9442
Standardization Methods	9442
Computation of the Statistics	9443
Computing Quantiles	9444
Constant Data	9446
Missing Values	9446
Output Data Sets	9446
Displayed Output	9448
ODS Table Names	9448
Example: STDIZE Procedure	9449
Example 114.1: Standardization of Variables in Cluster Analysis	9449
References	9459

Overview: STDIZE Procedure

The STDIZE procedure standardizes one or more numeric variables in a SAS data set by subtracting a location measure and dividing by a scale measure. A variety of location and scale measures are provided, including estimates that are resistant to outliers and clustering. Some of the well-known standardization methods such as mean, median, standard deviation, range, Huber's estimate, Tukey's biweight estimate, and Andrew's wave estimate are available in the STDIZE procedure.

In addition, you can multiply each standardized value by a constant and add a constant. Thus, the final output value is

$$result = add + multiply \times \frac{original - location}{scale}$$

where

result = final output value
add = constant to add (ADD= option)
multiply = constant to multiply by (MULT= option)
original = original input value
location = location measure
scale = scale measure

PROC STDIZE can also find quantiles in one pass of the data, a capability that is especially useful for very large data sets. With such data sets, the UNIVARIATE procedure might have high or excessive memory or time requirements.

Getting Started: STDIZE Procedure

The following example demonstrates how you can use the STDIZE procedure to obtain location and scale measures of your data.

In the following hypothetical data set, a random sample of grade twelve students is selected from a number of coeducational schools. Each school is classified as one of two types: Urban or Rural. There are 40 observations.

The variables are *id* (student identification), *Type* (type of school attended: 'urban'=urban area and 'rural'=rural area), and *total* (total assessment scores in History, Geometry, and Chemistry).

The following DATA step creates the SAS data set *TotalScores*.

```
data TotalScores;
  title 'High School Scores Data';
  input id Type $ total @@;
  datalines;
1 rural 135 2 rural 125 3 rural 223 4 rural 224 5 rural 133
6 rural 253 7 rural 144 8 rural 193 9 rural 152 10 rural 178
11 rural 120 12 rural 180 13 rural 154 14 rural 184 15 rural 187
```

```

16 rural 111 17 rural 190 18 rural 128 19 rural 110 20 rural 217
21 urban 192 22 urban 186 23 urban 64 24 urban 159 25 urban 133
26 urban 163 27 urban 130 28 urban 163 29 urban 189 30 urban 144
31 urban 154 32 urban 198 33 urban 150 34 urban 151 35 urban 152
36 urban 151 37 urban 127 38 urban 167 39 urban 170 40 urban 123
;

```

Suppose you now want to standardize the total scores in different types of schools prior to any further analysis. Before standardizing the total scores, you can use the box plot from PROC BOXPLOT to summarize the total scores for both types of schools.

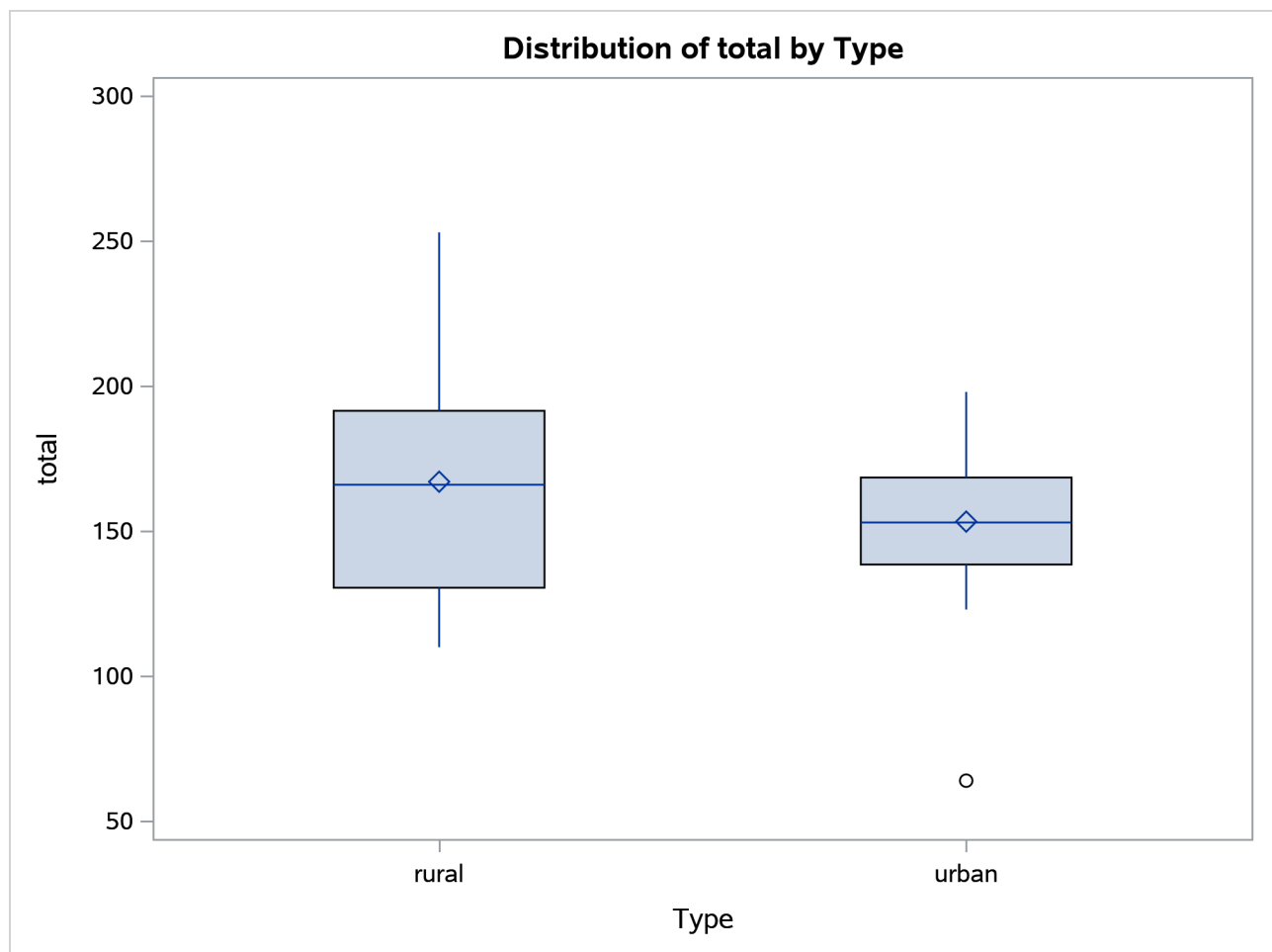
```

ods graphics on;
proc boxplot data=TotalScores;
  plot total*Type / boxstyle=schematic noserifs;
run;

```

The PLOT statement in the PROC BOXPLOT statement creates the schematic plots (without the serifs) when you specify **boxstyle=schematic noserifs**. Figure 114.1 displays a box plot for each type of school.

Figure 114.1 Schematic Plots from PROC BOXPLOT



Inspection reveals that one urban score is a low outlier. Also, if you compare the lengths of two box plots,

there seems to be twice as much dispersion for the rural scores as for the urban scores.

The following PROC UNIVARIATE statement reports the information about the extreme values of the Score variable for each type of school:

```
proc univariate data=TotalScores;
  var total;
  by Type;
run;
```

Figure 114.2 displays the table from PROC UNIVARIATE for the lowest and highest five total scores for urban schools. The outlier (Obs = 23), marked in Figure 114.2 by the symbol ‘0’, has a score of 64.

Figure 114.2 Table for Extreme Observations When Type=urban

High School Scores Data

The UNIVARIATE Procedure Variable: total

Type=urban			
Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
64	23	170	39
123	40	186	22
127	37	189	29
130	27	192	21
133	25	198	32

The following PROC STDIZE procedure requests the METHOD=STD option for computing the location and scale measures:

```
proc stdize data=totalscores method=std pstat;
  title2 'METHOD=STD';
  var total;
  by Type;
run;
```

Figure 114.3 displays the table of location and scale measures from the PROC STDIZE statement. PROC STDIZE uses the sample mean as the location measure and the sample standard deviation as the scale measure for standardizing. The PSTAT option displays a table containing these two measures.

Figure 114.3 Location and Scale Measures Table When METHOD=STD

High School Scores Data METHOD=STD

The STDIZE Procedure

Type=rural

Location and Scale Measures

Location = mean

Scale = standard deviation

Name	Location	Scale	N
total	167.050000	41.956713	20

High School Scores Data METHOD=STD

The STDIZE Procedure

Type=urban

Location and Scale Measures

Location = mean

Scale = standard deviation

Name	Location	Scale	N
total	153.300000	30.066768	20

The ratio of the scale of rural scores to the scale of urban scores is approximately 1.4 (41.96/30.07). This ratio is smaller than the dispersion ratio observed in the previous schematic plots.

The STDIZE procedure provides several location and scale measures that are resistant to outliers. The following statements invoke three different standardization methods and display the tables for the location and scale measures:

```
proc stdize data=totalscores method=mad pstat;
  title2 'METHOD=MAD';
  var total;
  by Type;
run;

proc stdize data=totalscores method=iqr pstat;
  title2 'METHOD=IQR';
  var total;
  by Type;
run;

proc stdize data=totalscores method=abw(4) pstat;
  title2 'METHOD=ABW(4)';
  var total;
  by Type;
run;
```

Figure 114.4 displays the table of location and scale measures when the standardization method is median absolute deviation (MAD). The location measure is the median, and the scale measure is the median absolute

deviation from the median. The ratio of the scale of rural scores to the scale of urban scores is approximately 2.06 (32.0/15.5) and is close to the dispersion ratio observed in the previous schematic plots.

Figure 114.4 Location and Scale Measures Table When METHOD=MAD

High School Scores Data METHOD=MAD

The STDIZE Procedure

Type=rural

Location and Scale Measures

Location = median

Scale = median abs dev from
median

Name	Location	Scale	N
total	166.000000	32.000000	20

High School Scores Data METHOD=MAD

The STDIZE Procedure

Type=urban

Location and Scale Measures

Location = median

Scale = median abs dev from
median

Name	Location	Scale	N
total	153.000000	15.500000	20

Figure 114.5 displays the table of location and scale measures when the standardization method is IQR. The location measure is the median, and the scale measure is the interquartile range. The ratio of the scale of rural scores to the scale of urban scores is approximately 2.03 (61/30) and is, in fact, the dispersion ratio observed in the previous schematic plots.

Figure 114.5 Location and Scale Measures Table When METHOD=IQR

High School Scores Data METHOD=IQR

The STDIZE Procedure

Type=rural

Location and Scale Measures

Location = median

Scale = interquartile range

Name	Location	Scale	N
total	166.000000	61.000000	20

Figure 114.5 *continued***High School Scores Data
METHOD=IQR****The STDIZE Procedure**

Type=urban

Location and Scale Measures

Location = median

Scale = interquartile range

Name	Location	Scale	N
total	153.000000	30.000000	20

Figure 114.6 displays the table of location and scale measures when the standardization method is ABW, for which the location measure is the biweight one-step M-estimate, and the scale measure is the biweight A-estimate. Note that the initial estimate for ABW is MAD. The following steps help to decide the value of the tuning constant:

1. For rural scores, the location estimate for MAD is 166.0, and the scale estimate for MAD is 32.0. The maximum of the rural scores is 253 (not shown), and the minimum is 110 (not shown). Thus, the tuning constant needs to be 3 so that it does not reject any observation that has a score between 110 to 253.
2. For urban scores, the location estimate for MAD is 153.0, and the scale estimate for MAD is 15.5. The maximum of the rural scores is 198, and the minimum (also an outlier) is 64. Thus, the tuning constant needs to be 4 so that it rejects the outlier (64) but includes the maximum (198) as a normal observation.
3. The maximum of the tuning constants, obtained in steps 1 and 2, is 4.

See Goodall (1983, Chapter 11) for details about the tuning constant. The ratio of the scale of rural scores to the scale of urban scores is approximately 2.06 (32.0/15.5). It is also close to the dispersion ratio observed in the previous schematic plots.

Figure 114.6 Location and Scale Measures Table When METHOD=ABW**High School Scores Data
METHOD=ABW(4)****The STDIZE Procedure**

Type=rural

Location and Scale Measures

Location = biweight 1-step

M-estimate Scale = biweight

A-estimate

Name	Location	Scale	N
total	162.889603	56.662855	20

Figure 114.6 *continued***High School Scores Data
METHOD=ABW(4)****The STDIZE Procedure**

Type=urban

Location and Scale Measures

Location = biweight 1-step

M-estimate Scale = biweight

A-estimate

Name	Location	Scale	N
total	156.014608	28.615980	20

The preceding analysis shows that METHOD=MAD, METHOD=IQR, and METHOD=ABW all provide better dispersion ratios than METHOD=STD does.

You can recompute the standard deviation after deleting the outlier from the original data set for comparison. The following statements create a data set NoOutlier that excludes the outlier from the TotalScores data set and invoke PROC STDIZE with METHOD=STD.

```
data NoOutlier;
  set totalscores;
  if (total = 64) then delete;
run;

proc stdize data=NoOutlier method=std pstat;
  title2 'After Removing Outlier, METHOD=STD';
  var total;
  by Type;
run;
```

Figure 114.7 displays the location and scale measures after deleting the outlier. The lack of resistance of the standard deviation to outliers is clearly illustrated: if you delete the outlier, the sample standard deviation of urban scores changes from 30.07 to 22.09. The new ratio of the scale of rural scores to the scale of urban scores is approximately 1.90 (41.96/22.09).

Figure 114.7 Location and Scale Measures Table When METHOD=STD without the Outlier**High School Scores Data
After Removing Outlier, METHOD=STD****The STDIZE Procedure**

Type=rural

Location and Scale Measures

Location = mean

Scale = standard deviation

Name	Location	Scale	N
total	167.050000	41.956713	20

Figure 114.7 *continued*
High School Scores Data
After Removing Outlier, METHOD=STD

The STDIZE Procedure

Type=urban

Location and Scale Measures			
Location = mean			
Scale = standard deviation			
Name	Location	Scale	N
total	158.000000	22.088207	19

Syntax: STDIZE Procedure

The following statements are available in the STDIZE procedure:

PROC STDIZE *< options >* ;
 BY *variables* ;
 FREQ *variable* ;
 LOCATION *variables* ;
 SCALE *variables* ;
 VAR *variables* ;
 WEIGHT *variable* ;

The PROC STDIZE statement is required. The BY, LOCATION, FREQ, VAR, SCALE, and WEIGHT statements are described in alphabetical order following the PROC STDIZE statement.

PROC STDIZE Statement

PROC STDIZE *< options >* ;

The PROC STDIZE statement invokes the STDIZE procedure. You can specify the following options in the PROC STDIZE statement. [Table 114.1](#) summarizes the options available in the PROC STDIZE statement.

Table 114.1 Summary of PROC STDIZE Statement Options

Option	Description
Specify standardization methods	
METHOD=	Specifies the name of the standardization method
INITIAL=	Specifies the method for computing initial estimates for the A estimates
Unstandardize variables	
UNSTD	Unstandardizes variables when you also specify the METHOD=IN option

Table 114.1 *continued*

Option	Description
Process missing values	
NOMISS	Omits observations with any missing values from computation
MISSING=	Specifies the method or a numeric value for replacing missing values
REPLACE	Replaces missing data with zero in the standardized data
REONLY	Replaces missing data with the location measure (does not standardize the data)
Specify data set details	
DATA=	Specifies the input data set
KEELEN	Specifies that output variables inherit the length of the analysis variable
OUT=	Specifies the output data set
OPREFIX=	Specifies that original variables appear in the OUT= data set
SPREFIX=	Specifies a prefix for the standardized variable names
OUTSTAT=	Specifies the output statistic data set
Specify computational settings	
VARDEF=	Specifies the variances divisor
NMARKERS=	Specifies the number of markers when you also specify PCTLMTD=ONEPASS
MULT=	Specifies the constant to multiply each value by after standardizing
ADD=	Specifies the constant to add to each value after standardizing and multiplying by the value specified in the MULT= option
FUZZ=	Specifies the relative fuzz factor for writing the output
Specify percentiles	
PCTLDEF=	Specifies the definition of percentiles when you also specify the PCTLMTD=ORD_STAT option
PCTLMTD=	Specifies the method used to estimate percentiles
PCTLPTS=	Writes observations containing percentiles to the data set specified in the OUTSTAT= option
Normalize scale estimators	
NORM	Normalizes the scale estimator to be consistent for the standard deviation of a normal distribution
SNORM	Normalizes the scale estimator to have an expectation of approximately 1 for a standard normal distribution
Specify output	
PSTAT	Displays the location and scale measures

These options and their abbreviations are described (in alphabetical order) in the remainder of this section.

ADD=*c*

specifies a constant, *c*, to add to each value after standardizing and multiplying by the value you specify in the MULT= option. The default value is 0.

DATA=*SAS-data-set*

specifies the input data set to be standardized. If you omit the DATA= option, the most recently created data set is used.

FUZZ=*c*

specifies the relative fuzz factor. The default value is 1E–14. For the OUT= data set, the score is computed as follows:

$$\text{if } |result| < m \times c \text{ then } result = 0$$

where *m* is the constant specified in the MULT= option, or 1 if MULT= option is not specified.

For the OUTSTAT= data set and the location and scale table, the *scale* and *location* values are computed as follows:

$$\text{if } scale < |location| \times c \text{ then } scale = 0$$

Otherwise,

$$\text{if } |location| < m \times c \text{ then } location = 0$$

INITIAL=*method*

specifies the method for computing initial estimates for the A estimates (ABW, AWAVE, and AHUBER). You cannot specify the following methods for initial estimates: INITIAL=ABW, INITIAL=AHUBER, INITIAL=AWAVE, and INITIAL=IN. The default is INITIAL=MAD.

KEEPLN

specifies that the standardized variables inherit the lengths of the analysis variables that PROC STDIZE uses to derive them. PROC STDIZE stores numbers in double-precision without this option.

Caution: The KEEPLEN option causes the standardized variables to permanently lose numeric precision by truncating or rounding the values. However, the precision of the output variables will match that of the input.

METHOD=*name*

specifies the name of the method for computing location and scale measures. Valid values for *name* are as follows: MEAN, MEDIAN, SUM, EUCLEN, USTD, STD, RANGE, MIDRANGE, MAXABS, IQR, MAD, ABW, AHUBER, AWAVE, AGK, SPACING, L, and IN.

For details about these methods, see the descriptions in the section “[Standardization Methods](#)” on page 9442. The default is METHOD=STD.

MISSING=method | value

specifies the method (or a numeric value) for replacing missing values. If you omit the MISSING= option, the REPLACE option replaces missing values with the location measure given by the METHOD= option. Specify the MISSING= option when you want to replace missing values with a different value. You can specify any name that is valid in the METHOD= option except the name IN. The corresponding location measure is used to replace missing values.

If a numeric value is given, the value replaces missing values after standardizing the data. However, you can specify the REONLY option with the MISSING= option to suppress standardization for cases in which you want only to replace missing values.

MULT=c

specifies a constant, *c*, by which to multiply each value after standardizing. The default value is 1.

NMARKERS=n

specifies the number of markers used when you specify the one-pass algorithm (PCTLMTD=ONEPASS). The value *n* must be greater than or equal to 5. The default value is 105.

NOMISS

omits observations with missing values for any of the analyzed variables from calculation of the location and scale measures. If you omit the NOMISS option, all nonmissing values are used.

NORM

normalizes the scale estimator to be consistent for the standard deviation of a normal distribution when you specify the option METHOD=AGK, METHOD=IQR, METHOD=MAD, or METHOD=SPACING.

OPREFIX=<o-prefix>

specifies that the original variables should appear in the OUT= data set. You can optionally specify an equal sign and a prefix. For example, if OPREFIX=Original, then the names of the variables are OriginalVAR1, OriginalVAR2, and so on, where VAR1 and VAR2 are the original variable names. The value of OPREFIX= must be different from the value of SPREFIX=. If you specify OPREFIX, without an equal sign and a prefix, then the default prefix is null and you must specify SPREFIX=*s-prefix*.

OUT=SAS-data-set

specifies the name of the SAS data set created by PROC STDIZE. By default, the output data set is a copy of the DATA= data set except that the analyzed variables have been standardized. Analyzed variables are those specified in the VAR statement or, if there is no VAR statement, all numeric variables not listed in any other statement. However, you can use the OPREFIX option to request that both the original and standardized variables be included in the output data set. You can change variable names by specifying prefixes with the OPREFIX= and SPREFIX= options. See the section “Output Data Sets” on page 9446 for more information.

If you want to create a SAS data set in a permanent library, you must specify a two-level name. For more information about permanent libraries and SAS data sets, see [SAS Programmers Guide: Essentials](#).

If you omit the OUT= option, PROC STDIZE creates an output data set named according to the DATA n convention.

OUTSTAT=SAS-data-set

specifies the name of the SAS data set containing the location and scale measures and other computed statistics. See the section “[Output Data Sets](#)” on page 9446 for more information.

PCTLDEF=percentiles

specifies which of five definitions is used to calculate percentiles when you specify the option PCTLMTD=ORD_STAT. By default, PCTLDEF=5. Note that the option PCTLMTD=ONEPASS implies PCTLDEF=5. See the section “[Computational Methods for the PCTLDEF= Option](#)” on page 9445 for details about percentile definition.

You cannot use PCTLDEF= when you compute weighted quantiles.

PCTLMTD=ORD_STAT | ONEPASS | P2

specifies the method used to estimate percentiles. Specify the PCTLMTD=ORD_STAT option to compute the percentiles by the order statistics method.

The PCTLMTD=ONEPASS option modifies an algorithm invented by Jain and Chlamtac (1985). See the section “[Computing Quantiles](#)” on page 9444 for more details about this algorithm.

PCTLPTS=*n*

writes percentiles to the OUTSTAT= data set. Values of *n* can be any decimal number between 0 and 100, inclusive.

A requested percentile is identified by the _TYPE_ variable in the OUTSTAT= data set with a value of *Pn*. For example, suppose you specify the option PCTLPTS=10, 30. The corresponding observations in the OUTSTAT= data set that contain the 10th and the 30th percentiles would then have values _TYPE_=P10 and _TYPE_=P30, respectively.

PSTAT

displays the location and scale measures.

REPLACE

replaces missing data with the value 0 in the standardized data (this value corresponds to the location measure before standardizing). To replace missing data by other values, see the preceding description of the MISSING= option. You cannot specify both the REPLACE and REONLY options.

REONLY

replaces missing data only; PROC STDIZE does not standardize the data. Missing values are replaced with the location measure unless you also specify the MISSING=*value* option, in which case missing values are replaced with *value*. You cannot specify both the REPLACE and REONLY options.

SNORM

normalizes the scale estimator to have an expectation of approximately 1 for a standard normal distribution when you specify the METHOD=SPACING option.

SPREFIX=<*s-prefix*>

specifies a prefix for the standardized variables. For example, if SPREFIX=Std, then the names of the standardized variables are StdVAR1, StdVAR2, and so on, where VAR1 and VAR2 are the original variable names. The value of SPREFIX= must be different from the value of [OPREFIX=](#). The default prefix is null. If you omit the SPREFIX option, the standardized variables still appear in the OUT= data set by default and the variable names remain the same. If you want to have the variable names changed, you need to specify a prefix with SPREFIX=*s-prefix*.

UNSTD**UNSTDIZE**

unstandardizes variables when you specify the `METHOD=IN(ds)` option. The location and scale measures, along with constants for addition and multiplication that the unstandardization is based on, are identified by the `_TYPE_` variable in the `ds` data set.

The `ds` data set must have a `_TYPE_` variable and contain the following two observations: a `_TYPE_ = 'LOCATION'` observation and a `_TYPE_ = 'SCALE'` observation. The variable `_TYPE_` can also contain the optional observations, 'ADD' and 'MULT'; if these observations are not found in the `ds` data set, the constants specified in the `ADD=` and `MULT=` options (or their default values) are used for unstandardization.

See the section “[OUTSTAT= Data Set](#)” on page 9447 for details about the statistics that each value of `_TYPE_` represents. The formula used for unstandardization is as follows: If the final output value from the previous standardization is calculated as

$$result = add + multiply \times \frac{original - location}{scale}$$

The unstandardized variable is computed as

$$original = scale \times \frac{result - add}{multiply} + location$$

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor to be used in the calculation of variances. By default, `VARDEF=DF`. The values and associated divisors are as follows.

Value	Divisor	Formula
DF	Degrees of freedom	$n - 1$
N	Number of observations	n
WDF	Sum of weights minus 1	$(\sum_i w_i) - 1$
WEIGHT WGT	Sum of weights	$\sum_i w_i$

BY Statement

BY *variables* ;

You can specify a `BY` statement in `PROC STDIZE` to obtain separate analyses of observations in groups that are defined by the `BY` variables. When a `BY` statement appears, the procedure expects the input data set to be sorted in order of the `BY` variables. If you specify more than one `BY` statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the `SORT` procedure with a similar `BY` statement.
- Specify the `NOTSORTED` or `DESCENDING` option in the `BY` statement in the `STDIZE` procedure. The `NOTSORTED` option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the `BY` variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

When you specify the option METHOD=IN(ds), the following rules are applied to BY-group processing:

- If the ds data set does not contain any of the BY variables, the entire DATA= data set is standardized by the location and scale measures (along with the constants for addition and multiplication) in the ds data set.
- If the ds data set contains some, but not all, of the BY variables or if some BY variables do not have the same type or length in the ds data set that they have in the DATA= data set, PROC STDIZE displays an error message and stops.
- If all of the BY variables appear in the ds data set with the same type and length as in the DATA= data set, each BY group in the DATA= data set is standardized using the location and scale measures (along with the constants for addition and multiplication) from the corresponding BY group in the ds data set. The BY groups in the ds data set must be in the same order in which they appear in the DATA= data set. All BY groups in the DATA= data set must also appear in the ds data set. If you do not specify the NOTSORTED option, some BY groups can appear in the ds data set but not in the DATA= data set; such BY groups are not used in standardizing data.

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

FREQ Statement

FREQ *variable* ;

If one variable in the input data set represents the frequency of occurrence for other values in the observation, specify the variable name in a FREQ statement. PROC STDIZE treats the data set as if each observation appeared n times, where n is the value of the FREQ variable for the observation. Nonintegral values of the FREQ variable are truncated to the largest integer less than the FREQ value. If the FREQ variable has a value that is less than 1 or is missing, the observation is not used in the analysis.

NOTRUNCATE

NOTRUNC

specifies that frequency values are not truncated to integers.

The nonintegral values of the FREQ variable can be used for the following standardization methods: AGK, ABW, AHUBER, AWAVE, EUCLEN, IQR, L, MAD, MEAN, MEDIAN, SPACING, STD, SUM, and USTD. The nonintegral frequency values are used for the MAD, MEDIAN, or IQR method only when PCTLMTD=ORD_STAT is specified. If PCTLMTD=ONEPASS is specified, the NOTRUNCATE option is ignored.

LOCATION Statement

LOCATION *variables* ;

The LOCATION statement specifies a list of numeric variables that contain location measures in the input data set specified by the METHOD=IN option.

SCALE Statement

SCALE *variables* ;

The SCALE statement specifies the list of numeric variables that contain scale measures in the input data set specified by the METHOD=IN option.

VAR Statement

VAR *variable* ;

The VAR statement lists numeric variables to be standardized. If you omit the VAR statement, all numeric variables not listed in the BY, FREQ, and WEIGHT statements are used.

WEIGHT Statement

WEIGHT *variable* ;

The WEIGHT statement specifies a numeric variable in the input data set with values that are used to weight each observation. Only one variable can be specified.

The WEIGHT variable values can be nonintegers. An observation is used in the analysis only if the value of the WEIGHT variable is greater than zero.

The WEIGHT variable applies only when you specify the following standardization methods: AGK, EUCLLEN, IQR, L, MAD, MEAN, MEDIAN, STD, SUM, and USTD. Weights are used for the METHOD=MAD, MEDIAN, or IQR only when PCTLMTD=ORD_STAT is specified; if PCTLMTD=ONEPASS is specified, the WEIGHT statement is ignored.

PROC STDIZE uses the value of the WEIGHT variable to calculate the sample mean and sample variances:

$$\bar{x}_w = \sum_i w_i x_i / \sum_i w_i \quad (\text{sample mean})$$

$$us_w^2 = \sum_i w_i x_i^2 / d \quad (\text{uncorrected sample variances})$$

$$s_w^2 = \sum_i w_i (x_i - \bar{x}_w)^2 / d \quad (\text{sample variances})$$

where w_i is the weight value of the i th observation, x_i is the value of the i th observation, and d is the divisor controlled by the VARDEF= option (see the VARDEF= option for details).

The following weighted statistics are defined accordingly:

MEAN	the weighted mean, $\bar{x}_w$
SUM	the weighted sum, $\sum_i w_i x_i$
USTD	the weighted uncorrected standard deviation, $\sqrt{us_w^2}$
STD	the weighted standard deviation, $\sqrt{s_w^2}$
EUCLEN	the weighted Euclidean length, computed as the square root of the weighted uncorrected sum of squares:

$$\sqrt{\sum_i w_i x_i^2}$$

MEDIAN	the weighted median. See the section “ Weighted Percentiles ” on page 9446 for the formulas and descriptions.
MAD	the weighted median absolute deviation from the weighted median. See the section “ Weighted Percentiles ” on page 9446 for the formulas and descriptions.
IQR	the weighted median, 25th percentile, and the 75th percentile. See the section “ Weighted Percentiles ” on page 9446 for the formulas and descriptions.
AGK	the AGK estimate. This estimate is documented further in the ACECLUS procedure as the METHOD=COUNT option. See the discussion of the WEIGHT statement in Chapter 26, “ The ACECLUS Procedure ,” for information about how the WEIGHT variable is applied to the AGK estimate.
L	the L_p estimate. This estimate is documented further in the FASTCLUS procedure as the LEAST= option. See the discussion of the WEIGHT statement in Chapter 44, “ The FASTCLUS Procedure ,” for information about how the WEIGHT variable is used to compute weighted cluster means. The number of clusters is always 1.

Details: STDIZE Procedure

Standardization Methods

The following table lists standardization methods and their corresponding location and scale measures available with the METHOD= option.

Table 114.2 Available Standardization Methods

Method	Location	Scale
MEAN	Mean	1
MEDIAN	Median	1
SUM	0	Sum
EUCLEN	0	Euclidean length
USTD	0	Standard deviation about origin
STD	Mean	Standard deviation
RANGE	Minimum	Range
MIDRANGE	Midrange	Range/2
MAXABS	0	Maximum absolute value
IQR	Median	Interquartile range
MAD	Median	Median absolute deviation from median
ABW(<i>c</i>)	Biweight one-step M-estimate	Biweight A-estimate
AHUBER(<i>c</i>)	Huber one-step M-estimate	Huber A-estimate
AWAVE(<i>c</i>)	Wave one-step M-estimate	Wave A-estimate
AGK(<i>p</i>)	Mean	AGK estimate (ACECLUS)
SPACING(<i>p</i>)	Mid-minimum spacing	Minimum spacing
L(<i>p</i>)	L(<i>p</i>)	L(<i>p</i>)
IN(ds)	Read from data set	Read from data set

For METHOD=ABW(*c*), METHOD=AHUBER(*c*), or METHOD=AWAVE(*c*), *c* is a positive numeric tuning constant.

For METHOD=AGK(*p*), *p* is a numeric constant that gives the proportion of pairs to be included in the estimation of the within-cluster variances.

For METHOD=SPACING(*p*), *p* is a numeric constant that gives the proportion of data to be contained in the spacing.

For METHOD=L(*p*), *p* is a numeric constant greater than or equal to 1 that specifies the power to which differences are to be raised in computing an L(*p*) or Minkowski metric.

For METHOD=IN(ds), ds is the name of a SAS data set that meets either of the following two conditions:

- The data set contains a _TYPE_ variable. The observation that contains the location measure corresponds to the value _TYPE_ = 'LOCATION', and the observation that contains the scale measure corresponds to the value _TYPE_ = 'SCALE'. You can also use a data set created by the OUTSTAT=

option from another PROC STDIZE statement as the `ds` data set. See the section “[Output Data Sets](#)” on page 9446 for the contents of the `OUTSTAT=` data set.

- The data set contains the location and scale variables specified by the `LOCATION` and `SCALE` statements.

PROC STDIZE reads in the location and scale variables in the `ds` data set by first looking for the `_TYPE_` variable in the `ds` data set. If it finds this variable, PROC STDIZE continues to search for all variables specified in the `VAR` statement. If it does not find the `_TYPE_` variable, PROC STDIZE searches for the location variables specified in the `LOCATION` statement and the scale variables specified in the `SCALE` statement.

The variable `_TYPE_` can also contain the optional observations, ‘ADD’ and ‘MULT’. If these observations are found in the `ds` data set, the values in the observation of `_TYPE_ = ‘MULT’` are the multiplication constants, and the values in the observation of `_TYPE_ = ‘ADD’` are the addition constants; otherwise, the constants specified in the `ADD=` and `MULT=` options (or their default values) are used.

For robust estimators, see Goodall (1983) and Iglewicz (1983). The MAD method has the highest breakdown point (50%), but it is somewhat inefficient. The ABW, AHUBER, and AWAVE methods provide a good compromise between breakdown and efficiency. The $L(p)$ location estimates are increasingly robust as p drops from 2 (which corresponds to least squares, or mean estimation) to 1 (which corresponds to least absolute value, or median estimation). However, the $L(p)$ scale estimates are not robust.

The SPACING method is robust to both outliers and clustering (Janssen et al. 1995) and is, therefore, a good choice for cluster analysis or nonparametric density estimation. The mid-minimum spacing method estimates the mode for small p . The AGK method is also robust to clustering and more efficient than the SPACING method, but it is not as robust to outliers and takes longer to compute. If you expect g clusters, the argument to `METHOD=SPACING` or `METHOD=AGK` should be $\frac{1}{g}$ or less. The AGK method is less biased than the SPACING method for small samples. As a general guide, it is reasonable to use AGK for samples of size 100 or less and SPACING for samples of size 1,000 or more, with the treatment of intermediate sample sizes depending on the available computer resources.

Computation of the Statistics

Formulas for statistics of `METHOD=MEAN`, `METHOD=MEDIAN`, `METHOD=SUM`, `METHOD=USTD`, `METHOD=STD`, `METHOD=RANGE`, and `METHOD=IQR` are given in the chapter “Elementary Statistics Procedures” (*Base SAS Procedures Guide*).

Note that the computations of median and upper and lower quartiles depend on the `PCTLMTD=` option.

The other statistics listed in [Table 114.2](#), except for `METHOD=IN`, are described as follows:

EUCLEN	Euclidean length. $\sqrt{\sum_{i=1}^n x_i^2}$, where x_i is the i th observation and n is the total number of observations in the sample.
$L(p)$	Minkowski metric. This metric is documented as the <code>LEAST=p</code> option in the PROC FASTCLUS statement of the FASTCLUS procedure (see Chapter 44, “ The FASTCLUS Procedure ”).

If you specify `METHOD=L(p)` in the PROC STDIZE statement, your results are similar to those obtained from PROC FASTCLUS if you specify the `LEAST=p` option with `MAXCLUS=1` (and use the default values of the `MAXITER=` option). The difference between the two types of calculations concerns the maximum number of iterations. In PROC STDIZE, it is a criterion for convergence on all variables; in PROC FASTCLUS, it is a criterion for convergence on a single variable.

The location and scale measures for `L(p)` are output to the `OUTSEED=` data set in PROC FASTCLUS.

MIDRANGE	$(\text{maximum} + \text{minimum})/2$
ABW(<i>c</i>)	Tukey's biweight. See Goodall (1983, pp. 376–378, p. 385) for the biweight one-step M-estimate. Also see Iglewicz (1983, pp. 416–418) for the biweight A-estimate.
AHUBER(<i>c</i>)	Hubers. See Goodall (1983, pp. 371–374) for the Huber one-step M-estimate. Also see Iglewicz (1983, pp. 416–418) for the Huber A-estimate of scale.
AWAVE(<i>c</i>)	Andrews' wave. See Goodall (1983, p. 376) for the Wave one-step M-estimate. Also see Iglewicz (1983, pp. 416–418) for the Wave A-estimate of scale.
AGK(<i>p</i>)	The noniterative univariate form of the estimator described by Art, Gnanadesikan, and Kettenring (1982). The AGK estimate is documented in the section on the <code>METHOD=</code> option in the PROC ACECLUS statement of the ACECLUS procedure (also see the section “ Background ” on page 992 in Chapter 26, “ The ACECLUS Procedure ”). Specifying <code>METHOD=AGK(<i>p</i>)</code> in the PROC STDIZE statement is the same as specifying <code>METHOD=COUNT</code> and <code>P=<i>p</i></code> in the PROC ACECLUS statement.
SPACING(<i>p</i>)	The absolute difference between two data values. The minimum spacing for a proportion <i>p</i> is the minimum absolute difference between two data values that contain a proportion <i>p</i> of the data between them. The mid-minimum spacing is the mean of these two data values.

Computing Quantiles

PROC STDIZE offers two methods for computing quantiles: the one-pass approach and the order-statistics approach (like that used in the UNIVARIATE procedure).

The one-pass approach used in PROC STDIZE modifies the P^2 algorithm for histograms proposed by Jain and Chlamtac (1985). The primary difference comes from the movement of markers. The one-pass method allows a marker to move to the right (or left) by more than one position (to the largest possible integer) as long as it does not result in two markers being in the same position. The modification is necessary in order to incorporate the `FREQ` variable.

You might obtain inaccurate results if you use the one-pass approach to estimate quantiles beyond the quartiles (that is, when you estimate quantiles $< P25$ or quantiles $> P75$). A large sample size (10,000 or more) is often required if the tail quantiles (quantiles $\leq P10$ or quantiles $\geq P90$) are requested. Note that, for variables with highly skewed or heavy-tailed distributions, tail quantile estimates might be inaccurate.

The order-statistics approach for estimating quantiles is faster than the one-pass method but requires that the entire data set be stored in memory. The accuracy in estimating the quantiles is comparable for

both methods when the requested percentiles are between the lower and upper quartiles. The default is PCTLMTD=ORD_STAT if enough memory is available; otherwise, PCTLMTD=ONEPASS.

Computational Methods for the PCTLDEF= Option

You can specify one of five methods for computing quantile statistics when you use the order-statistics approach (PCTLMTD=ORD_STAT); otherwise, the PCTLDEF=5 method is used when you use the one-pass approach (PCTLMTD=ONEPASS).

Percentile Definitions Let n be the number of nonmissing values for a variable, and let $x_1, x_2, \dots, x_n$ represent the ordered values of the variable. For the t th percentile, let $p = t/100$. In the following definitions numbered 1, 2, 3, and 5, let

$$np = j + g$$

where j is the integer part and g is the fractional part of np . For definition 4, let

$$(n + 1)p = j + g$$

Given the preceding definitions, the t th percentile, y , is defined as follows:

PCTLDEF=1 weighted average at x_{np}

$$y = (1 - g)x_j + gx_{j+1}$$

where x_0 is taken to be x_1

PCTLDEF=2 observation numbered closest to np

$$y = x_i$$

where i is the integer part of $np + 1/2$ if $g \neq 1/2$. If $g = 1/2$, then

$y = x_j$ if j is even, or

$y = x_{j+1}$ if j is odd

PCTLDEF=3 empirical distribution function

$$y = x_j \text{ if } g = 0$$

$$y = x_{j+1} \text{ if } g > 0$$

PCTLDEF=4 weighted average aimed at $x_{p(n+1)}$

$$y = (1 - g)x_j + gx_{j+1}$$

where x_{n+1} is taken to be x_n

PCTLDEF=5 empirical distribution function with averaging

$$y = (x_j + x_{j+1})/2 \text{ if } g = 0$$

$$y = x_{j+1} \text{ if } g > 0$$

Weighted Percentiles

When you specify a WEIGHT statement, or specify the NOTRUNCATE option in a FREQ statement, the percentiles are computed differently. The 100 p th weighted percentile y is computed from the empirical distribution function with averaging

$$y = \begin{cases} \frac{1}{2}(x_i + x_{i+1}) & \text{if } \sum_{j=1}^i w_j = pW \\ x_{i+1} & \text{if } \sum_{j=1}^i w_j < pW < \sum_{j=1}^{i+1} w_j \end{cases}$$

where w_i is the weight associated with x_i , and where $W = \sum_{i=1}^n w_i$ is the sum of the weights.

For PCTLMTD= ORD_STAT, the PCTLDEF= option is not applicable when a WEIGHT statement is used, or when a NOTRUNCATE option is specified in a FREQ statement. However, in this case, if all the weights are identical, the weighted percentiles are the same as the percentiles that would be computed without a WEIGHT statement and with PCTLDEF=5.

For PCTLMTD= ONEPASS, the quantile computation currently does not use any weights.

Constant Data

Constant variables are not standardized. The scale value is set to missing when the data are constant.

Missing Values

Missing values can be replaced by the location measure or by any specified constant (see the REPLACE option and the MISSING= option). You can also suppress standardization if you want only to replace missing values (see the REONLY option).

If you specify the NOMISS option, PROC STDIZE omits observations with any missing values in the analyzed variables from computation of the location and scale measures.

Output Data Sets

OUT= Data Set

By default, the output data set is a copy of the DATA= data set except that the analyzed variables have been standardized. Analyzed variables are those specified in the VAR statement or, if there is no VAR statement, all numeric variables not listed in any other statement. However, you can use the OPREFIX option to request that both the original and standardized variables be included in the output data set. You can change variable names by specifying prefixes with the OPREFIX=*o-prefix* and SPREFIX=*s-prefix* options, but keep in mind that the two prefixes must be different. See [OPREFIX](#) and [SPREFIX](#) for more information.

OUTSTAT= Data Set

The new data set contains the following variables:

- the BY variables, if any
- `_TYPE_`, a character variable
- the analyzed variables

Each observation in the new data set contains a type of statistic as indicated by the `_TYPE_` variable. The values of the `_TYPE_` variable are as follows:

LOCATION	location measure of each variable
SCALE	scale measure of each variable
ADD	constant specified in the ADD= option. This value is the same for each variable.
MULT	constant specified in the MULT= option. This value is the same for each variable.
N	total number of nonmissing positive frequencies of each variable
NORM	norm measure of each variable. This observation is produced only when you specify the NORM option with METHOD=AGK, METHOD=IQR, METHOD=MAD, or METHOD=SPACING or when you specify the SNORM option with METHOD=SPACING.
NObsRead	number of physical records read
NObsUsed	number of physical records used in the analysis
NObsMiss	number of physical records containing missing values
P_n	percentiles of each variable, as specified by the PCTLPTS= option. The argument n is any real number such that $0 \leq n \leq 100$
SumFreqsRead	sum of the frequency variable (or the sum of NObsUsed ones when there is no frequency variable) for all observations read
SumFreqsUsed	sum of the frequency variable (or the sum of NObsUsed ones when there is no frequency variable) for all observations used in the analysis
SumWeightsRead	sum of the weight variable (or the sum of NObsUsed ones when there is no weight variable) for all observations read
SumWeightsUsed	sum of the weight variable (or the sum of NObsUsed ones when there is no weight variable) for all observations used in the analysis

Displayed Output

If you specify the PSTAT option, PROC STDIZE displays the following statistics for each variable:

- the name of the variable, Name
- the location estimate, Location
- the scale estimate, Scale
- the norm estimate, Norm (when you specify the NORM option with METHOD=AGK, METHOD=IQR, METHOD=MAD, or METHOD=SPACING or when you specify the SNORM option with METHOD=SPACING)
- sum of nonmissing positive frequencies, N
- sum of nonmissing positive weights if the WEIGHT statement is specified, Sum of Weights

ODS Table Names

PROC STDIZE assigns a name to the single table it creates. You can use this name to reference the table when using the Output Delivery System (ODS) to select a table or create an output data set. This name is listed in Table 114.3. For more information about ODS, see Chapter 22, “Using the Output Delivery System.”

Table 114.3 ODS Table Produced by PROC STDIZE

ODS Table Name	Description	Statement	Option
Statistics	Location and Scale Measures	PROC	PSTAT

Example: STDIZE Procedure

Example 114.1: Standardization of Variables in Cluster Analysis

To illustrate the effect of standardization in cluster analysis, this example uses the Fish data set described in the “Getting Started” section of Chapter 44, “The FASTCLUS Procedure.” The numbers are measurements taken on 159 fish caught from the same lake (Laengelmaevesi) near Tampere in Finland (Puranen 1917). The fish data set is available from the Sashelp library.

The species (bream, parkki, pike, perch, roach, smelt, and whitefish), weight, three different length measurements (measured from the nose of the fish to the beginning of its tail, the notch of its tail, and the end of its tail), height, and width of each fish are recorded.

A couple of new variables are created in the Fish data set: Weight3 and logLengthRatio. The weight of a fish indicates its size—a heavier pike tends to be larger than a lighter pike. To get a one-dimensional measure of the size of a fish, take the cubic root of the weight (Weight3). The variables Height, Width, Length1, Length2, and Length3 are rescaled in order to adjust for dimensionality. The logLengthRatio variable measures the tail length.

Because the new variables Weight3–logLengthRatio depend on the variable Weight, observations with missing values for Weight are not added to the data set. Consequently, there are 157 observations in the SAS data set Sashelp.Fish.

Before you perform a cluster analysis on coordinate data, it is necessary to consider scaling or transforming the variables since variables with large variances tend to have a larger effect on the resulting clusters than variables with small variances do.

This example uses three different approaches to standardize or transform the data prior to the cluster analysis. The first approach uses several standardization methods provided in the STDIZE procedure. However, since standardization is not always appropriate prior to the clustering (see Milligan and Cooper (1987) for a Monte Carlo study on various methods of variable standardization), the second approach performs the cluster analysis with no standardization. The third approach invokes the ACECLUS procedure to transform the data into a within-cluster covariance matrix.

The clustering is performed by the FASTCLUS procedure to find seven clusters. Note that the variables Length2 and Length3 are eliminated from this analysis since they both are significantly and highly correlated with the variable Length1. The correlation coefficients are 0.9958 and 0.9604, respectively. An output data set is created, and the FREQ procedure is invoked to compare the clusters with the species classification.

The DATA step is as follows:

```
title 'Fish Measurement Data';

data Fish;
  set sashelp.fish;
  if Weight <= 0 or Weight = . then delete;
  Weight3 = Weight ** (1/3);
  Height = Height / Weight3;
  Width = Width / Weight3;
  Length1 = Length1 / Weight3;
```

```

    Length2 = Length2 / Weight3;
    Length3 = Length3 / Weight3;
    LogLengthRatio = log(Length3 / Length1);
run;

```

The following macro, Std, standardizes the Fish data. The macro reads a single argument, mtd, which selects the METHOD= specification to be used in PROC STDIZE.

```

/*--- macro for standardization ---*/

%macro Std(mtd);
    title2 "Data are Standardized by PROC STDIZE with METHOD= &mtd";
    proc stdize data=fish out=sdzout method=&mtd;
        var Length1 logLengthRatio Height Width Weight3;
    run;
%mend Std;

```

The following macro, FastFreq, includes a PROC FASTCLUS statement for performing cluster analysis and a PROC FREQ statement for crosstabulating species with the cluster membership information that is derived from the previous PROC FASTCLUS statement. The macro reads a single argument, ds, which selects the input data set to be used in PROC FASTCLUS.

```

/*--- macro for clustering and crosstabulating ---*/
/*--- cluster membership with species ---*/

%macro FastFreq(ds);
    proc fastclus data=&ds out=clust maxclusters=7 maxiter=100 noprint;
        var Length1 logLengthRatio Height Width Weight3;
    run;

    proc freq data=clust;
        tables species*cluster;
    run;
%mend FastFreq;

```

The following analysis (labeled ‘Approach 1’) includes 18 different methods of standardization followed by clustering. Since there is a large amount of output from this approach, only results from METHOD=STD, METHOD=RANGE, METHOD=AGK(0.14), and METHOD=SPACING(0.14) are shown. The following statements produce [Output 114.1.1](#) through [Output 114.1.4](#).

```

/*      Approach 1: data are standardized by PROC STDIZE      */

%Std(MEAN);
%FastFreq(sdzout);

%Std(MEDIAN);
%FastFreq(sdzout);

%Std(SUM);
%FastFreq(sdzout);

%Std(EUCLEN);
%FastFreq(sdzout);

```

```
%Std(USTD);
%FastFreq(sdzout);

%Std(STD);
%FastFreq(sdzout);

%Std(RANGE);
%FastFreq(sdzout);

%Std(MIDRANGE);
%FastFreq(sdzout);

%Std(MAXABS);
%FastFreq(sdzout);

%Std(IQR);
%FastFreq(sdzout);

%Std(MAD);
%FastFreq(sdzout);

%Std(AGK(.14));
%FastFreq(sdzout);

%Std(SPACING(.14));
%FastFreq(sdzout);

%Std(ABW(5));
%FastFreq(sdzout);

%Std(AWAVE(5));
%FastFreq(sdzout);

%Std(L(1));
%FastFreq(sdzout);

%Std(L(1.5));
%FastFreq(sdzout);

%Std(L(2));
%FastFreq(sdzout);
```

Output 114.1.1 Data Are Standardized by PROC STDIZE with METHOD=STD

Fish Measurement Data
Data are Standardized by PROC STDIZE with METHOD= STD

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Species by CLUSTER								
	Species	1	2	3	4	5	6	7	Total
	Bream	0	0	0	0	0	34	0	34
		0.00	0.00	0.00	0.00	0.00	21.66	0.00	21.66
		0.00	0.00	0.00	0.00	0.00	100.00	0.00	
		0.00	0.00	0.00	0.00	0.00	100.00	0.00	
	Parkki	0	0	0	0	11	0	0	11
		0.00	0.00	0.00	0.00	7.01	0.00	0.00	7.01
		0.00	0.00	0.00	0.00	100.00	0.00	0.00	
		0.00	0.00	0.00	0.00	100.00	0.00	0.00	
	Perch	0	17	0	12	0	0	27	56
		0.00	10.83	0.00	7.64	0.00	0.00	17.20	35.67
		0.00	30.36	0.00	21.43	0.00	0.00	48.21	
		0.00	89.47	0.00	92.31	0.00	0.00	54.00	
	Pike	17	0	0	0	0	0	0	17
		10.83	0.00	0.00	0.00	0.00	0.00	0.00	10.83
		100.00	0.00	0.00	0.00	0.00	0.00	0.00	
		100.00	0.00	0.00	0.00	0.00	0.00	0.00	
	Roach	0	0	0	0	0	0	19	19
		0.00	0.00	0.00	0.00	0.00	0.00	12.10	12.10
		0.00	0.00	0.00	0.00	0.00	0.00	100.00	
		0.00	0.00	0.00	0.00	0.00	0.00	38.00	
	Smelt	0	0	13	0	0	0	1	14
		0.00	0.00	8.28	0.00	0.00	0.00	0.64	8.92
		0.00	0.00	92.86	0.00	0.00	0.00	7.14	
		0.00	0.00	100.00	0.00	0.00	0.00	2.00	
	Whitefish	0	2	0	1	0	0	3	6
		0.00	1.27	0.00	0.64	0.00	0.00	1.91	3.82
		0.00	33.33	0.00	16.67	0.00	0.00	50.00	
		0.00	10.53	0.00	7.69	0.00	0.00	6.00	
	Total	17	19	13	13	11	34	50	157
		10.83	12.10	8.28	8.28	7.01	21.66	31.85	100.00

Output 114.1.2 Data Are Standardized by PROC STDIZE with METHOD=RANGE

Fish Measurement Data
Data are Standardized by PROC STDIZE with METHOD= RANGE

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Species by CLUSTER							
	Species	CLUSTER(Cluster)						
		1	2	3	4	5	6	7
Bream		0	0	34	0	0	0	0
		0.00	0.00	21.66	0.00	0.00	0.00	0.00
		0.00	0.00	100.00	0.00	0.00	0.00	0.00
		0.00	0.00	100.00	0.00	0.00	0.00	0.00
Parkki		0	0	0	0	0	11	0
		0.00	0.00	0.00	0.00	0.00	7.01	0.00
		0.00	0.00	0.00	0.00	0.00	100.00	0.00
		0.00	0.00	0.00	0.00	0.00	100.00	0.00
Perch		0	0	0	9	20	0	27
		0.00	0.00	0.00	5.73	12.74	0.00	17.20
		0.00	0.00	0.00	16.07	35.71	0.00	48.21
		0.00	0.00	0.00	29.03	86.96	0.00	100.00
Pike		17	0	0	0	0	0	17
		10.83	0.00	0.00	0.00	0.00	0.00	10.83
		100.00	0.00	0.00	0.00	0.00	0.00	0.00
		100.00	0.00	0.00	0.00	0.00	0.00	0.00
Roach		0	0	0	19	0	0	19
		0.00	0.00	0.00	12.10	0.00	0.00	12.10
		0.00	0.00	0.00	100.00	0.00	0.00	0.00
		0.00	0.00	0.00	61.29	0.00	0.00	0.00
Smelt		0	14	0	0	0	0	14
		0.00	8.92	0.00	0.00	0.00	0.00	8.92
		0.00	100.00	0.00	0.00	0.00	0.00	0.00
		0.00	100.00	0.00	0.00	0.00	0.00	0.00
Whitefish		0	0	0	3	3	0	6
		0.00	0.00	0.00	1.91	1.91	0.00	3.82
		0.00	0.00	0.00	50.00	50.00	0.00	0.00
		0.00	0.00	0.00	9.68	13.04	0.00	0.00
Total		17	14	34	31	23	11	27
		10.83	8.92	21.66	19.75	14.65	7.01	17.20

Output 114.1.3 Data Are Standardized by PROC STDIZE with METHOD=AGK(0.14)

Fish Measurement Data
Data are Standardized by PROC STDIZE with METHOD= AGK(.14)

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Species by CLUSTER							
	Species	CLUSTER(Cluster)						
		1	2	3	4	5	6	7
Bream		0	0	34	0	0	0	0
		0.00	0.00	21.66	0.00	0.00	0.00	0.00
		0.00	0.00	100.00	0.00	0.00	0.00	0.00
		0.00	0.00	100.00	0.00	0.00	0.00	0.00
Parkki		11	0	0	0	0	0	0
		7.01	0.00	0.00	0.00	0.00	0.00	0.00
		100.00	0.00	0.00	0.00	0.00	0.00	0.00
		100.00	0.00	0.00	0.00	0.00	0.00	0.00
Perch		0	0	0	3	0	20	33
		0.00	0.00	0.00	1.91	0.00	12.74	21.02
		0.00	0.00	0.00	5.36	0.00	35.71	58.93
		0.00	0.00	0.00	13.04	0.00	86.96	94.29
Pike		0	0	0	0	17	0	0
		0.00	0.00	0.00	0.00	10.83	0.00	0.00
		0.00	0.00	0.00	0.00	100.00	0.00	0.00
		0.00	0.00	0.00	0.00	100.00	0.00	0.00
Roach		0	0	0	17	0	0	2
		0.00	0.00	0.00	10.83	0.00	0.00	1.27
		0.00	0.00	0.00	89.47	0.00	0.00	10.53
		0.00	0.00	0.00	73.91	0.00	0.00	5.71
Smelt		0	14	0	0	0	0	0
		0.00	8.92	0.00	0.00	0.00	0.00	0.00
		0.00	100.00	0.00	0.00	0.00	0.00	0.00
		0.00	100.00	0.00	0.00	0.00	0.00	0.00
Whitefish		0	0	0	3	0	3	0
		0.00	0.00	0.00	1.91	0.00	1.91	0.00
		0.00	0.00	0.00	50.00	0.00	50.00	0.00
		0.00	0.00	0.00	13.04	0.00	13.04	0.00
Total		11	14	34	23	17	23	35
		7.01	8.92	21.66	14.65	10.83	14.65	22.29

Output 114.1.4 Data Are Standardized by PROC STDIZE with METHOD=SPACING(0.14)

Fish Measurement Data
Data are Standardized by PROC STDIZE with METHOD= SPACING(.14)

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Species by CLUSTER							
	Species	CLUSTER(Cluster)						
		1	2	3	4	5	6	7
Bream		0	0	0	0	0	0	34
		0.00	0.00	0.00	0.00	0.00	0.00	21.66
		0.00	0.00	0.00	0.00	0.00	0.00	100.00
		0.00	0.00	0.00	0.00	0.00	0.00	100.00
Parkki		0	0	11	0	0	0	11
		0.00	0.00	7.01	0.00	0.00	0.00	7.01
		0.00	0.00	100.00	0.00	0.00	0.00	0.00
		0.00	0.00	100.00	0.00	0.00	0.00	0.00
Perch		20	0	0	0	0	36	0
		12.74	0.00	0.00	0.00	0.00	22.93	0.00
		35.71	0.00	0.00	0.00	0.00	64.29	0.00
		86.96	0.00	0.00	0.00	0.00	94.74	0.00
Pike		0	17	0	0	0	0	17
		0.00	10.83	0.00	0.00	0.00	0.00	10.83
		0.00	100.00	0.00	0.00	0.00	0.00	0.00
		0.00	100.00	0.00	0.00	0.00	0.00	0.00
Roach		0	0	0	17	0	2	19
		0.00	0.00	0.00	10.83	0.00	1.27	0.00
		0.00	0.00	0.00	89.47	0.00	10.53	0.00
		0.00	0.00	0.00	85.00	0.00	5.26	0.00
Smelt		0	0	0	0	14	0	14
		0.00	0.00	0.00	0.00	8.92	0.00	8.92
		0.00	0.00	0.00	0.00	100.00	0.00	0.00
		0.00	0.00	0.00	0.00	100.00	0.00	0.00
Whitefish		3	0	0	3	0	0	6
		1.91	0.00	0.00	1.91	0.00	0.00	3.82
		50.00	0.00	0.00	50.00	0.00	0.00	0.00
		13.04	0.00	0.00	15.00	0.00	0.00	0.00
Total		23	17	11	20	14	38	157
		14.65	10.83	7.01	12.74	8.92	24.20	21.66

The following analysis (labeled 'Approach 2') applies the cluster analysis directly to the original data. The following statements produce [Output 114.1.5](#).

```

/*          Approach 2: data are untransformed          */

title2 'Data are Untransformed';
%FastFreq(fish);

```

Output 114.1.5 Untransformed Data**Fish Measurement Data
Data are Untransformed****The FREQ Procedure**

Frequency Percent Row Pct Col Pct	Table of Species by CLUSTER								
	Species	CLUSTER(Cluster)							Total
		1	2	3	4	5	6	7	
Bream		13	0	0	0	0	0	21	34
		8.28	0.00	0.00	0.00	0.00	0.00	13.38	21.66
		38.24	0.00	0.00	0.00	0.00	0.00	61.76	
		44.83	0.00	0.00	0.00	0.00	0.00	47.73	
Parkki		2	3	0	0	6	0	0	11
		1.27	1.91	0.00	0.00	3.82	0.00	0.00	7.01
		18.18	27.27	0.00	0.00	54.55	0.00	0.00	
		6.90	18.75	0.00	0.00	15.38	0.00	0.00	
Perch		8	9	0	1	20	0	18	56
		5.10	5.73	0.00	0.64	12.74	0.00	11.46	35.67
		14.29	16.07	0.00	1.79	35.71	0.00	32.14	
		27.59	56.25	0.00	6.67	51.28	0.00	40.91	
Pike		0	0	10	0	1	4	2	17
		0.00	0.00	6.37	0.00	0.64	2.55	1.27	10.83
		0.00	0.00	58.82	0.00	5.88	23.53	11.76	
		0.00	0.00	100.00	0.00	2.56	100.00	4.55	
Roach		3	4	0	0	12	0	0	19
		1.91	2.55	0.00	0.00	7.64	0.00	0.00	12.10
		15.79	21.05	0.00	0.00	63.16	0.00	0.00	
		10.34	25.00	0.00	0.00	30.77	0.00	0.00	
Smelt		0	0	0	14	0	0	0	14
		0.00	0.00	0.00	8.92	0.00	0.00	0.00	8.92
		0.00	0.00	0.00	100.00	0.00	0.00	0.00	
		0.00	0.00	0.00	93.33	0.00	0.00	0.00	
Whitefish		3	0	0	0	0	0	3	6
		1.91	0.00	0.00	0.00	0.00	0.00	1.91	3.82
		50.00	0.00	0.00	0.00	0.00	0.00	50.00	
		10.34	0.00	0.00	0.00	0.00	0.00	6.82	
Total		29	16	10	15	39	4	44	157
		18.47	10.19	6.37	9.55	24.84	2.55	28.03	100.00

The following analysis (labeled 'Approach 3') transforms the original data with the ACECLUS procedure and creates a TYPE=ACE output data set that is used as an input data set for the cluster analysis. The following statements produce [Output 114.1.6](#).

```

/*    Approach 3: data are transformed by PROC ACECLUS    */

title2 'Data are Transformed by PROC ACECLUS';
proc aceclus data=fish out=ace p=.02 noprint;
    var Length1 logLengthRatio Height Width Weight3;
run;
%FastFreq(ace);

```

Output 114.1.6 Data Are Transformed by PROC ACECLUS**Fish Measurement Data**
Data are Transformed by PROC ACECLUS**The FREQ Procedure**

Frequency Percent Row Pct Col Pct	Table of Species by CLUSTER								
	Species	CLUSTER(Cluster)							Total
		1	2	3	4	5	6	7	
Bream		13	0	0	0	0	0	21	34
		8.28	0.00	0.00	0.00	0.00	0.00	13.38	21.66
		38.24	0.00	0.00	0.00	0.00	0.00	61.76	
		44.83	0.00	0.00	0.00	0.00	0.00	47.73	
Parkki		2	3	0	0	6	0	0	11
		1.27	1.91	0.00	0.00	3.82	0.00	0.00	7.01
		18.18	27.27	0.00	0.00	54.55	0.00	0.00	
		6.90	18.75	0.00	0.00	15.38	0.00	0.00	
Perch		8	9	0	1	20	0	18	56
		5.10	5.73	0.00	0.64	12.74	0.00	11.46	35.67
		14.29	16.07	0.00	1.79	35.71	0.00	32.14	
		27.59	56.25	0.00	6.67	51.28	0.00	40.91	
Pike		0	0	10	0	1	4	2	17
		0.00	0.00	6.37	0.00	0.64	2.55	1.27	10.83
		0.00	0.00	58.82	0.00	5.88	23.53	11.76	
		0.00	0.00	100.00	0.00	2.56	100.00	4.55	
Roach		3	4	0	0	12	0	0	19
		1.91	2.55	0.00	0.00	7.64	0.00	0.00	12.10
		15.79	21.05	0.00	0.00	63.16	0.00	0.00	
		10.34	25.00	0.00	0.00	30.77	0.00	0.00	
Smelt		0	0	0	14	0	0	0	14
		0.00	0.00	0.00	8.92	0.00	0.00	0.00	8.92
		0.00	0.00	0.00	100.00	0.00	0.00	0.00	
		0.00	0.00	0.00	93.33	0.00	0.00	0.00	
Whitefish		3	0	0	0	0	0	3	6
		1.91	0.00	0.00	0.00	0.00	0.00	1.91	3.82
		50.00	0.00	0.00	0.00	0.00	0.00	50.00	
		10.34	0.00	0.00	0.00	0.00	0.00	6.82	
Total		29	16	10	15	39	4	44	157
		18.47	10.19	6.37	9.55	24.84	2.55	28.03	100.00

Table 114.4 displays a table summarizing each classification results. In this table, the first column represents the standardization method, the second column represents the number of clusters that the seven species are classified into, and the third column represents the total number of observations that are misclassified.

Table 114.4 Summary of Clustering Results

Method of Standardization	Number of Clusters	Misclassification
MEAN	5	71
MEDIAN	5	71
SUM	6	51
EUCLEN	6	45
USTD	6	45
STD	5	33
RANGE	7	32
MIDRANGE	7	32
MAXABS	7	26
IQR	5	28
MAD	4	35
ABW(5)	6	34
AWAVE(5)	6	29
AGK(0.14)	7	28
SPACING(0.14)	7	25
L(1)	6	41
L(1.5)	5	33
L(2)	5	33
untransformed	5	71
PROC ACECLUS	5	71

Consider the results displayed in [Output 114.1.1](#). In that analysis, the method of standardization is STD, and the number of clusters and the number of misclassifications are computed as shown in [Table 114.5](#).

Table 114.5 Computations of Numbers of Clusters and Misclassification When Standardization Method Is STD

Species	Cluster Number	Misclassification in Each Species
Bream	6	0
Parkki	5	0
Perch	7	29
Pike	1	0
Roach	7	0
Smelt	3	1
Whitefish	7	3

In [Output 114.1.1](#), the bream species is classified as cluster 6 since all 34 bream are categorized into cluster 6 with no misclassification. A similar pattern is seen with the roach, parkki, pike, and smelt species.

For the whitefish species, two fish are categorized into cluster 2, one fish is categorized into cluster 4, and three fish are categorized into cluster 7. Because the majority of this species is categorized into cluster 7, it is recorded in [Table 114.5](#) as being classified as cluster 7 with 3 misclassifications. A similar pattern is seen with the perch species: it is classified as cluster 7 with 29 misclassifications.

In summary, when the standardization method is STD, seven species of fish are classified into only five clusters and the total number of misclassified observations is 33.

The result of this analysis demonstrates that when variables are standardized by the STDIZE procedure with methods including RANGE, MIDRANGE, MAXABS, AGK(0.14), and SPACING(0.14), the FASTCLUS procedure produces the correct number of clusters and less misclassification than it does when other standardization methods are used. The SPACING method attains the best result, probably because the variables Length1 and Height both exhibit marked groupings (bimodality) in their distributions.

References

- Art, D., Gnanadesikan, R., and Kettenring, J. R. (1982). "Data-Based Metrics for Cluster Analysis." *Utilitas Mathematica* 75–99.
- Goodall, C. (1983). "M-Estimators of Location: An Outline of Theory." In *Understanding Robust and Exploratory Data Analysis*, edited by D. C. Hoaglin, M. Mosteller, and J. W. Tukey, 339–403. New York: John Wiley & Sons.
- Iglewicz, B. (1983). "Robust Scale Estimators and Confidence Intervals for Location." In *Understanding Robust and Exploratory Data Analysis*, edited by D. C. Hoaglin, M. Mosteller, and J. W. Tukey, 404–431. New York: John Wiley & Sons.
- Jain, R., and Chlamtac, I. (1985). "The P^2 Algorithm for Dynamic Calculation of Quantiles and Histograms without Storing Observations." *Communications of the ACM* 28:1076–1085.
- Janssen, P., Marron, J. S., Veraverbeke, N., and Sarle, W. S. (1995). "Scale Measures for Bandwidth Selection." *Journal of Nonparametric Statistics* 5:359–380.
- Milligan, G. W., and Cooper, M. C. (1987). "A Study of Variable Standardization." College of Administrative Science Working Paper Series, No. 87-63, Ohio State University.
- Puranen, J. (1917). "Fish Catch data set (1917)." Journal of Statistics Education Data Archive. Accessed October 29, 2019. <https://www.amstat.org/publications/jse/datasets/fishcatch.txt>.

Subject Index

- AGK estimate
 - STDIZE procedure, [9444](#)
- cluster analysis (STDIZE)
 - standardizing, [9449](#)
- clustering and scaling
 - STDIZE procedure, example, [9449](#)
- Euclidean length
 - STDIZE procedure, [9443](#)
- fractional frequencies
 - STDIZE procedure, [9439](#)
- Minkowski metric
 - STDIZE procedure, [9443](#)
- missing values
 - STDIZE procedure, [9436](#), [9437](#)
- percentiles
 - weighted, [9446](#)
- quantile computation
 - STDIZE procedure, [9426](#), [9444](#)
- robust
 - estimators (STDIZE), [9443](#)
- scaling variables
 - STDIZE procedure, [9449](#)
- spacing
 - STDIZE procedure, [9444](#)
- standardizing
 - cluster analysis (STDIZE), [9449](#)
 - values (STDIZE), [9426](#)
- STDIZE procedure
 - AGK estimate, [9444](#)
 - Andrew's wave estimate, [9444](#)
 - breakdown point and efficiency, [9443](#)
 - comparisons of quantile computation, PCTLMTD option, [9444](#)
 - computational methods, PCTLDEF option, [9445](#)
 - Euclidean length, [9443](#)
 - examples, [9426](#), [9449](#)
 - final output value, [9426](#)
 - formulas for statistics, [9443](#)
 - fractional frequencies, [9439](#)
 - fuzz factor, [9435](#)
 - Huber's estimate, [9444](#)
 - initial estimates for A estimates, [9435](#)
 - input data set (METHOD=IN()), [9442](#)
 - methods resistant to clustering, [9443](#)
 - methods resistant to outliers, [9429](#), [9443](#)
 - Minkowski metric, [9443](#)
 - missing values, [9436](#), [9437](#), [9446](#)
 - normalization, [9436](#), [9437](#)
 - one-pass quantile computations, [9444](#)
 - OUT= data set, [9435](#), [9446](#)
 - output data sets, [9436](#), [9437](#), [9446](#)
 - output table names, [9448](#)
 - OUTSTAT= data set, [9447](#)
 - quantile computation, [9426](#), [9444](#)
 - robust estimators, [9443](#)
 - spacing, [9444](#)
 - standardization methods, [9426](#), [9442](#)
 - standardization with weights, [9440](#)
 - Tukey's biweight estimate, [9431](#), [9444](#)
 - tuning constant, [9431](#), [9442](#)
 - unstandardization, [9438](#)
 - weights, [9440](#)
- weighted percentiles, [9446](#)

Syntax Index

ADD= option
PROC STDIZE statement, [9435](#)

BY statement
STDIZE procedure, [9438](#)

DATA= option
PROC STDIZE statement, [9435](#)

FREQ statement
STDIZE procedure, [9439](#)

FUZZ= option
PROC STDIZE statement, [9435](#)

INITIAL= option
PROC STDIZE statement, [9435](#)

KEEPLEN option
PROC STDIZE statement, [9435](#)

METHOD= option
PROC STDIZE statement, [9435](#)

MISSING= option
PROC STDIZE statement, [9436](#)

MULT= option
PROC STDIZE statement, [9436](#)

NMARKERS= option
PROC STDIZE statement, [9436](#)

NOMISS option
PROC STDIZE statement, [9436](#)

NORM option
PROC STDIZE statement, [9436](#)

NOTRUNCATE option
FREQ statement (STDIZE), [9439](#)

OPREFIX option
PROC STDIZE statement, [9436](#)

OUT= option
PROC STDIZE statement, [9436](#)

OUTSTAT= option
PROC STDIZE statement, [9437](#)

PCTLDEF= option
PROC STDIZE statement, [9437](#)

PCTLMTD option
PROC STDIZE statement, [9437](#)

PCTLPTS option
PROC STDIZE statement, [9437](#)

PROC STDIZE statement, *see* STDIZE procedure

PSTAT option
PROC STDIZE statement, [9437](#)

REPLACE option
PROC STDIZE statement, [9437](#)

REONLY option
PROC STDIZE statement, [9437](#)

SNORM option
PROC STDIZE statement, [9437](#)

SPREFIX option
PROC STDIZE statement, [9437](#)

STDIZE procedure
syntax, [9433](#)

STDIZE procedure, BY statement, [9438](#)

STDIZE procedure, FREQ statement, [9439](#)

NOTRUNCATE option, [9439](#)

STDIZE procedure, LOCATION statement, [9440](#)

STDIZE procedure, PROC STDIZE statement, [9433](#)

ADD= option, [9435](#)

DATA= option, [9435](#)

FUZZ= option, [9435](#)

INITIAL= option, [9435](#)

KEEPLEN, [9435](#)

METHOD= option, [9435](#)

MISSING= option, [9436](#)

MULT= option, [9436](#)

NMARKERS= option, [9436](#)

NOMISS option, [9436](#)

NORM option, [9436](#)

OPREFIX option, [9436](#)

OUT= option, [9436](#)

OUTSTAT= option, [9437](#)

PCTLDEF= option, [9437](#)

PCTLMTD option, [9437](#)

PCTLPTS option, [9437](#)

PSTAT option, [9437](#)

REPLACE option, [9437](#)

REONLY option, [9437](#)

SNORM option, [9437](#)

SPREFIX option, [9437](#)

UNSTD option, [9438](#)

VARDEF option, [9438](#)

STDIZE procedure, SCALE statement, [9440](#)

STDIZE procedure, VAR statement, [9440](#)

STDIZE procedure, WGT statement, [9440](#)

UNSTD option
PROC STDIZE statement, [9438](#)

VAR statement

STDIZE procedure, [9440](#)

VARDEF option

PROC STDIZE statement, [9438](#)

WGT statement

STDIZE procedure, [9440](#)