

SAS/STAT 15.2[®] User's Guide Shared Concepts and Topics in High Performance Statistical Procedures

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 21

Shared Concepts and Topics in High-Performance Statistical Procedures

Contents

Common Features of SAS High-Performance Statistical Procedures	536
Syntax Common to SAS High-Performance Statistical Procedures	536
CLASS Statement	536
FREQ Statement	540
ID Statement	540
SELECTION Statement	540
VAR Statement	546
WEIGHT Statement	546
Model Selection	546
Methods	546
Full Model Fitted	546
Forward Selection	546
Backward Elimination	549
Stepwise Selection	549
Forward-Swap Selection	551
Least Angle Regression	552
Lasso Selection	552
Adaptive Lasso Selection	553
References	553

Common Features of SAS High-Performance Statistical Procedures

SAS high-performance statistical procedures behave in many ways like other procedures in the SAS System. This chapter provides details about and describes common syntax elements that are supported by many high-performance statistical procedures. Any deviation by a high-performance statistical procedure from the common syntax is documented in the specific chapter for the procedure.

Syntax Common to SAS High-Performance Statistical Procedures

CLASS Statement

CLASS *variable* <(options)>... <*variable* <(options)>> </global-options> ;

The CLASS statement names the classification variables to be used as explanatory variables in the analysis. These variables enter the analysis not through their values, but through levels to which the unique values are mapped. For more information about these mappings, see the section “[Levelization of Classification Variables](#)” on page 387.

If a CLASS statement is specified, it must precede the MODEL statement in high-performance statistical procedures that support a MODEL statement.

If the procedure permits a classification variable as a response (dependent variable or target), the response does not need to be specified in the CLASS statement.

You can specify options either as individual variable *options* or as *global-options*. You can specify *options* for each variable by enclosing the options in parentheses after the variable name. You can also specify *global-options* for the CLASS statement by placing them after a slash (/). *Global-options* are applied to all the variables that are specified in the CLASS statement. If you specify more than one CLASS statement, the *global-options* that are specified in any one CLASS statement apply to all CLASS statements. However, individual CLASS variable *options* override the *global-options*.

You can specify the following values for either an *option* or a *global-option* (except for the HPLMIXED procedure, which does not support options in this statement):

DESCENDING

DESC

reverses the sort order of the classification variable. If both the DESCENDING and [ORDER=](#) options are specified, high-performance statistical procedures order the categories according to the ORDER= option and then reverse that order.

ORDER=DATA | FORMATTED | INTERNAL**ORDER=FREQ | FREQDATA | FREQFORMATTED | FREQINTERNAL**

specifies the sort order for the levels of classification variables. This ordering determines which parameters in the model correspond to each level in the data. By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine-dependent. When ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values.

Table 21.1 shows how high-performance statistical procedures interpret values of the ORDER= option.

Table 21.1 Sort Order

Value of ORDER=	Levels Sorted By
DATA	Order of appearance in the input data set
FORMATTED	External formatted values, except for numeric variables that have no explicit format, which are sorted by their unformatted (internal) values
FREQ	Descending frequency count (levels that have more observations come earlier in the order)
FREQDATA	Order of descending frequency count, and within counts by order of appearance in the input data set when counts are tied
FREQFORMATTED	Order of descending frequency count, and within counts by formatted value when counts are tied
FREQINTERNAL	Order of descending frequency count, and within counts by unformatted (internal) value when counts are tied
INTERNAL	Unformatted value

For more information about sort order, see the chapter about the SORT procedure in *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Programmers Guide: Essentials*.

REF= 'level' | keyword

REFERENCE= 'level' | keyword

specifies the reference level that is used when you specify **PARAM=REFERENCE**. For an individual (but not a global) variable REF= *option*, you can specify the *level* of the variable to use as the reference level. Specify the formatted value of the variable if a format is assigned. For a REF= *option* or *global-option*, you can use one of the following *keywords*. The default is REF=LAST.

FIRST designates the first ordered level as reference.

LAST designates the last ordered level as reference.

If you choose a reference level for any CLASS variable, all variables are parameterized in the reference parameterization for computational efficiency. In other words, high-performance statistical procedures apply a single parameterization method to all classification variables.

Suppose that the variable temp has three levels ('hot', 'warm', and 'cold') and that the variable gender has two levels ('M' and 'F'). The following statements fit a logistic regression model:

```
proc hplogistic;
  class gender(ref='F') temp;
  model y = gender gender*temp;
run;
```

Both CLASS variables are in reference parameterization in this model. The reference levels are 'F' for the variable gender and 'warm' for the variable temp, because the statements are equivalent to the following statements:

```
proc hplogistic;
  class gender(ref='F') temp(ref=last);
  model y = gender gender*temp;
run;
```

SPLIT

requests that the columns of the design matrix that correspond to any effect that contains a split classification variable can be selected to enter or leave a model independently of the other design columns of that effect. This option is specific to the HPREG procedure.

Suppose that the variable temp has three levels ('hot', 'warm', and 'cold'), that the variable gender has two levels ('M' and 'F'), and that the variables are used in a PROC HPREG run as follows:

```
proc hpreg;
  class temp gender / split;
  model y = gender gender*temp;
run;
```

The two effects in the MODEL statement are split into eight independent effects. The effect “gender” is split into two effects that are labeled “gender_M” and “gender_F”. The effect “gender*temp” is split into six effects that are labeled “gender_M*temp_hot”, “gender_F*temp_hot”, “gender_M*temp_warm”, “gender_F*temp_warm”, “gender_M*temp_cold”, and “gender_F*temp_cold”. The previous PROC HPREG step is equivalent to the following:

```
proc hpreg;
  model y = gender_M gender_F
           gender_M*temp_hot gender_F*temp_hot
           gender_M*temp_warm gender_F*temp_warm
           gender_M*temp_cold gender_F*temp_cold;
run;
```

The SPLIT option can be used on individual classification variables. For example, consider the following PROC HPREG step:

```
proc hpreg;
  class temp(split) gender;
  model y = gender gender*temp;
run;
```

In this case, the effect “gender” is not split and the effect “gender*temp” is split into three effects, which are labeled “gender*temp_hot”, “gender*temp_warm”, and “gender*temp_cold”. Furthermore, each of these three split effects now has two parameters that correspond to the two levels of “gender.” The PROC HPREG step is equivalent to the following:

```
proc hpreg;
  class gender;
  model y = gender gender*temp_hot gender*temp_warm gender*temp_cold;
run;
```

You can specify the following *global-options*:

MISSING

treats missing values (“.”, “.A”, . . . , “.Z” for numeric variables and blanks for character variables) as valid values for the CLASS variable.

If you do not specify the MISSING option, observations that have missing values for CLASS variables are removed from the analysis, even if the CLASS variables are not used in the model formulation.

PARAM=keyword

specifies the parameterization method for the classification variable or variables. You can specify the following *keywords*:

GLM specifies a less-than-full-rank reference cell coding. This parameterization is used in, for example, the GLM, MIXED, and GLIMMIX procedures in SAS/STAT.

REFERENCE specifies a reference cell encoding. You can choose the reference value by specifying an option for a specific *variable* or set of *variables* in the CLASS statement, or designate the first or last ordered value by specifying a *global-option*. The default is REF=LAST.

For example, suppose that the variable temp has three levels ('hot', 'warm', and 'cold'), that the variable gender has two levels ('M' and 'F'), and that the variables are used in a CLASS statement as follows:

```
class gender(ref='F') temp / param=ref;
```

Then 'F' is used as the reference level for gender and 'warm' is used as the reference level for temp.

The GLM parameterization is the default. For more information about how parameterization of classification variables affects the construction and interpretation of model effects, see the section “[Parameterization of Model Effects](#)” on page 389.

TRUNCATE<=n>

specifies the truncation width of formatted values of CLASS variables when the optional *n* is specified.

If *n* is not specified, the TRUNCATE option requests that classification levels be determined by using no more than the first 16 characters of the formatted values of CLASS variables.

FREQ Statement

FREQ *variable* ;

The *variable* in the FREQ statement identifies a numeric variable in the data set that contains the frequency of occurrence for each observation. High-performance statistical procedures that support the FREQ statement treat each observation as if it appeared f times, where f is the value of the FREQ variable for the observation. If the frequency value is not an integer, it is truncated to an integer. If the frequency value is less than 1 or missing, the observation is not used in the analysis. When the FREQ statement is not specified, each observation is assigned a frequency of 1.

ID Statement

ID *variables* ;

The ID statement lists one or more variables from the input data set that are transferred to output data sets that are created by high-performance statistical procedures, provided that the output data set contains one (or more) records per input observation. For example, when an OUTPUT statement is used to produce observationwise scores or prediction statistics, ID variables are added to the output data set.

By default, high-performance statistical procedures do not include all variables from the input data set in output data sets. In the following statements, a logistic regression model is fit and then scored. The input and output data are stored in the Greenplum database. The output data set contains three columns (p, account, trans_date) where p is computed during the scoring process and the account and transaction date are transferred from the input data set. (High-performance statistical procedures also transfer any distribution keys from the input to the output data.)

```
libname GPLib greenplm server=gpdc user=XXX password=YYY
      database=ZZZ;
proc hplogistic data=gplib.myData;
  class a b;
  model y = a b x1-x20;
  output out=gplib.scores pred=p;
  id account trans_date;
run;
```

SELECTION Statement

SELECTION *<options>* ;

High-performance statistical procedures that support model selection use the SELECTION statement to control details about the model selection process. This statement is supported in different degrees by the HPGENSELECT, HPREG, and HPLOGISTIC procedures. The HPREG procedure supports the most complete set of options.

You can specify the following *options* in the SELECTION statement:

METHOD=NONE | *method*< *method-options* >

specifies the method used to select the model. You can also specify *method-options* that apply to the specified method by enclosing them in parentheses after the *method*. The default selection method (when the METHOD= option is not specified) is METHOD=STEPWISE.

The following *methods* are available and are explained in detail in the section “Methods” on page 546.

NONE	specifies no model selection.
FORWARD	specifies forward selection. This method starts with no effects in the model and adds effects.
BACKWARD	specifies backward elimination. This method starts with all effects in the model and deletes effects.
STEPWISE	specifies stepwise regression. This method is similar to the FORWARD method except that effects already in the model do not necessarily stay there.
FORWARDSWAP	specifies forward-swap selection, which is an extension of the forward selection method. Before any addition step, all pairwise swaps of one effect in the model and one effect out of the current model that improve the selection criterion are made. When the selection criterion is R square, this method is the same as the MAXR method in the REG procedure in SAS/STAT software. The high-performance statistical procedure that supports this method is the HPREG procedure.
LAR	specifies least angle regression. Like forward selection, this method starts by adding effects to an empty model. The parameter estimates at any step are “shrunk” when they are compared to the corresponding least squares estimates. If the model contains classification variables, then these classification variables are split. See the SPLIT option in the CLASS statement for details. The only high-performance statistical procedure that supports this method is the HPREG procedure.
LASSO	adds and deletes parameters by using a version of ordinary least squares in which the sum of the absolute regression coefficients is constrained. If the model contains classification variables, then these classification variables are split. For more information, see the SPLIT option in the CLASS statement.

Table 21.2 lists the applicable *method-options* for each of these methods.

Table 21.2 Applicable *method-options* by *method*

<i>method-option</i>	FORWARD	BACKWARD	STEPWISE	FORWARDSWAP	LAR	LASSO
ADAPTIVE						X
CHOOSE =	X	X	X		X	X
COMPETITIVE			X			
CRITERION =	X	X	X	X		
FAST		X				
LSCOEFFS					X	X
MAXEFFECTS =	X		X	X	X	X
MAXSTEPS =	X	X	X	X	X	X
MINEFFECTS =		X	X			
SELECT =	X	X	X	X		

Table 21.2 *continued*

<i>method-option</i>	FORWARD	BACKWARD	STEPWISE	FORWARDSWAP	LAR	LASSO
SLENTRY =	x		x		x	x
SLSTAY =		x	x			x
STOP =	x	x	x	x	x	x

The syntax of the *method-options* that you can specify in parentheses after the **SELECTION=** option *method* follows. As described in Table 21.2, not all selection *method-options* are applicable to every **METHOD=** *method*.

ADAPTIVE <(GAMMA=*nonnegative number*)>

requests that adaptive weights be applied to each of the coefficients when **METHOD=LASSO**. Ordinary least squares estimates of the model parameters are used to form the adaptive weights. You use the **GAMMA=** option to specify the power transformation that is applied to the parameters in forming the adaptive weights. The default value is **GAMMA=1**.

CHOOSE=*criterion*

chooses from the list of models (at each step of the selection process) the model that yields the best value of the specified criterion. If the optimal value of the specified criterion occurs for models at more than one step, then the model that has the smallest number of parameters is chosen. If you do not specify the **CHOOSE=** option, then the selected model is the model at the final step in the selection process. The criteria that are supported depend on the type of model that is being fit. For the supported criteria, see the chapters for the relevant high-performance statistical procedures.

COMPETITIVE

is applicable only as a *method-option* when **METHOD=STEPWISE** and the **SELECT** criterion is not **SL**. If you specify the **COMPETITIVE** option, then the **SELECT=** criterion is evaluated for all models in which an effect currently in the model is dropped or an effect not yet in the model is added. The effect whose removal from or addition to the model yields the maximum improvement to the **SELECT=** criterion is dropped or added. The **HPREG** procedure is the only high-performance statistical procedure that supports this option.

CRITERION=*criterion*

is an alias for the **SELECT** option.

FAST

implements the computational algorithm of Lawless and Singhal (1978) to compute a first-order approximation to the remaining slope estimates for each subsequent elimination of a variable from the model. When applied in backward selection, this option essentially leads to approximating the selection process as the selection process of a linear regression model in which the crossproducts matrix equals the Hessian matrix in the full model under consideration. The **FAST** option is available only when **METHOD=BACKWARD** in the **HPLOGISTIC** procedure. It is computationally efficient in logistic regression models because the model is not fit after removal of each effect.

LSCOEFFS

requests a hybrid version of the LAR and LASSO methods, in which the sequence of models is determined by the LAR or LASSO algorithm but the coefficients of the parameters for the model at any step are determined by using ordinary least squares.

MAXEFFECTS=*n*

specifies the maximum number of effects in any model that is considered during the selection process. This option is ignored with METHOD=BACKWARD. If at some step of the selection process the model contains the specified maximum number of effects, then no candidates for addition are considered.

MAXSTEPS=*n*

specifies the maximum number of selection steps that are performed. The default value of *n* is the number of effects in the MODEL statement when METHOD=FORWARD, METHOD=BACKWARD, or METHOD=LAR. The default is three times the number of effects when METHOD=STEPWISE or METHOD=LASSO.

MINEFFECTS=*n*

specifies the minimum number of effects in any model that is considered during backward selection. This option is ignored unless METHOD=BACKWARD is specified. The backward selection process terminates if, at some step of the selection process, the model contains the specified minimum number of effects.

SELECT=SL | *criterion*

specifies the criterion that the procedure uses to determine the order in which effects enter or leave at each step of the selection method. The criteria that are supported depend on type of model that is being fit. See the chapter for the relevant high-performance statistical procedure for the supported criteria.

The SELECT option is not valid when METHOD=LAR or METHOD=LASSO. You can use SELECT=SL to request the traditional approach, where effects enter and leave the model based on the significance level. When the value of the SELECT= option is not SL, the effect that is selected to enter or leave at any step of the selection process is the effect whose addition to or removal from the current model yields the maximum improvement in the specified criterion.

SLENTRY=*value***SLE=*value***

specifies the significance level for entry when STOP=SL or SELECT=SL. The default is 0.05.

SLSTAY=*value***SLS=*value***

specifies the significance level for staying in the model when STOP=SL or SELECT=SL. The default is 0.05.

STOP=SL | NONE | *criterion*

specifies a criterion that is used to stop the selection process. The criteria that are supported depend on the type of model that is being fit. For information about the supported criteria, see the chapter about the relevant high-performance statistical procedure.

If you do not specify the STOP= option but do specify the SELECT= option, then the criterion specified in the SELECT= option is also used as the STOP= criterion.

If you specify `STOP=NONE`, then the selection process stops if no suitable add or drop candidates can be found or if a size-based limit is reached. For example, if you specify `STOP=NONE MAXEFFECTS=5`, then the selection process stops at the first step that produces a model with five effects.

When `STOP=SL`, selection stops at the step where the significance level of the candidate for entry is greater than the `SLENTY=` value for addition steps when `METHOD=FORWARD` or `METHOD=STEPWISE` and where the significance level of the candidate for removal is greater than the `SLSTAY=` value when `METHOD=BACKWARD` or `METHOD=STEPWISE`.

If you specify a criterion other than `SL` for the `STOP=` option, then the selection process stops if the selection process produces a local extremum of this criterion or if a size-based limit is reached. For example, if you specify `STOP=AIC MAXSTEPS=5`, then the selection process stops before step 5 if the sequence of models has a local minimum of the AIC criterion before step 5. The determination of whether a local minimum is reached is made on the basis of a stop horizon. The default stop horizon is 3, but you can change it by using the `STOPHORIZON=` option. If the stop horizon is n and the `STOP=` criterion at any step is better than the stop criterion at the next n steps, then the selection process terminates.

DETAILS=NONE | SUMMARY | ALL

DETAILS=STEPS<(CANDIDATES(ALL | n))>

specifies the level of detail to be produced about the selection process. The default is `DETAILS=SUMMARY`.

The `DETAILS=ALL` and `DETAILS=STEPS` options produce the following output:

- tables that provide information about the model that is selected at each step of the selection process.
- entry and removal statistics for inclusion or exclusion candidates at each step. By default, only the top 10 candidates at each step are shown. If you specify `STEPS(CANDIDATES( $n$ ))`, then the best n candidates are shown. If you specify `STEPS(CANDIDATES(ALL))`, then all candidates are shown.
- a selection summary table that shows by step the effect that is added to or removed from the model in addition to the values of the `SELECT`, `STOP`, and `CHOOSE` criteria for the resulting model.
- a stop reason table that describes why the selection process stopped.
- a selection reason table that describes why the selected model was chosen.
- a selected effects table that lists the effects that are in the selected model.

The `DETAILS=SUMMARY` option produces only the selection summary, stop reason, selection reason, and selected effects tables.

HIERARCHY=NONE | SINGLE | SINGLECLASS

specifies whether and how the model hierarchy requirement is applied. You can specify that only classification effects, or both classification and continuous effects, be subject to the hierarchy requirement. The `HIERARCHY=` option is ignored unless you also specify one of the following options: `METHOD=FORWARD`, `METHOD=BACKWARD`, or `METHOD=STEPWISE`.

Model hierarchy refers to the requirement that, for any term to be in the model, all model effects that are contained in the term must be present in the model. For example, in order for the interaction `A*B`

to enter the model, the main effects A and B must be in the model. Likewise, neither effect A nor effect B can leave the model while the interaction A*B is in the model.

You can specify the following values:

- NONE** specifies that model hierarchy not be maintained. Any single effect can enter or leave the model at any given step of the selection process.
- SINGLE** specifies that only one effect enter or leave the model at one time, subject to the model hierarchy requirement. For example, suppose that the model contains the main effects A and B and the interaction A*B. In the first step of the selection process, either A or B can enter the model. In the second step, the other main effect can enter the model. The interaction effect can enter the model only when both main effects have already entered. Also, before A or B can be removed from the model, the A*B interaction must first be removed. All effects (CLASS and interval) are subject to the hierarchy requirement.
- SINGLECLASS** is the same as HIERARCHY=SINGLE except that only CLASS effects are subject to the hierarchy requirement.

The default value is HIERARCHY=NONE.

SCREEN <(*global-screen-options*)> <= *screen-options*>

requests that a subset of the effects specified in the MODEL statement be chosen as candidate effects for model selection. You use the *global-screen-options* and *screen-options* to specify how such a subset is chosen and to control the detail level of the associated output. The SCREEN option is fully documented in the section “[SELECTION Statement](#)” on page 5012 in Chapter 66, “[The HPREG Procedure](#),” which is the only high-performance statistical procedure that supports the SCREEN option.

SELECTION=NONE | BACKWARD | FORWARD | FORWARDSWAP | STEPWISE | LAR | LASSO

is an alias for the [METHOD=](#) option.

STOPHORIZON=*n*

specifies the number of consecutive steps at which the STOP= criterion must worsen in order for a local extremum to be detected. The default value is STOPHORIZON=3. The stop horizon value is ignored if you also specify STOP=NONE or STOP=SL. For example, suppose that STOP=AIC and the sequence of AIC values at steps 1 to 6 of a selection are 10, 7, 4, 6, 5, 2. If STOPHORIZON=2, then the AIC criterion is deemed to have a local minimum at step 3 because the AIC value at the next two steps are greater than the value 4 that occurs at step 3. However, if STOPHORIZON=3, then the value at step 3 is not deemed to be a local minimum because the AIC value at step 6 is lower than the AIC value at step 3.

VAR Statement

VAR *variable-list* ;

Some high-performance statistical procedures (in particular procedures that do not support a MODEL statement) use a VAR statement to identify numerical variables for the analysis.

WEIGHT Statement

WEIGHT *variable* ;

The *variable* in the WEIGHT statement is used as a weight to perform a weighted analysis of the data. Observations with nonpositive or missing weights are not included in the analysis. If a WEIGHT statement is not included, all observations that are used in the analysis are assigned a weight of 1.

Model Selection

Methods

The model selection methods implemented in high-performance statistical procedures are specified in the **METHOD=** option in the **SELECTION** statement. The following methods are available, although specific procedures might support only a subset of these methods. Furthermore, the examples in this section refer to fit criteria that might not be supported by a specific procedure.

Full Model Fitted

When **METHOD=NONE**, the complete model that is specified in the MODEL statement is used to fit the model, and no effect selection is done.

Forward Selection

METHOD=FORWARD specifies the forward selection technique, which begins with just the intercept and then sequentially adds the effect that most improves the fit. The process terminates when no significant improvement can be obtained by adding any effect.

In the traditional implementation of forward selection, the statistic that is used to determine whether to add an effect is the significance level of a hypothesis test that reflects an effect's contribution to the model if it is included. At each step, the effect that is most significant is added. The process stops when the significance level for adding any effect is greater than some specified entry significance level.

An alternative approach to address the critical problem of when to stop the selection process is to assess the quality of the models that are produced by the forward selection method and choose the model from this sequence that “best” balances goodness of fit against model complexity. You can use several criteria for this purpose. These criteria fall into two groups—information criteria and criteria that are based on out-of-sample prediction performance.

You use the **CHOOSE=** option to specify the criterion for selecting one model from the sequence of models produced. If you do not specify a **CHOOSE=** criterion, then the model at the final step is the selected model.

For example, if you specify the following statement, then forward selection terminates at the step where no effect can be added at the 0.2 significance level:

```
selection method=forward(select=SL choose=AIC SLE=0.2);
```

However, the selected model is the first one that has the minimum value of Akaike's information criterion. In some cases, this minimum value might occur at a step much earlier than the final step. In other cases, the AIC might start increasing only if more steps are performed—that is, a larger value is used for the significance level for entry. If you want to minimize AIC, then too many steps are performed in the former case and too few in the latter case. To address this issue, high-performance statistical procedures enable you to specify a stopping criterion by using the **STOP=** option. When you specify a stopping criterion, forward selection continues until a local extremum of the stopping criterion in the sequence of models generated is reached. To be deemed a local extremum, a criterion value at a given step must be better than its value at the next n steps, where n is known as the “stop horizon.” By default, the stop horizon is three steps, but you can change this by specifying the **STOPHORIZON=** option.

For example, if you specify the following statement, then forward selection terminates at the step where the effect to be added at the next step would produce a model that has an AIC statistic larger than the AIC statistic of the current model:

```
selection method=forward(select=SL stop=AIC) stophorizon=1;
```

In most cases, provided that the entry significance level is large enough that the local extremum of the named criterion occurs before the final step, specifying either of the following statements selects the same model, but more steps are done in the first case:

```
selection method=forward(select=SL choose=CRITERION);  
selection method=forward(select=SL stop=CRITERION);
```

In some cases, there might be a better local extremum that cannot be reached if you specify the **STOP=** option but can be found if you use the **CHOOSE=** option. Also, you can use the **CHOOSE=** option in preference to the **STOP=** option if you want to examine how the named criterion behaves as you move beyond the step where the first local minimum of this criterion occurs.

You can specify both the **CHOOSE=** and **STOP=** options. You can also use these options together with options that specify size-based limits on the selected model. You might want to consider models that are generated by forward selection and have at most some fixed number of effects, but select from within this set based on a criterion that you specify. For example, specifying the following statements requests that forward selection continue until there are 20 effects in the final model and chooses among the sequence of models the one that has the largest value of the adjusted R-square statistic:

```
selection method=forward(stop=none maxeffects=20 choose=ADJRSQ);
```

You can also combine these options to select a model where one of two conditions is met. For example, the following statement chooses whatever occurs first between a local minimum of the sum of squares on validation data and a local minimum of the corrected Akaike's information criterion (AICC):

```
selection method=forward(stop=AICC choose=VALIDATE);
```

It is important to keep in mind that forward selection bases the decision about what effect to add at any step by considering models that differ by one effect from the current model. This search paradigm cannot

guarantee reaching a “best” subset model. Furthermore, the add decision is greedy in the sense that the effect that is deemed most significant is the effect that is added. However, if your goal is to find a model that is best in terms of some selection criterion other than the significance level of the entering effect, then even this one step choice might not be optimal. For example, the effect that you would add to get a model that has the smallest value of the Mallows’ $C(p)$ statistic at the next step is not necessarily the same effect that is most significant based on a hypothesis test. High-performance statistical procedures enable you to specify the criterion to optimize at each step by using the **SELECT=** option. For example, the following statement requests that at each step the effect that is added be the one that produces a model that has the smallest value of the Mallows’ $C(p)$ statistic:

```
selection method=forward(select=CP);
```

In the case where all effects are variables (that is, effects with one degree of freedom and no hierarchy), using ADJRSQ, AIC, AICC, BIC, CP, RSQUARE, or SBC as the selection criterion for forward selection produces the same sequence of additions. However, if the degrees of freedom contributed by different effects are not constant or if an out-of-sample prediction-based criterion is used, then different sequences of additions might be obtained.

You can use the **SELECT=** option together with the **CHOOSE=** and **STOP=** options. If you specify only the **SELECT=** criterion, then this criterion is also used as the stopping criterion. In the previous example where only the selection criterion is specified, not only do effects enter based on the Mallows’ $C(p)$ statistic, but the selection terminates when the $C(p)$ statistic has a local minimum.

You can find discussion and references to studies about criteria for variable selection in Burnham and Anderson (2002), along with some cautions and recommendations.

Examples of Forward Selection Specifications

The following statement adds effects that at each step produce the lowest value of the SBC statistic and stops at the step where adding any effect would increase the SBC statistic:

```
selection method=forward stophorizon=1;
```

The following statement adds effects based on significance level and stops when all candidate effects for entry at a step have a significance level greater than the default entry significance level of 0.05:

```
selection=forward(select=SL);
```

The following statement adds effects based on significance level and stops at a step where adding any effect increases the error sum of squares computed on the validation data:

```
selection=forward(select=SL stop=validation) stophorizon=1;
```

The following statement adds effects that at each step produce the lowest value of the AIC statistic and stops at the first step whose AIC value is smaller than the AIC value at the next three steps:

```
selection=forward(select=AIC);
```

The following statement adds effects that at each step produce the largest value of the adjusted R-square statistic and stops at the step where the significance level that corresponds to the addition of this effect is greater than 0.2:


```
selection=forward(select=ADJRSQ stop=SL SLE=0.2);
```

Backward Elimination

METHOD=BACKWARD specifies the backward elimination technique. This technique starts from the full model, which includes all independent effects. Then effects are deleted one by one until a stopping condition is satisfied. At each step, the effect that shows the smallest contribution to the model is deleted.

In the traditional implementation of backward selection, the statistic that is used to determine whether to drop an effect is significance level. At any step, the least significant predictor is dropped and the process continues until all effects that remain in the model are significant at a specified stay significance level (SLS).

Just as with forward selection, you can use the **SELECT=** option to change the criterion that is used to assess effect contributions. You can also specify a stopping criterion in the **STOP=** option and use a **CHOOSE=** option to provide a criterion for selecting among the sequence of models produced. For more information, see the discussion in the section “[Forward Selection](#)” on page 546.

Examples of Backward Selection Specifications

The following statement removes effects that at each step produce the largest value of the Schwarz Bayesian information criterion (SBC) statistic and stops at the step where removing any effect increases the SBC statistic:

```
selection method=backward stophorizon=1;
```

The following statement bases removal of effects on significance level and stops when all candidate effects for removal at a step are significant at the default stay significance level of 0.05:

```
selection method=backward(select=SL);
```

The following statement bases removal of effects on significance level and stops when all effects in the model are significant at the 0.1 level. Finally, from the sequence of models generated, the chosen model is the one that produces the smallest average square error when scored on the validation data:

```
selection method=backward(select=SL choose=validate SLS=0.1);
```

The following statement applies in logistic regression models the fast backward technique of Lawless and Singhal (1978), a first-order approximation that has greater numerical efficiency than full backward selection:

```
selection method=backward(fast);
```

The fast technique fits an initial full logistic model and a reduced model after the candidate effects have been dropped. On the other hand, full backward selection fits a logistic regression model each time an effect is removed from the model.

Stepwise Selection

METHOD=STEPWISE specifies the stepwise method, which is a modification of the forward selection technique that differs in that effects already in the model do not necessarily stay there.

In the traditional implementation of stepwise selection method, the same entry and removal significance levels for the forward selection and backward elimination methods are used to assess contributions of effects as they are added to or removed from a model. If, at a step of the stepwise method, any effect in the model is

not significant at the SLSTAY= level, then the least significant of these effects is removed from the model and the algorithm proceeds to the next step. This ensures that no effect can be added to a model while some effect currently in the model is not deemed significant. Only after all necessary deletions have been accomplished can another effect be added to the model. In this case the effect whose addition is the most significant is added to the model and the algorithm proceeds to the next step. The stepwise process ends when none of the effects outside the model is significant at the SLENTY= level and every effect in the model is significant at the SLSTAY= level. In some cases, neither of these two conditions for stopping is met and the sequence of models cycles. In this case, the stepwise method terminates at the end of the cycle.

Just as you can in forward selection and backward elimination, you can use the **SELECT=** option to change the criterion that is used to assess effect contributions. You can also use the **STOP=** option to specify a stopping criterion and use a **CHOOSE=** option to provide a criterion for selecting among the sequence of models produced. For more information, see the section “**Forward Selection**” on page 546.

For selection criteria other than significance level, the HPREG procedure optionally supports a further modification in the stepwise method. In the standard stepwise method, no effect can enter the model if removing any effect currently in the model would yield an improved value of the selection criterion. In the modification, you can use the **COMPETITIVE** option to specify that addition and deletion of effects should be treated competitively. The selection criterion is evaluated for all models that are produced by deleting an effect from the current model or by adding an effect to this model. The action that most improves the selection criterion is the action taken.

Examples of Stepwise Selection Specifications

The following statement requests stepwise selection based on the SBC criterion:

```
selection method=stepwise;
```

First, if removing any effect yields a model that has a lower SBC statistic than the current model, then the effect that produces the smallest SBC statistic is removed. If removing any effect increases the SBC statistic, then provided that adding some effect lowers the SBC statistic, the effect that produces the model that has the lowest SBC is added.

The following statement requests the traditional stepwise method:

```
selection=stepwise(select=SL)
```

First, if the removal of any effect in the model is not significant at the default stay level of 0.05, then the least significant effect is removed and the algorithm proceeds to the next step. Otherwise, the effect whose addition is the most significant is added, provided that it is significant at the default entry level of 0.05.

The following statement requests the traditional stepwise method, where effects enter and leave based on significance levels, but with the following extra check: if any effect to be added or removed yields a model whose SBC statistic is greater than the SBC statistic of the current model, then the stepwise method terminates at the current model.

```
selection method=stepwise(select=SL stop=SBC) stophorizon=1;
```

In this case, the entry and stay significance levels still play a role because they determine whether an effect is deleted from or added to the model. This extra check might result in the selection terminating before a local minimum of the SBC criterion is found.

The following statement selects effects to enter or drop as in the previous example except that the significance level for entry is now 0.1 and the significance level to stay is 0.08. From the sequence of models produced,

the selected model is chosen to yield the minimum AIC statistic:

```
selection method=stepwise(select=SL SLE=0.1 SLS=0.08 choose=AIC);
```

The following statement requests stepwise selection that is based on the AICC criterion and treats additions and deletions competitively:

```
selection method=stepwise(select=AICC competitive);
```

Each step evaluates the AICC statistics that correspond to the removal of any effect in the current model or the addition of any effect to the current model and chooses the addition or removal that produced the minimum value, provided that this minimum is lower than the AICC statistic of the current model.

The following statement requests stepwise selection that is based on the SBC criterion, treats additions and deletions competitively, and stops based on the average square error over the validation data:

```
selection=stepwise(select=SBC competitive stop=VALIDATE);
```

At any step, SBC statistics that correspond to the removal of any effect from the current model or the addition of any effect to the current model are evaluated. The addition or removal that produces the minimum SBC value is made. The average square error on the validation data for the model with this addition or removal is evaluated. The selection stops when the average square error so produced increases for three consecutive steps.

Forward-Swap Selection

METHOD=FORWARDSSWAP specifies the forward-swap selection method, which is an extension of the forward selection method. The forward-swap selection method incorporates steps that improve a model by replacing an effect in the model with an effect that is not in the model. When the model selection criterion is R square, this method is the same as the maximum R-square improvement (MAXR) method that is implemented in the REG procedure in SAS/STAT software. You cannot use the effect significance level as the selection criterion for the forward-swap method.

The forward-swap selection method begins by finding the one-effect model that produces the best value of the selection criterion. Then another effect (the one that yields the greatest improvement in the selection criterion) is added. After the two-effect model is obtained, each of the effects in the model is compared to each effect that is not in the model. For each comparison, the forward-swap method determines whether removing one effect and replacing it with the other effect improves the selection criterion. After comparing all possible swaps, the forward-swap method makes the swap that produces the greatest improvement in the selection criterion. Comparisons begin again, and the process continues until the forward-swap method finds that no other swap could improve the selection criterion. Thus, the two-variable model that is produced is considered the “best” two-variable model that the technique can find. Another variable is then added to the model, and the comparing-and-swapping process is repeated to find the “best” three-variable model, and so on.

The difference between the stepwise selection method and the forward-swap selection method is that all swaps are evaluated before any addition is made in the forward-swap method. In the stepwise selection method, the “worst” effect might be removed without considering what adding the “best” remaining effects might accomplish. Because the forward-swap method needs to examine all possible pairwise effect swaps at each step of the selection process, the forward-swap method is much more computationally expensive than the stepwise selection method; it might not be appropriate for models that contain a large number of effects.

Least Angle Regression

METHOD=LAR specifies least angle regression (LAR), which is supported in the HPREG procedure. LAR was introduced by Efron et al. (2004). Not only does this algorithm provide a selection method in its own right, but with one additional modification, it can be used to efficiently produce LASSO solutions. Just like the forward selection method, the LAR algorithm produces a sequence of regression models in which one parameter is added at each step, terminating at the full least squares solution when all parameters have entered the model.

The algorithm starts by centering the covariates and response and scaling the covariates so that they all have the same corrected sum of squares. Initially all coefficients are zero, as is the predicted response. The predictor that is most correlated with the current residual is determined, and a step is taken in the direction of this predictor. The length of this step determines the coefficient of this predictor and is chosen so that some other predictor and the current predicted response have the same correlation with the current residual. At this point, the predicted response moves in the direction that is equiangular between these two predictors. Moving in this direction ensures that these two predictors continue to have a common correlation with the current residual. The predicted response moves in this direction until a third predictor has the same correlation with the current residual as the two predictors already in the model. A new direction is determined that is equiangular among these three predictors, and the predicted response moves in this direction until a fourth predictor, which has the same correlation with the current residual, joins the set. This process continues until all predictors are in the model.

As in other selection methods, the issue of when to stop the selection process is crucial. You can use the CHOOSE= option to specify a criterion for choosing among the models at each step. You can also use the STOP= option to specify a stopping criterion. These formulas use the approximation that at step k of the LAR algorithm, the model has k degrees of freedom. See Efron et al. (2004) for a detailed discussion of this so-called simple approximation.

A modification of LAR selection that is suggested in Efron et al. (2004) uses the LAR algorithm to select the set of covariates in the model at any step, but it uses ordinary least squares regression with just these covariates to obtain the regression coefficients. You can request this hybrid method by specifying the LSCOEFFS suboption of METHOD=LAR.

Lasso Selection

Method=LASSO specifies the least absolute shrinkage and selection operator (LASSO) method, which is supported in the HPREG procedure. LASSO arises from a constrained form of ordinary least squares regression where the sum of the absolute values of the regression coefficients is constrained to be smaller than a specified parameter. More precisely let $X = (x_1, x_2, \dots, x_m)$ denote the matrix of covariates and let y denote the response, where the x_i s have been centered and scaled to have unit standard deviation and mean zero and y has mean zero. Then for a given parameter t , the LASSO regression coefficients $\beta = (\beta_1, \beta_2, \dots, \beta_m)$ are the solution to the following constrained optimization problem:

$$\text{minimize } ||y - X\beta||^2 \quad \text{subject to} \quad \sum_{j=1}^m |\beta_j| \leq t$$

Provided that the LASSO parameter t is small enough, some of the regression coefficients are exactly 0. Hence, you can view the LASSO as selecting a subset of the regression coefficients for each LASSO

parameter. By increasing the LASSO parameter in discrete steps, you obtain a sequence of regression coefficients in which the nonzero coefficients at each step correspond to selected parameters.

Early implementations (Tibshirani 1996) of LASSO selection used quadratic programming techniques to solve the constrained least squares problem for each LASSO parameter of interest. Later Osborne, Presnell, and Turlach (2000) developed a “homotopy method” that generates the LASSO solutions for all values of t . Efron et al. (2004) derived a variant of their algorithm for least angle regression that can be used to obtain a sequence of LASSO solutions from which all other LASSO solutions can be obtained by linear interpolation. This algorithm for **METHOD=LASSO** is used in PROC HPREG. It can be viewed as a stepwise procedure with a single addition to or deletion from the set of nonzero regression coefficients at any step.

As in the other selection methods that are supported by high-performance statistical procedures, you can use the **CHOOSE=** option to specify a criterion to choose among the models at each step of the LASSO algorithm. You can also use the **STOP=** option to specify a stopping criterion. For more information, see the discussion in the section “**Forward Selection**” on page 546. The model degrees of freedom that PROC GLMSELECT uses at any step of the LASSO are simply the number of nonzero regression coefficients in the model at that step. Efron et al. (2004) cite empirical evidence for doing this but do not give any mathematical justification for this choice.

A modification of LASSO selection suggested in Efron et al. (2004) uses the LASSO algorithm to select the set of covariates in the model at any step, but it uses ordinary least squares regression and just these covariates to obtain the regression coefficients. You can request this hybrid method by specifying the **LSCOEFFS** suboption of **SELECTION=LASSO**.

Adaptive Lasso Selection

Adaptive lasso selection is a modification of lasso selection; in adaptive lasso selection, weights are applied to each of the parameters in forming the lasso constraint (Zou 2006). More precisely, suppose that the response y has mean 0 and the regressors x are scaled to have mean 0 and common standard deviation. Furthermore, suppose that you can find a suitable estimator $\hat{\beta}$ of the parameters in the true model and you define a weight vector by $w = 1/|\hat{\beta}|^\gamma$, where $\gamma \geq 0$. Then the adaptive lasso regression coefficients $\beta = (\beta_1, \beta_2, \dots, \beta_m)$ are the solution to the following constrained optimization problem:

$$\text{minimize } ||y - \mathbf{X}\beta||^2 \quad \text{subject to} \quad \sum_{j=1}^m |w_j \beta_j| \leq t$$

PROC HPREG uses the solution to the unconstrained least squares problem as the estimator $\hat{\beta}$. This is appropriate unless collinearity is a concern. If the regressors are collinear or nearly collinear, then Zou (2006) suggests using a ridge regression estimate to form the adaptive weights.

References

Burnham, K. P., and Anderson, D. R. (2002). *Model Selection and Multimodel Inference*. 2nd ed. New York: Springer-Verlag.

- Efron, B., Hastie, T. J., Johnstone, I. M., and Tibshirani, R. (2004). “Least Angle Regression.” *Annals of Statistics* 32:407–499. With discussion.
- Lawless, J. F., and Singhal, K. (1978). “Efficient Screening of Nonnormal Regression Models.” *Biometrics* 34:318–327.
- Osborne, M. R., Presnell, B., and Turlach, B. A. (2000). “A New Approach to Variable Selection in Least Squares Problems.” *IMA Journal of Numerical Analysis* 20:389–404.
- Tibshirani, R. (1996). “Regression Shrinkage and Selection via the Lasso.” *Journal of the Royal Statistical Society, Series B* 58:267–288.
- Zou, H. (2006). “The Adaptive Lasso and Its Oracle Properties.” *Journal of the American Statistical Association* 101:1418–1429.

Subject Index

- adaptive lasso selection
 - high-performance statistical procedures,
SELECTION statement, [553](#)
- backward elimination
 - high-performance statistical procedures,
SELECTION statement, [549](#)
- forward selection
 - high-performance statistical procedures,
SELECTION statement, [546](#)
- forward swap selection
 - high-performance statistical procedures,
SELECTION statement, [551](#)
- frequency variable
 - high-performance statistical procedures, [540](#)
- high-performance statistical procedures, SELECTION
statement
 - adaptive lasso selection, [553](#)
 - backward elimination, [549](#)
 - forward selection, [546](#)
 - forward swap selection, [551](#)
 - LAR selection, [552](#)
 - LASSO selection, [552](#)
 - stepwise selection, [549](#)
- LAR selection
 - high-performance statistical procedures,
SELECTION statement, [552](#)
- LASSO selection
 - high-performance statistical procedures,
SELECTION statement, [552](#)
- stepwise selection
 - high-performance statistical procedures,
SELECTION statement, [549](#)

Syntax Index

ADAPTIVE option
 SELECTION statement (high-performance statistical procedures), [542](#)

CHOOSE= option
 MODEL statement (high-performance statistical procedures), [542](#)

CLASS statement
 high-performance statistical procedures, [536](#)

COMPETITIVE option
 SELECTION statement (high-performance statistical procedures), [542](#)

CRITERION= option
 SELECTION statement (high-performance statistical procedures), [542](#)

DESCENDING option
 CLASS statement (high-performance statistical procedures), [536](#)

DETAILS= option
 SELECTION statement (high-performance statistical procedures), [544](#)

FAST option
 SELECTION statement (high-performance statistical procedures), [542](#)

FREQ statement
 high-performance statistical procedures, [540](#)

HIERARCHY= option
 SELECTION statement (high-performance statistical procedures), [544](#)

high-performance statistical procedures
 FREQ statement, [540](#)
 ID statement, [540](#)
 VAR statement, [546](#)
 WEIGHT statement, [546](#)

high-performance statistical procedures, CLASS statement, [536](#)
 DESCENDING option, [536](#)
 MISSING option, [539](#)
 ORDER= option, [537](#)
 PARAM= option, [539](#)
 REF= option, [537](#)
 TRUNCATE= option, [539](#)

high-performance statistical procedures, FREQ statement, [540](#)

high-performance statistical procedures, ID statement, [540](#)

high-performance statistical procedures, SELECTION statement, [540](#)
 ADAPTIVE option, [542](#)
 CHOOSE= option, [542](#)
 COMPETITIVE option, [542](#)
 CRITERION= option, [542](#)
 DETAILS= option, [544](#)
 HIERARCHY= option, [544](#)
 LSCOEFFS option, [543](#)
 MAXEFFECTS= option, [543](#)
 MAXSTEP= option, [543](#)
 METHOD= option, [541](#)
 MINEFFECTS= option, [543](#)
 SCREEN= option, [545](#)
 SELECT= option, [543](#)
 SELECTION= option, [545](#)
 SLE= option, [543](#)
 SLENTY= option, [543](#)
 SLS= option, [543](#)
 SLSTAY= option, [543](#)
 STOP= option, [543](#)
 STOPHORIZON= option, [545](#)

high-performance statistical procedures, VAR statement, [546](#)

high-performance statistical procedures, WEIGHT statement, [546](#)

HPREG procedure, CLASS statement
 SPLIT option, [538](#)

ID statement
 high-performance statistical procedures, [540](#)

LSCOEFFS option
 SELECTION statement (high-performance statistical procedures), [543](#)

MAXEFFECTS= option
 SELECTION statement (high-performance statistical procedures), [543](#)

MAXSTEPS= option
 SELECTION statement (high-performance statistical procedures), [543](#)

METHOD= option
 SELECTION statement (high-performance statistical procedures), [541](#)

MINEFFECTS= option
 SELECTION statement (high-performance statistical procedures), [543](#)

MISSING option

CLASS statement (high-performance statistical procedures), [539](#)

ORDER= option

CLASS statement (high-performance statistical procedures), [537](#)

CLASS statement (HPREG), [538](#)

PARAM= option

CLASS statement (high-performance statistical procedures), [539](#)

REF= option

CLASS statement (high-performance statistical procedures), [537](#)

SCREEN= option

SELECTION statement (high-performance statistical procedures), [545](#)

SELECT= option

SELECTION statement (high-performance statistical procedures), [543](#)

SELECTION statement

high-performance statistical procedures, [540](#)

SELECTION= option

SELECTION statement (high-performance statistical procedures), [545](#)

SLE= option

SELECTION statement (high-performance statistical procedures), [543](#)

SLENTY= option

SELECTION statement (high-performance statistical procedures), [543](#)

SLS= option

SELECTION statement (high-performance statistical procedures), [543](#)

SLSTAY= option

SELECTION statement (high-performance statistical procedures), [543](#)

STOP= option

SELECTION statement (high-performance statistical procedures), [543](#)

STOPHORIZON= option

SELECTION statement (high-performance statistical procedures), [545](#)

TRUNCATE= option

CLASS statement (high-performance statistical procedures), [539](#)

VAR statement

high-performance statistical procedures, [546](#)

WEIGHT statement

high-performance statistical procedures, [546](#)