The GLMSELECT Procedure (Experimental)

Contents

The GLMSELECT Procedure (Experimental)

Overview

The GLMSELECT procedure performs effect selection in the framework of general linear models. A variety of model selection methods are available, including the LASSO method of Tibshirani (1996) and the related LAR method of Efron et. al (2004). The procedure offers extensive capabilities for customizing the selection with a wide variety of selection and stopping criteria, from traditional and computationally efficient significance-level-based criteria to more computationally intensive validation-based criteria. The procedure also provides graphical summaries of the selection search.

The GLMSELECT compares most closely to REG and GLM. The REG procedure supports a variety of model-selection methods but does not support a CLASS statement. The GLM procedure supports a CLASS statement but does not include effect selection methods. The GLMSELECT procedure fills this gap. GLMSELECT focuses on the standard independently and identically distributed general linear model for univariate responses and offers great flexibility for and insight into the model selection algorithm. GLMSELECT provides results (displayed tables, output data sets, and macro variables) that make it easy to take the model it selects and explore it in more detail in a subsequent procedure such as REG or GLM.

Features

The main features of the GLMSELECT procedure are as follows:

- multiple effect selection methods
- effect selection based on a variety of selection criteria
- stopping rules based on a variety of model evaluation criteria
- leave-one-out cross validation
- k-fold cross validation
- different parameterizations for classification effects
- specify any degree of interaction (crossed effects) and nested effects
- supports hierarchy among effects
- provides for internal partioning of data into training, validation, and testing roles
- produces graphical representation of selection process
- produces output data sets containing predicted values and residuals
- produces macro variables containing selected models

- supports parallel processing of BY groups
- enables selection from a very large number of effects (tens of thousands)
- selection of individual levels of classification effects
- multiple SCORE statements

The GLMSELECT PROCEDURE supports the following effect selection methods. These methods are explained in detail in the "Model-Selection Methods" section on page 35.

FORWARD	Forward selection. This method starts with no effects in the model and adds effects.
BACKWARD	Backward elimination. This method starts with all effects in the model and deletes effects.
STEPWISE	Stepwise regression. This is similar to the FORWARD method except that effects already in the model do not necessarily stay there.
LAR	Least angle regression. This method, like forward selection, starts with no effects in the model and adds effects. The parameter estimates at any step are "shrunk" when compared to the corresponding least-squares estimates.
LASSO	This method adds and deletes parameters based on a version of ordinary least squares where the sum of the absolute regression coefficients is constrained.

Hybrid versions of LAR and LASSO are also supported, which use the respective method to select the model but then estimate the regression coefficients by ordinary weighted least squares.

The GLMSELECT procedure is intended primarily as a model selection procedure and does not include regression diagnostics or other post-selection facilities such as hypothesis testing, contrasts and LS-means analyses. The intention is that you use PROC GLMSELECT to select a model or a set of candidate models. Further investigation of these models can be done by using these models in existing regression procedures.

Getting Started

The following data set contains salary and performance information for Major League Baseball players who played at least one game in both the 1986 and 1987 seasons, excluding pitchers. The salaries (*Sports Illustrated,* April 20, 1987) are for the 1987 season and the performance measures are from 1986 (Collier Books, *The 1987 Baseball Encyclopedia Update*).

```
data baseball;
  length name $ 18;
  length team $ 12;
```

```
   input name $ 1-18 nAtBat nHits nHome nRuns nRBI nBB
         yrMajor crAtBat crHits crHome crRuns crRbi crBB
         league $ division $ team $ position $ nOuts nAssts
         nError salary;
   label name="Player's Name"
      nAtBat="Times at Bat in 1986"
      nHits="Hits in 1986"
      nHome="Home Runs in 1986"
      nRuns="Runs in 1986"
      nRBI="RBIs in 1986"
      nBB="Walks in 1986"
      yrMajor="Years in the Major Leagues"
      crAtBat="Career times at bat"
      crHits="Career Hits"
      crHome="Career Home Runs"
      crRuns="Career Runs"
      crRbi="Career RBIs"
      crBB="Career Walks"
      league="League at the end of 1986"
      division="Division at the end of 1986"
      team="Team at the end of 1986"
      position="Position(s) in 1986"
      nOuts="Put Outs in 1986"
      nAssts="Assists in 1986"
      nError="Errors in 1986"
      salary="1987 Salary in $ Thousands";
      logSalary = log(Salary);
   datalines;
Allanson, Andy       293    66     1    30    29    14
                       1   293    66     1    30    29    14
                    American East Cleveland C 446 33 20 .
Ashby, Alan          315    81     7    24    38    39
                      14  3449   835    69   321   414   375
                    National West Houston C 632 43 10 475


... more datalines ...

Randolph, Willie     492   136     5    76    50    94
                      12  5511  1511    39   897   451   875
                    American East NewYork 2B 313 381 20 875
Tolleson, Wayne      475   126     3    61    43    52
                       6  1700   433     7   217    93   146
                    American West Chicago 3B 37 113 7 385
Upshaw, Willie       573   144     9    85    60    78
                       8  3198   857    97   470   420   332
                    American East Toronto 1B 1314 131 12 960
Wilson, Willie       631   170     9    77    44    31
                      11  4908  1457    30   775   357   249
                    American West KansasCity CF 408 4 3 1000
;
```

Suppose you want to investigate whether you can model the players salaries for the 1987 season based on performance measures for the previous season. The aim is to obtain a parsimonious model that does not overfit this particular data, making it useful for prediction. This example shows how you can use PROC GLMSELECT as a starting point for such an analysis. Since the variation of salaries is much greater

for the higher salaries, it is appropriate to apply a log transformation to the salaries before doing the model selection.

The following code selects a model using the default settings:

```
ods html;
ods graphics on;

proc glmselect data=baseball plots=all;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
                  / details=all stats=all;
run;

ods graphics off;
ods html close;
```

PROC GLMSELECT performs effect selection where effects can contain classification variables that you specify on a CLASS statement. The "Class Level Information" table shown in Figure 1 lists the levels of the classification variables "division" and "league."

The GLMSELECT Procedure

Class Level Information

Class	Levels	Values
league	2	American National
division	2	East West

Figure 1. Class Level Information

When you specify effects that contain classification variables the number of parameters is usually larger than the number of effects. The "Dimensions" table in Figure 2 shows the number of effects and the number of parameters considered.

Dimensions

Number of Effects	19
Number of Parameters	21

Figure 2. Dimensions

Data Set	WORK.BASEBALL
Dependent Variable	logSalary
Selection Method	Stepwise
Select Criterion	SBC
Stop Criterion	SBC
Effect Hierarchy Enforced	None

Figure 3. Model Information

You find details of the default search settings in the "Model Information" table shown in Figure 3. The default selection method is a variant of the traditional stepwise selection where the decisions about what effects to add or drop at any step and when to terminate the selection are both based on the Schwarz Bayesian Information Criterion (SBC). The effect in the current model whose removal yields the maximal decrease in the SBC statistic is dropped provided this lowers the SBC value. Once no decrease in the SBC value can be obtained by dropping an effect in the model, the effect whose addition to the model yields the lowest SBC statistic is added and the whole process is repeated. The method terminates when dropping or adding any effect increases the SBC statistic.

Stepwise Selection: Step 2
Effect Entered: nHits

Best 10 Entry Candidates

Rank	Effect	SBC
1	nHits	-252.5794
2	nAtBat	-241.5789
3	nRuns	-240.1010
4	nRBI	-232.2880
5	nBB	-223.3741
6	nHome	-208.0565
7	nOuts	-205.8107
8	division	-194.4688
9	crBB	-191.5141
10	nAssts	-190.9425

Figure 4. Candidates For entry at Step Two

The DETAILS=ALL option requests details of each step of the selection process. The "Candidates" table at each step shows the candidates for inclusion or removal at that step ranked from best to worst in terms of the selection criterion, which in this example is the SBC statistic. By default only the 10 best candidates are shown. Figure 4 shows the "Candidates" table at step two.

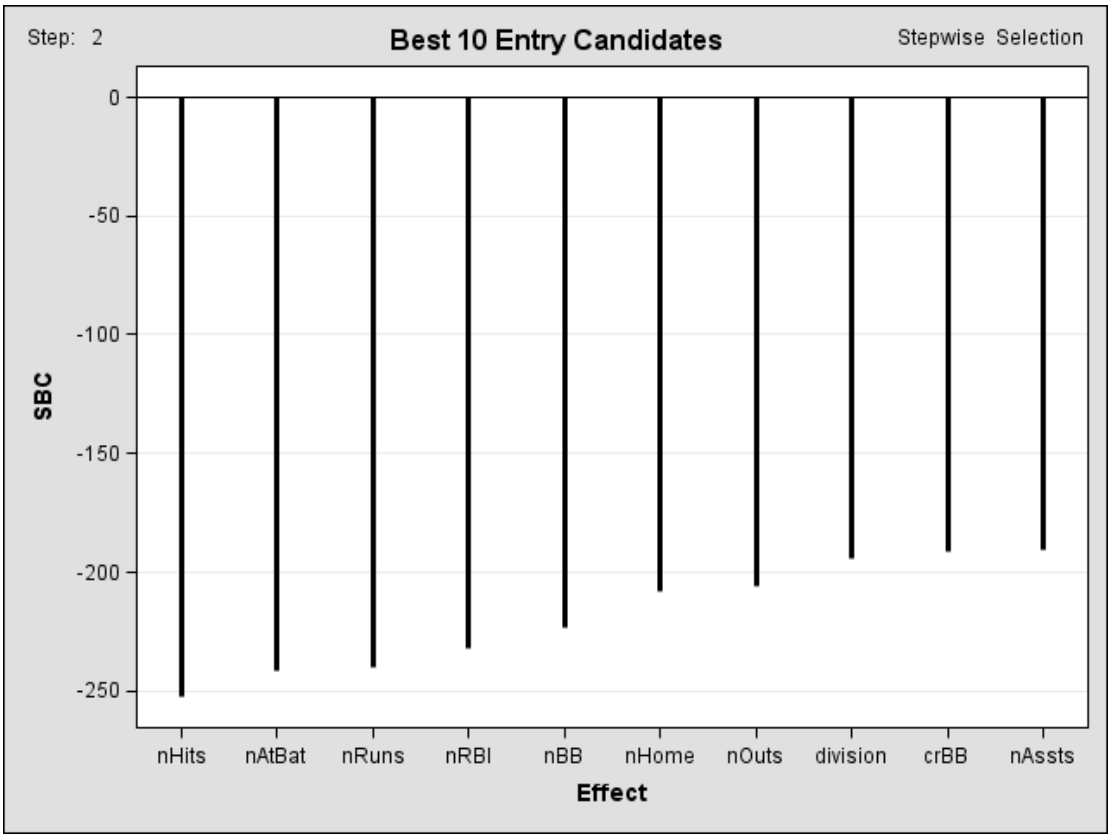


Figure 5. Needle Plot of Entry Candidates at Step Two

To help in the interpretation of the selection process, you can use graphics supported by PROC GLMSELECT. You enable these graphical displays by specifying the ODS GRAPHICS statement. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*). With ODS graphics enabled, the PLOTS=ALL option together with the DETAILS=STEPS option on the MODEL statement produces a needle plot view of the "Candidates" tables. The plot corresponding to the "Candidates" table at step two is shown in Figure 5. You can see that adding the effect "nHits" yields the smallest SBC value and so this effect is added at step two.

Stepwise Selection Summary

Step	Effect Entered	Effect Removed	Number Effects In	Number Parms In	Model R-Square	Adjusted R-Square
0	Intercept		1	1	0.0000	0.0000
1	crRuns		2	2	0.4187	0.4165
2	nHits		3	3	0.5440	0.5405
3	yrMajor		4	4	0.5705	0.5655
4		crRuns	3	3	0.5614	0.5581
5	nBB		4	4	0.5818	0.5770*

* Optimal Value Of Criterion

Stepwise Selection Summary

Step	Effect Entered	Effect Removed	AIC	AICC	BIC
0	Intercept		-60.7762	0.7767	-60.6397
1	crRuns		-201.4609	0.2419	-200.7872
2	nHits		-263.2959	0.0071	-261.8807
3	yrMajor		-277.0208	-0.0448	-275.3333
4		crRuns	-273.5517	-0.0319	-271.9095
5	nBB		-284.0690*	-0.0716*	-282.1700*

* Optimal Value Of Criterion

Stepwise Selection Summary

Step	Effect Entered	Effect Removed	CP	SBC	PRESS
0	Intercept		375.9275	-57.2041	208.7381
1	crRuns		111.2315	-194.3166	123.9195
2	nHits		33.4438	-252.5794	97.6368
3	yrMajor		18.5870	-262.7322	92.2998
4		crRuns	22.3357	-262.8353	93.1482
5	nBB		11.3524*	-269.7804*	89.5434*

* Optimal Value Of Criterion

Stepwise Selection Summary

Step	Effect Entered	Effect Removed	ASE	F Value	Pr > F
0	Intercept		0.7877	0.00	1.0000
1	crRuns		0.4578	188.01	<.0001
2	nHits		0.3592	71.42	<.0001
3	yrMajor		0.3383	15.96	<.0001
4		crRuns	0.3454	5.44	0.0204
5	nBB		0.3294	12.62	0.0005

* Optimal Value Of Criterion

Figure 6. Selection Summary Table

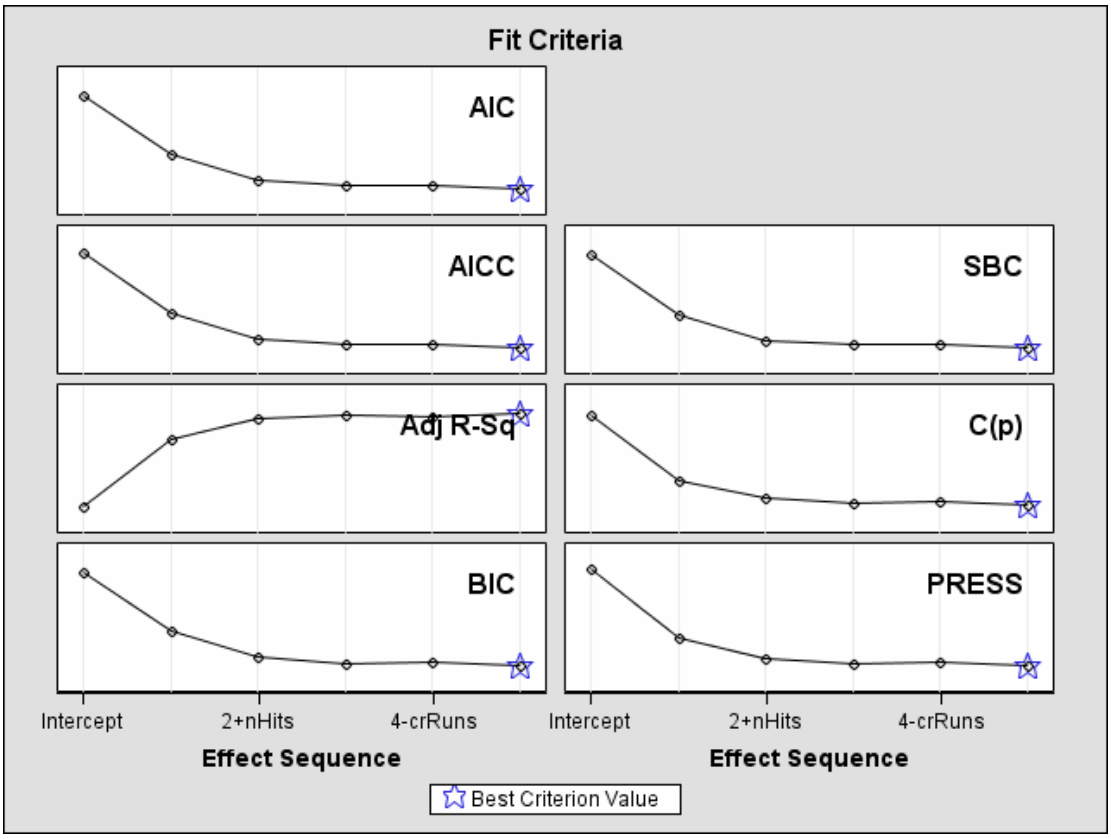


Figure 7. Criterion Panel

The "Selection Summary" table in Figure 6 shows the effect that was added or dropped at each step of the selection process together with fit statistics for the model at each step. The STATS=ALL option on the MODEL statement requests that all the available fit statistics are displayed. See the "Criteria Used in Model Selection Methods" section on page 44 for descriptions and formulae. The "Criterion Panel" plot in Figure 7 provides a graphical view of the evolution of these fit criteria as the selection process progresses. Note that none of these criteria has a local minimum before step five.

Selection stopped at a local minimum of the SBC criterion.

Stop Details

Candidate For	Effect	Candidate SBC		Compare SBC
Entry	division	-268.6094	>	-269.7804
Removal	nBB	-262.8353	>	-269.7804

Figure 8. Stopping Details

The "Stop Reason" and "Stop Details" tables in Figure 8 gives details of why the selection process terminated. This table shows at step five the best add candidate, "division," and the best drop candidate, "nBB," yield models with SBC values of -268.6094 and -262.83 respectively. Both of these values are larger the current SBC value of -269.7804 and so the selection process stops at the model at step five.

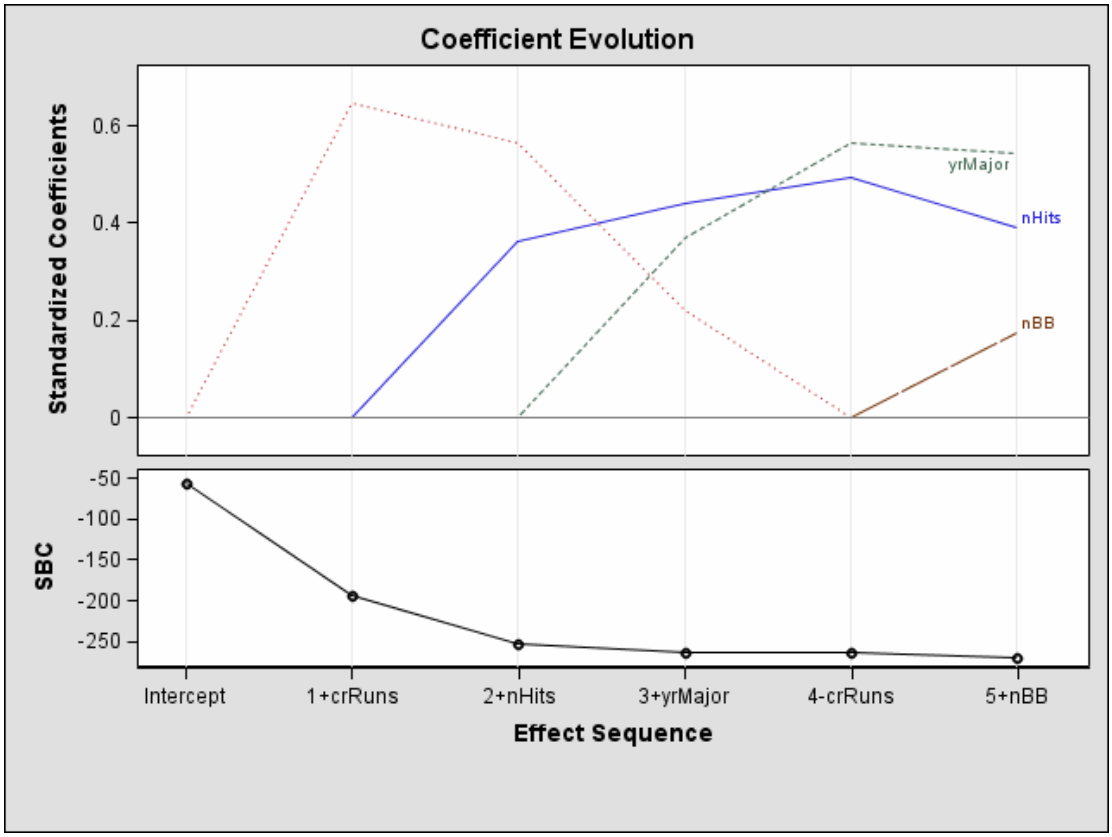


Figure 9. Coefficient Evolution

The "Coefficient Panel" in Figure 9 enables you to visualize the selection process. In this plot, standardized coefficients of all the effects selected at some step of the stepwise method are plotted as a function of the step number. This enables you to assess the relative importance of the effects selected at any step of the selection process as well as providing information as to when effects entered the model. The lower plot in the panel shows how the criterion used to choose the selected model changes as effects enter or leave the model.

The "Selected Effects," "ANOVA," "Fit Statistics," and "Parameter Estimates" tables shown in Figure 10 give details of the selected model.

Selected Model

The selected model is the model at the last step (Step 5).

Effects: Intercept nHits nBB yrMajor

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	3	120.52553	40.17518	120.12
Error	259	86.62820	0.33447	
Corrected Total	262	207.15373		

Root MSE	0.57834
Dependent Mean	5.92722
R-Square	0.5818
Adj R-Sq	0.5770
AIC	-284.06903
AICC	-0.07162
BIC	-282.17004
C(p)	11.35235
PRESS	89.54336
SBC	-269.78041
ASE	0.32938

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value
Intercept	1	4.013911	0.111290	36.07
nHits	1	0.007929	0.000994	7.98
nBB	1	0.007280	0.002049	3.55
yrMajor	1	0.100663	0.007551	13.33

Figure 10. Details of the Selected Model

PROC GLMSELECT provides you with the flexibility to use several selection methods and many fit criteria for selecting effects that enter or leave the model. You can also specify criteria to determine when to stop the selection process and to choose among the models at each step of the selection process. You can find continued exploration of the baseball data using a variety of these methods in Example 1.

Syntax

The following statements are available in PROC GLMSELECT:

```
PROC GLMSELECT < options >;
   BY variables ;
   CLASS variable <(v-options)> <variable <(v-options)>... >
         < / v-options > < options >;
   FREQ variable ;
   MODEL variable = < effects >< / options >;
   OUTPUT < OUT=SAS-data-set > < keyword<=name> >
      < ... keyword<=name> > ;
   PARTITION < options >;
   PERFORMANCE < options >;
   SCORE < DATA=SAS-data-set > < OUT=SAS-data-set > > ;
   WEIGHT variable ;
```

The PROC GLMSELECT statement invokes the procedure. All statements other than the MODEL statement are optional and multiple SCORE statements can be used. The CLASS statement, if present, must precede the MODEL statement.

PROC GLMSELECT Statement

PROC GLMSELECT < *options* > ;

Table 1. PROC GLMSELECT Statement Options

Option	Description
Data Set Options	
DATA=	names a data set to use for the regression
MAXMACRO=	sets the maximum number of macro variables produced
TESTDATA=	names a data set containing test data
VALDATA=	names a data set containing validation data
ODS Graphics Options	
PLOTS=	produces ODS graphics
Other Options	
NAMELEN=	sets the length of effect names in tables and output data sets
NOPRINT	suppresses displayed output including plots
SEED=	sets the seed used for pseudo-random number generation

Following are explanations of the options that you can specify in the PROC GLMSELECT statement (in alphabetical order).

DATA=*SAS-data-set*

names the SAS data set to be used by PROC GLMSELECT. If the DATA= option is not specified, PROC GLMSELECT uses the most recently created SAS data set. If the named data set contains a variable named _ROLE_ then this variable is to assign observations for training, validation, and testing roles. See the "Using Validation and Test Data" section on page 54 for details on using an _ROLE_ variable.

MAXMACRO=*n*

specifies the maximum number of macro variables with selected effects to create. By default, MAXMACRO=100. PROC GLMSELECT saves the list of selected effects in a macro variable, &_GLSIND. Say your input effect list consists of x1-x10, then &_GLSIND would be set to x1 x3 x4 x10 if for example the first, third, fourth, and tenth effects were selected for the model. This list can be used, for example, in the model statement of a subsequent procedure.

With BY processing, one macro variable is created for each BY group, and they are indexed by the BY group number. The MAXMACRO= option can be used to either limit or increase the number of these macro variables when processing data sets with many BY groups.

With no BY processing, PROC GLMSELECT creates:

_GLSIND	selected effects
_GLSIND1	selected effects
_GLSNUMBYS	number of BY groups (1)
_GLSNUMMACROBYS	number of _GLSINDi macro variables actually made (1)

With BY processing, PROC GLMSELECT creates:

_GLSIND	selected effects for BY group 1
_GLSIND1	selected effects for BY group 1
_GLSIND2	selected effects for BY group 2
.	
.	
.	
_GLSINDm	selected effects for BY group m where a number is substituted for m
_GLSNUMBYS	n, the number of BY groups
_GLSNUMMACROBYS	the number m of _GLSINDi macro variables actually made. This value can be less than _GLSNUMBYS = n, and it is less than or equal to the MAXMACRO= value.

See the "Macro Variables Containing Selected Models" section on page 50 for further details.

NOPRINT

suppresses all displayed output including plots.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be *n* characters long, where *n* is a value between 20 and 200 characters. The default length is 20 characters.

PLOTS<**(***global-options***)**> <**=** *specific-plot*<**(***specific-plot-options***)**>>
PLOTS<**(***global-options***)**><**= (** *specific-plot*<**(***specific-plot-options***)**>
... *specific-plot*<**(***specific-plot-options***)**>**)**>

requests that the GLMSELECT procedure produces statistical graphics via the Output Delivery System, provided that the ODS GRAPHICS statement has been specified. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*).

The *global-options* apply to all plots generated by the GLMSELECT procedure, unless it is altered by a *specific-plot-option*. The following global plot options are available:

MAXSTEPLABEL= *n*

specifies the maximum number of characters beyond which labels of effects on plots are truncated.

MAXPARMLABEL= *n*

specifies the maximum number of characters beyond which parameter labels on plots are truncated.

STEPAXIS= EFFECT | NORMBETA | NUMBER

specifies the the horizontal axis to be used on the plots where this axis represents the sequence of entering or departing effects.

STEPAXIS=EFFECT

requests that each step be labeled by the effect that enters or leaves at that step.

STEPAXIS=NORMBETA

is valid only with LAR and LASSO selection methods and requests that the horizontal axis value at step i is the L1 norm of the parameters at step i, normalized by the L1 norm of the parameters at the final step.

STEPAXIS=NUMBER

requests that each step be labeled by the step number.

UNPACK | UNPACKPANELS

breaks a plot that is otherwise paneled into individual component plots.

The following specific plots are available:

ALL

requests that all default plots are produced. Note that candidate plots are produced only if you request DETAILS=STEPS or DETAILS=ALL on the MODEL statement.

ASE | ASEPLOT <(*aseplot-options*)>

plots the evolution average square error on the training data, and the test and validation data whenever these data are provided with the TESTDATA= and VALDATA= options or are produced using a PARTITION statement. The following *aseplot-options* option is available:

STEPAXIS= *EFFECT | NORMBETA | NUMBER*

specifies the horizontal axis to be used.

CANDIDATES | CANDIDATESPLOT <(*candidatesplot-options*)>

produces a needle plot of the values select criterion for the candidates for entry or removal at each step of the selection process, ordered from best to worst. Candidates plots are not available if you request SELECTION=NONE, SELECTION=LAR, SELECTION=LASSO on the model statement, or if you have not requested DETAILS=ALL or DETAILS=STEPS on the MODEL statement. The following *candidatesplot-options* are available:

LOGP | LOGPVALUE

requests that the natural logarithm of the entry and removal significance levels be displayed. This option is ignored if the select criterion is not significance level.

SHOW= *number*

specifies the maximum number of candidates displayed at each step. The default is SHOW=10.

COEFFICIENTS | COEFFICIENTPANEL(<*coefficientPanel-options*)>

plots a panel of two plots. The upper plot shows the evolution of the parameter values as the selection process proceeds. The lower plot shows the evolution of the CHOOSE= criterion. If no choose criterion is in effect, then the AICC criterion is displayed. The following *coefficientPanel-options* are available:

LABELGAP=*percentage*

specifies percentage of the vertical axis range that forms the minimum gap between successive parameter labels at the final step of the coefficient evolution plot. If the values of more than one parameter at the final step are closer than this gap, then the labels on all but one of these parameters is suppressed. The default value is LABELGAP=5.

LOGP | LOGPVALUE

requests that the natural logarithm of the entry and removal significance levels be displayed if the choose criterion is significance level.

STEPAXIS= EFFECT | NORMBETA | NUMBER

specifies the horizontal axis to be used.

UNPACK | UNPACKPANEL

displays the coefficient evolution and the choose criterion evolution on separate plots.

CRITERION | CRITERIONPANEL<(*criterionPanel-options***)>**

plots a panel of model fit criteria. The criteria that are displayed are ADJRSQ, AIC, AICC and SBC, as well as any other criteria that are named on the CHOOSE=, SELECT=, STOP=, or STATS= options on the MODEL statement. The following *criterionPanel-options* are available:

STEPAXIS= EFFECT | NORMBETA | NUMBER

specifies the horizontal axis to be used.

UNPACK | UNPACKPANEL

displays the coefficient evolution and the choose criterion evolution on separate plots.

NONE

suppresses all plots.

Multiple options and suboptions can be specified together, for example,

```
plots(StepAxis=effect)=( Coefficients(unpack logp)
                         ASE(StepAxis=number))
```

You can also specify other options with ALL. For example, to request all plots and unpack just the CRITERIONPANEL, specify:

```
plots=(all criterionPanel(unpack)).
```

SEED=*number*

specifies an integer used to start the pseudo-random number generator for random cross validation and random partioning of data for training, testing, and validation. If you do not specify a seed, or specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer's clock.

TESTDATA=*SAS-data-set*

names a SAS data set containing test data. This data set must contain all the variables specified in the MODEL statement. Furthermore, when a BY statement is used and the TESTDATA=data set contains any of the BY variables, then the TESTDATA= data set must also contain all the BY variables sorted in the order of the BY variables. In this case, only the test data for a specific BY group is used with the corresponding BY group in the analysis data. If the TESTDATA= data set contains none of the BY

variables, then the entire TESTDATA = data set is used with each BY group of the analysis data.

If you specify a TESTDATA=data set then you cannot also reserve observations for testing using a PARTITION statement.

VALDATA=*SAS-data-set*
names a SAS data set containing validation data. This data set must contain all the variables specified in the MODEL statement. Furthermore, when a BY statement is used and the VALDATA=data set contains any of the BY variables, then the VALDATA= data set must also contains all the BY variables sorted in the order of the BY variables. In this case, only the validation data for a specific by group is used with the corresponding BY group in the analysis data. If the VALDATA= data set contains none of the BY variables, then the entire VALDATA = data set is used with each BY group of the analysis data.

If you specify a VALDATA=data set then you cannot also reserve observations for validation using a PARTITION statement.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC GLMSELECT to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in the order of the BY variables. The *variables* are one or more variables in the input data set.

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

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the GLMSELECT 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 using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Contents*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

The CLASS statement names the classification variables to be used in the analysis. The CLASS statement must precede the MODEL statement.

The following options can be specified after a slash (/).

DELIMITER=*quoted character*
specifies the delimiter that is used between levels of class variables in building parameter names and lists of class level values. The default if you do not specify DELIMITER= is a space. This option is useful if the levels of a class variable contain embedded blanks.

SHOW | SHOWCODING
requests a table for each classification variable that shows the coding used for that variable.

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

The following *v-options* are available:

DESCENDING
DESC
reverses the sorting order of the classification variable.

MISSING
allows missing value ('.' for a numeric variable and blanks for a character variables) as a valid value for the CLASS variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sorting order for the levels of classification variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option may be useful when you use the CONTRAST or ESTIMATE statement. If ORDER=FORMATTED for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. Note that this represents a change from previous releases for how class levels are ordered. In releases previous to Version 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering. The following table shows how PROC GLMSELECT interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information on sorting order, refer to the chapter on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*keyword*

specifies the parameterization method for the classification variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes. The default is PARAM=GLM. If PARAM=ORTHPOLY or PARAM=POLY, and the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used. See the "CLASS Variable Parameterization" section on page 45 for further details.

EFFECT	specifies effect coding
GLM	specifies less-than-full-rank, reference-cell coding; this option can only be used as a global option
ORDINAL THERMOMETER	specifies the cumulative parameterization for an ordinal CLASS variable.
POLYNOMIAL POLY	specifies polynomial coding
REFERENCE REF	specifies reference cell coding
ORTHEFFECT	orthogonalizes PARAM=EFFECT
ORTHORDINAL ORTHOTHERM	orthogonalizes PARAM=ORDINAL
ORTHPOLY	orthogonalizes PARAM=POLYNOMIAL
ORTHREF	orthogonalizes PARAM=REFERENCE

The EFFECT, POLYNOMIAL, REFERENCE, ORDINAL, and their orthogonal parameterizations are full rank. The REF= option in the CLASS statement determines the reference level for the EFFECT, REFERENCE, and their orthogonal parameterizations.

REF=*'level'* | *keyword*

specifies the reference level for PARAM=EFFECT, PARAM=REFERENCE, and their orthogonalizations. For an individual (but not a global) variable REF= *option*, you can specify the *level* of the variable to use as the reference level. For a global or individual variable REF= *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

SPLIT

requests that the columns of the design matrix corresponding to any effect containing a split class variable can be selected to enter or leave a model independently of the other design columns of that effect. For example, suppose a variable named "temp" has three levels with values "hot," "warm," and "cold," and a variable named "sex" has two levels with values "M" and "F" are used in a PROC GLMSELECT job as follows:

```
proc glmselect;
   class temp sex/split;
   model depVar = sex sex*temp;
run;
```

As both the classification variables are split, the two effects named on the MODEL statement are split into eight independent effects. The effect "sex" is split into two effects labeled "sex_M" and "sex_F". The effect "sex*temp" is split into six effects labeled "sex_M*temp_hot", "sex_F*temp_hot", "sex_M*temp_warm", "sex_F*temp_warm", "sex_M*temp_cold", and "sex_F*temp_cold", and the previous PROC GLMSELECT step is equivalent to

```
proc glmselect;
   model depVar =  sex_M sex_F sex_M*temp_hot  sex_F*temp_hot
                               sex_M*temp_warm sex_F*temp_warm
                               sex_M*temp_cold sex_F*temp_cold;
run;
```

The split option can be used on individual class variables. For example, consider the following PROC GLMSELECT job:

```
proc glmselect;
   class temp(split) sex;
   model depVar = sex sex*temp;
run;
```

In this case the effect sex is not split and the effect sex*temp is split into three effects labeled "sex*temp_hot", "sex*temp_warm", and "sex*temp_cold". Furthermore each of these three split effects now has two parameters corresponding to the two levels of "sex," and the PROC GLMSELECT step is equivalent to

```
proc glmselect;
   class sex;
   model depVar = sex sex*temp_hot sex*temp_warm sex*temp_cold;
run;
```

FREQ Statement

FREQ *variable* **;**

The variable specified on the FREQ statement identifies a variable in the input data set containing the frequency of occurrence of each observation. PROC GLMSELECT treats each observation as if it appears n times, where n is the value of the FREQ variable for the observation. If it is not an integer, the frequency value is truncated to an integer. If it is less than 1 or if it is missing, the observation is not used.

MODEL Statement

MODEL *dependent*=< *effects* > / <*options* > **;**

The MODEL statement names the dependent variable and the explanatory effects, including covariates, main effects, interactions, and nested effects; see the section "Specification of Effects" (Chapter 32, (*SAS/STAT User's Guide*) on page 1785 of Chapter 32, "The GLM Procedure," (*SAS/STAT User's Guide*) for more information. If you omit the explanatory effects, the procedure fits an intercept-only model.

After the keyword MODEL, the dependent (response) variable is specified, followed by an equal sign. The explanatory effects follow the equal sign.

Table 2 lists the options available in the MODEL statement.

Table 2. MODEL Statement Options

Option	Description
CVDETAILS=	requests details when cross validation is used
CVMETHOD=	specifies how subsets for cross validation are formed
DETAILS=	specifies details to be displayed
HIERARCHY=	specifies hierarchy of effects to impose
NOINT	specifies models without an explicit intercept
ORDERSELECT	requests parameter estimates be displayed in the order that the parameters first entered the model
SELECTION=	specifies model selection method
STATS=	specifies additional statistics to be displayed
SHOWPVALUES	requests p-values in "ANOVA" and "Parameter Estimates" tables
STB	adds standardized coefficients to "Parameter Estimates" tables

You can specify the following options in the MODEL statement after a slash (/).

CVDETAILS=ALL
CVDETAILS=COEFFS
CVDETAILS=CVPRESS

specifies the details produced when cross validation is requested as the CHOOSE=, SELECT=, or STOP= criterion on the MODEL statement. If n-fold cross validation is being used, then the training data are subdivided into n parts and at each step of the selection process, models are obtained on each of the n subsets of the data obtained by omitting one of these parts. CVDETAILS=COEFFS requests that the parameter estimates obtained for each of these n subsets be included in the parameter estimate table. CVDETAILS=CVPRESS requests a table containing the predicted residual sum of squares of each of these models scored on the omitted subset. CVDETAILS=ALL requests both CVDETAILS=COEFFS and CVDETAILS=CVPRESS. If DETAILS=STEPS or DETAILS=ALL has been specified on the MODEL statement, then the requested CVDETAILS are produced for every step of the selection process.

CVMETHOD=BLOCK < (*n*) >
CVMETHOD=RANDOM < (*n*) >
CVMETHOD=SPLIT < (*n*) >
CVMETHOD=INDEX (*variable*)

specifies how the training data are subdivided into n parts when you request n-fold cross validation using any of the CHOOSE=CV, SELECT=CV, or STOP=CV suboptions on the SELECTION= option on the MODEL statement.

- CVMETHOD=BLOCK requests parts be formed of n blocks of consecutive training observations.
- CVMETHOD=SPLIT requests that the *i*th part consists of training observations $i, i+n, i+2n, ...$
- CVMETHOD=RANDOM assigns each training observation randomly to one of the n parts.
- CVMETHOD=*variable* assigns observations to parts based on the formatted value of the named variable. This input data set variable is treated as a classification variable and the number of parts n is the number of distinct levels of this variable. By optionally naming this variable on a CLASS statement you can use the CLASS statement options ORDER= and MISSING to control how the levelization of this variable is done.

n defaults to 5 with CVMETHOD=BLOCK, CVMETHOD=SPLIT, or CVMETHOD=RANDOM. If you do not specify the CVMETHOD= option, then the CVMETHOD defaults to CVMETHOD=RANDOM(5).

DETAILS=*level*
DETAILS=STEPS < (*step options*) >

specifies the level of detail produced, where *level* can be ALL, STEPS, or SUMMARY. The default if the DETAILS= option is omitted is DETAILS=SUMMARY. The DETAILS=ALL option produces

- entry and removal statistics for each variable selected in the model building process
- ANOVA, fit statistics, and parameter estimates
- entry and removal statistics for the top 10 candidates for inclusion or exclusion at each step
- a selection summary table

The DETAILS=SUMMARY option produces only the selection summary table.

The option DETAILS=STEPS < (*step options*) > provides the step information and the selection summary table. The following options can be specified within parentheses after the DETAILS=STEPS option:

ALL requests ANOVA, fit statistics, parameter estimates, and entry or removal statistics for the top 10 candidates for inclusion or exclusion at each selection step.

ANOVA requests ANOVA at each selection step.

FITSTATISTICS | FITSTATS | FIT requests fit statistics at each selection step. The default set of statistics includes all of the statistics named in the CHOOSE=, SELECT=, and STOP= specified on the MODEL statement SELECTION= option, but additional statistics can be requested with the STATS= option on the MODEL statement.

PARAMETERESTIMATES | PARMEST requests parameter estimates at each selection step.

CANDIDATES <(SHOW= ALL | *n*)> requests entry or removal statistics for the best *n* candidate effects for inclusion or exclusion at each step. If you specify SHOW=ALL then all candidate are shown. If SHOW= is not specified then the best 10 candidates are shown. The entry or removal statistic is the statistic named in the SELECT= option that is specified on the MODEL statement SELECTION= option.

HIERARCHY=*keyword*
HIER=*keyword*

specifies whether and how the model hierarchy requirement is applied. This option also controls whether a single effect or multiple effects are allowed to enter or leave the model in one step. 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: SELECTION=FORWARD, SELECTION=BACKWARD, or SELECTION=STEPWISE.

Model hierarchy refers to the requirement that for any term to be in the model, all model effects 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 B can leave the model while the interaction A*B is in the model.

The keywords you can specify in the HIERARCHY= option are as follows:

NONE	Model hierarchy is not maintained. Any single effect can enter or leave the model at any given step of the selection process.
SINGLE	Only one effect can 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	This is the same as HIERARCHY=SINGLE except that only CLASS effects are subject to the hierarchy requirement.

The default value is HIERARCHY=NONE.

NOINT
suppresses the intercept term that is otherwise included in the model.

ORDERSELECT
specifies that for the selected model, effects are displayed in the order that they first entered the model. If you do not specify the ORDERSELECT option, then effects in the selected model are displayed in the order that they appeared on the MODEL statement.

SELECTION=*method* <**(method options)**>
specifies the method used to select the model, optionally followed by parentheses enclosing options applicable to the specified method. The default if the SELECTION= option is omitted is SELECTION=FORWARD.

The following methods are available and are explained in detail in the "Model-Selection Methods" section on page 35.

NONE	no model selection.
FORWARD	forward selection. This method starts with no effects in the model and adds effects.
BACKWARD	backward elimination. This method starts with all effects in the model and deletes effects.
STEPWISE	stepwise regression. This is similar to the FORWARD method except that effects already in the model do not necessarily stay there.
LAR	least angle regression. This method, like forward selection, starts with no effects in the model and adds effects. The parameter

estimates at any step are "shrunk" when compared to the corresponding least squares estimates. If the model contains classification variables, then these classification variables are split. See the SPLIT option on the CLASS statement for details.

LASSO — This method adds and deletes parameters based on a version of ordinary least squares where the sum of the absolute regression coefficients is constrained. If the model contains classification variables, then these classification variables are split. See the SPLIT option on the CLASS statement for details.

Table 3 lists the applicable suboptions for each of these methods:

Table 3. Applicable SELECTION= Suboptions by Method

Option	FORWARD	BACKWARD	STEPWISE	LAR LASSO
STOP =	x	x	x	x
CHOOSE =	x	x	x	x
STEPS =	x	x	x	x
MAXSTEPS =	x	x	x	x
SELECT =	x	x	x	
INCLUDE =	x	x	x	
SLENTRY =	x		x	
SLSTAY =		x	x	
DROP =			x	
LSCOEFFS				x

The syntax of the suboptions that you can specify in parentheses after the SELECTION= option method follows. Note that, as described in Table 3, not all selection suboptions are applicable to every SELECTION= method.

CHOOSE=*criterion*

chooses from the list of models at the steps 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 with 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 you can specify in the CHOOSE= option are shown in Table 4. See the "Criteria Used in Model Selection Methods" section on page 44 for more detailed descriptions of these criteria.

Table 4. Criteria for the CHOOSE= Option

Option	Criteria
ADJRSQ	Adjusted R-square statistic
AIC	Akaike information criterion
AICC	Corrected Akaike information criterion
BIC	Sawa Bayesian information criterion
CP	Mallow C(p) statistic
CV	Predicted residual sum of square with k-fold cross validation
PRESS	Predicted residual sum of squares
SBC	Schwarz Bayesian information criterion
VALIDATE	Average square error for the validation data

For ADJRSQ the chosen value is the largest one; for all other criteria, the smallest value is chosen. You can use the VALIDATE option only if you have specified a VALDATA= data set on the PROC GLMSELECT statement or if you have reserved part of the input data for validation using either a PARTITION statement or a _ROLE_ variable in the input data.

DROP=*policy*

specifies when effects are eligible to be dropped in the STEPWISE method. Valid values for policy are BEFOREADD and COMPETITIVE.

If you specify DROP=BEFOREADD, then effects currently in the model are examined to see if any meet the requirements to be removed from the model. If so, the effect that gives the best value of the removal criterion is dropped from the model and the stepwise method proceeds to the next step. Only when no effect currently in the model meets the requirement to be removed from the model are any effects be added to the model.

DROP=COMPETITIVE can only be specified if the SELECT= criterion is not SL. If you specify DROP=COMPETITIVE then the SELECT= criterion is evaluated for all models where an effect currently in the model is dropped or an effect not yet in the model is added. The effect whose removal or addition to the model yields the maximum improvement to the SELECT= criterion is dropped or added.

The default if you do not specify DROP= with the STEPWISE method is DROP=BEFOREADD. If the SELECT=SL (the default), then this yields the traditional stepwise method as implemented in PROC REG.

INCLUDE=*n*

forces the first n effects listed in the MODEL statement to be included in all models. The selection methods are performed on the other effects in the MODEL statement. The INCLUDE= option is available only with SELECTION=FORWARD, SELECTION=STEPWISE, and SELECTION=BACKWARD.

LSCOEFFS

requests a hybrid version of the LAR and LASSO methods where 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 using ordinary least squares.

MAXSTEP=*n*

specifies the maximum number of selection steps that are done. The default value of *n* is the number of effects in the model statement for the FORWARD, BACKWARDS, and LAR methods and is three times the number of effects for the STEPWISE and LASSO methods.

SELECT=*criterion*

specifies the criterion that PROC GLMSELECT uses to determine the order in which effects enter and/or leave at each step of the specified selection method. The SELECT option is not valid with the LAR and LASSO methods. The criteria that you can specify with the SELECT= option are ADJRSQ, AIC, AICC, BIC, CP, CV, PRESS, RSQUARE, SBC, SL, and VALIDATE. See the "Criteria Used in Model Selection Methods" section on page 44 for a description of these criteria. The default value of the SELECT= criterion is SELECT=SL for selecting by significance level. With this default the FORWARD, BACKWARDS, and STEPWISE selection methods are the traditional approaches where effects enter and leave the model based on the significance level. With other SELECT= criteria, the effect that is selected to enter or leave at a step of the selection process is the effect whose addition to or removal from the current model gives the maximum improvement in the specified criterion.

SLENTRY=*value*
SLE=*value*

specifies the significance level for entry, used when the STOP=SL or SELECT=SL options are in effect. The default is 0.15.

SLSTAY=*value*
SLS=*value*

specifies the significance level for staying in the model, used when the STOP=SL or SELECT=SL options are in effect. The default is 0.15.

STEPS=*n*

specifies the number of selection steps to be done. If the STEPS= option is specified, the STOP= and MAXSTEP= options are ignored.

STOP=*n*
STOP=*criterion*

specifies when PROC GLMSELECT stops the selection process. If the STEPS= option is specified, then the STOP= option is ignored. If the STOP=option does not cause the selection process to stop before the maximum number of steps for the selection method, then the selection process terminates at the maximum number of steps.

If you do not specify the STOP= option but do specify the SELECT= option, then the criterion named in the SELECT=option is also used as the STOP= criterion. If you do not specify either of the STOP= or SELECT= options, then the default is STOP=SL for the FORWARD, BACKWARD and STEPWISE methods and STOP=NONE for the LAR and LASSO methods.

If STOP=*n* is specified then PROC GLMSELECT stops selection at the first step for which the selected model has *n* effects.

The non-numeric arguments that you can specify in the STOP= option are shown in Table 5. See the "Criteria Used in Model Selection Methods" section on page 44 for more detailed descriptions of these criteria.

Table 5. Non-Numeric Criteria for the STOP= Option

Option	Criteria
NONE	
ADJRSQ	Adjusted R-square statistic
AIC	Akaike information criterion
AIC	Corrected Akaike information criterion
BIC	Sawa Bayesian information criterion
CP	Mallow C(p) statistic
CV	Predicted residual sum of square with k-fold cross validation
PRESS	Predicted residual sum of squares
SBC	Schwarz Bayesian information criterion
SL	Significance Level
VALIDATE	Average square error for the validation data

With the SL criterion, selection stops at the step where the significance level for entry of all the effects not yet in the model is greater than the SLE= value for addition steps in the FORWARDS and STEPWISE methods and the significance level for removal of any effect in the current model is greater than the SLS= value in the BACKWARDS and STEPWISE methods. With the ADJRSQ criterion, selection stops at the step where the next step would yield a model with a smaller value of the Adjusted R-square statistic; for all other criteria, selection stops at the step where the next step would yield a model with a larger value of the criteria. You can use the VALIDATE option only if you have specified a VALDATA= data set on the PROC GLMSELECT statement or if you have reserved part of the input data for validation using a either a PARTITION statement or a _ROLE_ variable in the input data.

STAT|STATS=*name*
STATS=(*names***)**

specifies which model fit statistics are displayed in the fit summary table and fit statistics tables. If you omit the STATS= option, the default set of statistics that are displayed in these tables includes all the criteria specified on any of the CHOOSE=, SELECT=, and STOP= options specified on the MODEL statement SELECTION= option.

The statistics that you can specify follow:

ADJRSQ — the Adjusted R-square statistic

AIC — the Akaike Information Criterion

AICC — the Corrected Akaike Information Criterion

ASE — the average square errors for the training, test, and validation data. The ASE statistics for the test and validation data are reported only if you have specified TESTDATA= and/or VALDATA= on the PROC GLMSELECT statement or if you have reserved part of the

	input data for testing and/or validation using a either a PARTITION statement or a _ROLE_ variable in the input data.
BIC	the Sawa Bayesian Information Criterion
CP	the Mallow C(p) statistic
FVALUE	the F statistic for entering or departing effects
PRESS	the predicted residual sum of squares statistic
RSQUARE	the R-square statistic
SBC	the Schwarz Bayesian Information Criterion
SL	the significance level of the F statistic for entering or departing effects

The statistics ADJRSQ, AIC, AICC, FVALUE, RSQUARE, SBC, and SL can be computed with little computation cost. However, computing BIC, CP, CVPRESS, PRESS, and ASE for test and validation data when these are not used in any of the CHOOSE=, SELECT=, and STOP= options specified on the MODEL statement SELECTION= option can negatively impact performance.

SHOWPVALUES
SHOWPVALS
displays p-values in the "ANOVA" and "Parameter Estimates" tables. These p-values are generally liberal because they are not adjusted for the fact that the terms in the model have been selected.

STB
produces standardized regression coefficients. A standardized regression coefficient is computed by dividing a parameter estimate by the ratio of the sample standard deviation of the dependent variable to the sample standard deviation of the regressor.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* > < *keyword* <=*name* > > ... < *keyword* <=*name* > > **;**

The OUTPUT statement creates a new SAS data set that saves diagnostic measures calculated for the selected model. If you do not specify any *keyword* then the only diagnostic included is the predicted response.

All the variables in the original data set are included in the new data set, along with variables created in the OUTPUT statement. These new variables contain the values of a variety of statistics and diagnostic measures that are calculated for each observation in the data set. If you specify a BY statement then a variable _BY_ that indexes the BY groups is included. For each observation, the value of _BY_ is the index of the BY group to which this observation belongs. This variable is useful for matching BY groups with macro variables that PROC GLMSELECT creates. See the "Macro Variables Containing Selected Models" section on page 50 for details.

If you have requested n-fold cross validation by requesting CHOOSE=CV, SELECT=CV, or STOP=CV on the MODEL statement, then a variable _CVINDEX_

is included in the output data set. For each observation used for model training the value of _CVINDEX_ is i if that observation is omitted in forming the ith subset of the training data. See the CVMETHOD= for additional details. The value of _CVINDEX_ is 0 for all observations in the input data set that are not used for model training.

If you have partitioned the input data with a PARTITION statement, then a character variable _ROLE_ is included in the output data set. For each observation the value of _ROLE_ is:

ROLE	Observation Role
TEST	testing
TRAIN	training
VALIDATE	validation

If you want to create a permanent SAS data set, you must specify a two-level name (for example, *libref.data-set-name*). For more information on permanent SAS data sets, refer to the section "SAS Files" in *SAS Language Reference: Concepts*.

Details on the specifications in the OUTPUT statement follow.

keyword <*=name* >
specifies the statistics to include in the output data set and optionally names the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), followed optionally by an equal sign, and variable to contain the statistic.

If you specify *keyword=name*, the new variable that contains the requested statistic has the specified name. If you omit the optional *=name* after a *keyword*, then the new variable name is formed by using a prefix of one or more characters that identify the statistic, followed by an underscore (_), followed by the dependent variable name.

The keywords allowed and the statistics they represent are as follows:

PREDICTED | PRED | P predicted values. The prefix for the default name is *p*.

RESIDUAL | RESID | R residual, calculated as ACTUAL − PREDICTED. The prefix for the default name is *r*.

OUT=*SAS data set*
gives the name of the new data set. By default, the procedure uses the DATA*n* convention to name the new data set.

PARTITION Statement

The PARTITION statement specifies how observations in the input data set are logically partitioned into disjoint subsets for model training, validation, and testing. You can either designate a variable in the input data set and a set of formatted values of that variable to determine the role of each observation or you can specify proportions to use for random assignment of observations for each role.

An alternative to using a PARTITION statement is to provide a variable named _ROLE_ in the input data set to define roles of observations in the input data. If you specify a PARTITION statement then the _ROLE_ variable if present in the input data set is ignored. If you do not use PARTITION statement and the input data does not contain a variable named _ROLE_ then all observations in the input data set are assigned to model training.

The following mutually exclusive options are available:

ROLEVAR | ROLE=variable(<TEST=value> <TRAIN=value> <VALIDATE=value>)
names the variable in the input data set whose values are used to assign roles to each observation. The formatted values of this variable that are used to assign observations roles are specified on the TEST=, TRAIN=, and VALIDATE= suboptions. If you do not specify the TRAIN= suboption, then all observations whose role is not determined by the TEST= or VALIDATE= suboptions are assigned to training. If you specify a TESTDATA= data set on the PROC GLMSELECT statement, then you cannot also specify the TEST= suboption on the PARTITION statement. If you specify a VALDATA= data set on the PROC GLMSELECT statement, then you cannot also specify the VALIDATE= suboption on the PARTITION statement.

FRACTION(<TEST=fraction> <VALIDATE=fraction>)
requests that specified proportions of the observations in the input data set be randomly assigned training and validation roles. You specify the proportions for testing and validation using the TEST= and VALIDATE=suboptions. If you specify both the TEST= and the VALIDATE= suboptions, then the sum of the specified fractions must be less than one and the remaining fraction of the observations are assigned to the training role. If you specify a TESTDATA= data set on the PROC GLMSELECT statement, then you cannot also specify the TEST= suboption on the PARTITION statement. If you specify a VALDATA= data set on the PROC GLMSELECT statement, then you cannot also specify the VALIDATE= suboption on the PARTITION statement.

PERFORMANCE Statement

PERFORMANCE < *options* > **;**

The PERFORMANCE statement is used to change default options that impact the performance of PROC GLMSELECT and to request tables that show the performance options in effect and timing details.

The following options are available:

CPUCOUNT = *1-1024*
CPUCOUNT = *ACTUAL*
specifies the number of processors that PROC GLMSELECT assumes are available for multithreaded BY group processing. CPUCOUNT=ACTUAL sets CPUCOUNT to be the number of physical processors available. Note that this can be less than the physical number of CPUs if the SAS process has been restricted by system administration tools. Setting CPUCOUNT= to a number greater than the actual number of available CPUs might result in reduced performance. This option overrides the SAS system option CPUCOUNT=. If CPUCOUNT=1, NOTHREADS is in effect, or no BY processing is being used, then PROC GLMSELECT uses singly threaded code.

DETAILS
requests the "PerfSettings" table that shows the performance settings in effect and the "Timing" table that provides a broad timing breakdown of the PROC GLMSELECT step.

BUILDSSCP=FULL
BUILDSSCP=INCREMENTAL
specifies whether the SSCP matrix is built incrementally as the selection process progresses or whether the SCCP matrix for the full model is built at the outset. Building the SSCP matrix incrementally can significantly reduce memory required and the time taken to perform model selection in cases where the number of parameters in the selected model is much smaller than the number of parameters in the full model, but can adversely impact performance in other cases as it requires at least one pass over the model training data at each step. If you use backward selection or no selection, or if the BIC or CP statistics are required in the selection process, then the BUILDSSCP=INCREMENTAL option is ignored. In other cases, BUILDSSCP=INCREMENTAL is used by default if the number of effects is greater than 100. See the "Building the SSCP Matrix" section on page 53 for further details.

THREADS
enables multithreaded BY group computation. This option overrides the SAS system option THREADS | NOTHREADS. If no BY processing is being used, then PROC GLMSELECT ignores this option and uses singly threaded code.

NOTHREADS
disables multithreaded BY group computation. This option overrides the SAS system option THREADS | NOTHREADS.

SCORE Statement

SCORE < **DATA=***SAS-data-set* > < **OUT=***SAS-data-set* > < *keyword* <=*name* > > . . . < *keyword* <=*name* > > **;**

The SCORE statement creates a new SAS data set containing predicted values and optionally residuals for data in a new data set that you name. If you do not specify a DATA= data set, then the input data are scored. If you have multiple data sets to predict, you can specify multiple SCORE statements. If you want to create a permanent SAS data set, you must specify a two-level name (for example, *libref.data-*

set-name) on the OUT= option. For more information on permanent SAS data sets, refer to the section "SAS Files" in *SAS Language Reference: Concepts*.

When a BY statement is used, the score data set must either contain all the BY variables sorted in the order of the BY variables or must contain none of the BY variables. If the score data set contains all of the BY variables, then the model selected for a given BY group is used to score just the matching observations in the score data set. If the score data set contains none of the BY variables, then the entire score data set is scored for each BY group.

All observations in the score data set are retained in the output data set. However, only those observations that contain nonmissing values for all the continuous regressors in the selected model and whose levels of the class variables appearing in effects of the selected model are represented in the corresponding class variable in the procedure's input data set are scored. All the variables in the input data set are included in the output data set, along with variables containing predicted values and optionally residuals.

Details on the specifications in the SCORE statement follow:

DATA=*SAS data set*

names the data set to be scored. If you omit this option then the input data set named in the DATA= option on the PROC GLMSELECT statement is scored.

keyword <*=name* >

specifies the statistics to include in the output data set and optionally names the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), followed optionally by an equal sign, and variable to contain the statistic.

If you specify *keyword=name*, the new variable that contains the requested statistic has the specified name. If you omit the optional *=name* after a *keyword*, then the new variable name is formed by using a prefix of one or more characters that identify the statistic, followed by an underscore (_), followed by the dependent variable name.

The keywords allowed and the statistics they represent are as follows:

PREDICTED | PRED | P predicted values. The prefix for the default name is *p*.

RESIDUAL | RESID | R residual, calculated as ACTUAL − PREDICTED. The prefix for the default name is *r*.

OUT=*SAS data set*

gives the name of the new output data set. By default, the procedure uses the DATA*n* convention to name the new data set.

WEIGHT Statement

WEIGHT *variable* **;**

A WEIGHT statement names a variable in the input data set with values that are relative weights for a weighted least-squares fit. If the weight value is proportional to the reciprocal of the variance for each observation, then the weighted estimates are the best linear unbiased estimates (BLUE).

Values of the weight variable must be nonnegative. If an observation's weight is zero, the observation is deleted from the analysis. If a weight is negative or missing, it is set to zero, and the observation is excluded from the analysis. A more complete description of the WEIGHT statement can be found in Chapter 32, "The GLM Procedure." (*SAS/STAT User's Guide*)

Details

Model-Selection Methods

The model selection methods implemented in PROC GLMSELECT are specified with the SELECTION= option on the MODEL statement.

Full Model Fitted (NONE)

The complete model specified in the MODEL statement is used to fit the model and no effect selection is done. You request this by specifying SELECTION=NONE on the model statement.

Forward Selection (FORWARD)

The forward selection technique 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 used to gauge improvement in fit is an F statistic that reflects an effect's contribution to the model if it is included. At each step, the effect that yields the most significant F statistic is added. Note that because effects can contribute different degrees of freedom to the model, it is necessary to compare the p-values corresponding to these F statistics.

More precisely, if the current model has p parameters excluding the intercept and you denote its residual sum of squares by RSS_p and you add an effect with k degrees of freedom and denote the residual sum of squares of the resulting model by RSS_{p+k}, then the F statistic for entry with k numerator degrees of freedom and $n-(p+k)-1$ denominator degrees of freedom is given by

$$F = \frac{(RSS_p - RSS_{p+k})/k}{RSS_{p+k}/(n-(p+k)-1)}$$

where n is number of observations used in the analysis.

The process stops when the significance level for adding any effect is greater than some specified entry significance level. A well known problem with this methodology is that these F statistics do not follow an F distribution (Draper, Guttman, and Kanemasu 1971). Hence these p-values cannot reliably be interpreted as probabilities. Various ways to approximate this distribution are described in Miller, 2002. Another issue in using significance levels of entering effects as a stopping criterion arises because the entry significance level is an *a priori* specification that does not depend on the data. Thus, the same entry significance level can result in overfitting for some data and underfitting for other data.

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

You use the CHOOSE= option of forward selection 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

```
selection=forward(choose=AIC SLE=0.2)
```

then forward selection terminates at the step where no effect can be added at the 0.2 significance level. However, the selected model is the first one with the minimal value of the Akaike Information Criterion. Note that in some cases this minimal value might occur at a step much earlier that the final step, while in other cases the AIC criterion might only start increasing if more steps are done (i.e., a larger value of SLE is used). If what you are interested in is minimizing AIC, then too many steps are done in the former case and too few in the latter case. To address this issue, PROC GLMSELECT enables you to specify a stopping criterion with the STOP= option. With a stopping criterion specified, forward selection continues until a local extremum of the stopping criterion in the sequence of models generated is reached. You can also specify STOP= number, which causes forward-selection to continue until there are the specified number of effects in the model.

For example, if you specify

```
selection=forward(stop=AIC)
```

then forward selection terminates at the step where the effect to be added at the next step would produce a model with AIC statistic larger than the AIC statistic of the current model. Note that in most cases, provided that the entry significance level is large enough so that the local extremum of the named criterion occurs before the final step, specifying

```
selection=forward(choose=CRITERION) or
selection=forward(stop=CRITERION)
```

selects the same model, but more steps are done in the former case. 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 examine how the named criterion behaves as you move beyond the step where the first local minimum of this criterion occurs.

Note that you can specify both the CHOOSE= and STOP= options. You may want to consider models generated by forward selection that have at most some fixed number of effects but select from within this set based on a criterion you specify. For example, specifying

```
selection=forward(stop=20 choose=ADJRSQ)
```

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. You can also combine these options to select a model where one of two conditions is met. For example,

```
selection=forward(stop=AICC choose=PRESS)
```

chooses whatever occurs first between a local minimum of the predicted residual sum of squares (PRESS) or at a local minimum of the Corrected Akaike Information Criterion (AICC).

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 may not be optimal. For example, the effect you would add to get a model with the smallest value of the PRESS statistic at the next step is not necessarily the same effect that has the most significant entry F statistic. PROC GLMSELECT enables you to specify the criterion to optimize at each step using the SELECT= option. For example,

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

requests that at each step the effect that is added is the one that gives a model with the smallest value of the Mallow's $C(p)$ statistic. Note that in the case where all effects are variables (that is, effects with one degree of freedom and no hierarchy), using any of 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 SELECT= together with CHOOSE= and STOP=. 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 Mallow's $C(p)$ statistic, but the selection terminates when the $C(p)$ statistic first increases.

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

```
selection=forward
```

Add effects based on significance level and stop when all candidate effects for entry at a step have significance level greater than the default entry significance level of 0.15.

```
selection=forward(stop=validation)
```

Add effects based on significance level and stop at a step where adding any effect increases the error sum of squares computed on the validation data.

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

Add effects that at each step give the lowest value of the AIC statistic and stop at the step where adding any effect would increase the AIC statistic.

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

adds effects that at each step give the largest value of the adjusted R-square statistic and stops at the step where the significance level corresponding to the addition of this effect is greater than 0.2.

Backward Elimination (BACKWARD)

The backward elimination technique starts from the full model including all independent effects. Then effects are deleted one by one until a stopping condition is satisfied. At each step, the effect showing the smallest contribution to the model is deleted. In traditional implementations of backwards elimination, the contribution of an effect to the model is assessed using an F statistic. At any step, the predictor producing the least significant F statistic is dropped and the process continues until all effects remaining in the model have F statistics significant at a stay significance level (SLS).

More precisely, if the current model has p parameters excluding the intercept and you denote its residual sum of squares by RSS_p and you drop an effect with k degrees of freedom and denote the residual sum of squares of the resulting model by RSS_{p-k},

then the F statistic for removal with k numerator degrees of freedom and $n - p - k$ denominator degrees of freedom is given by

$$F = \frac{(RSS_{p-k} - RSS_p)/k}{RSS_p/(n-p-k)}$$

where n is number of observations used in the analysis.

Just as with forward selection, you can change the criterion used to assess effect contributions with the SELECT= option. You can also specify a stopping criterion with the STOP= option and you can use a CHOOSE= option to provide a criterion used to select among the sequence of models produced. See the discussion in the "Forward Selection (FORWARD)" section on page 35 for additional details.

Examples of Backward Selection Specifications

```
selection=backward
```

Remove effects based on significance level and stop when all candidate effects for removal at a step have significance level less than the default stay significance level of 0.15.

```
selection=backward(stop=press)
```

Remove effects based on significance level and stop at the step where removing any effect increases the predicted residual sum of squares (PRESS).

```
selection=backward(choose=validate SLS=0.1)
```

Remove effects based on significance level and stop when all effects in the model are significant at the 0.1 level. Finally, from the sequence of models generated, choose the one that gives the smallest average square error when scored on the validation data.

```
selection=backward(select=BIC)
```

Removes effects that at each step produce the largest value of Sawa's Bayesian Information Criterion (BIC) statistic and stop at the step where removing any effect increases the BIC statistic.

Stepwise Selection(STEPWISE)

The stepwise method is a modification of the forward selection technique and 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 F statistics 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 yields the most significant F value 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 has an F statistic significant at the SLENTRY= level and every effect in the model is significant at the SLSTAY= level. In some cases, neither of these two conditions for stopping are met and the sequence of models cycles. In this case, the stepwise method terminates at the end of the second cycle.

Just as with forward selection and backwards elimination, you can change the criterion used to assess effect contributions, with the SELECT= option. You can also specify a stopping criterion with the STOP= option and you can use a CHOOSE= option to provide a criterion used to select among the sequence of models produced. See the discussion in the "Forward Selection (FORWARD)" section on page 35 for additional details.

For selection criteria other than significance level, PROC GLMSELECT 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 DROP=COMPETITIVE option to specify that addition and deletion of effects should be treated competitively. The selection criterion is evaluated for all models obtained 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

```
selection=stepwise
```

Requests the traditional stepwise method. First if the removal of any effect yields an F statistic that is not significant at the default stay level of 0.15 level then the effect whose removal produces the least significant F statistic is removed and the algorithm proceeds to the next step. Otherwise the effect whose addition yields the most significant F statistic is added, provided that it is significant at the default entry level of 0.15.

```
selection=stepwise(SLE=0.1 SLS=0.08 choose=AIC)
```

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=stepwise(stop=SBC)
```

This is the traditional stepwise method where effects enter and leave based on significance levels but with the following extra check: If any effect that is 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. Note that in this case, the entry and stay significance levels still play a role as they determine whether an effect is deleted from or added to the model. This might result in the selection terminating before a local minimum of the SBC criterion is found.

```
selection=stepwise(select=SBC)
```

This requests stepwise selection based on the SBC criterion. First, if removing any effect yields a model with a lower SBC statistic than the current model, then the effect producing the smallest SBC statistic is removed. When removing any effect increases the SBC statistic, then provided adding some effect lowers the SBC statistic, the effect producing the model with the lowest SBC is added.

```
selection=stepwise(select=SBC drop=COMPETITIVE)
```

This requests stepwise selection based on the SBC criterion with steps treated competitively. At any step, evaluate the SBC statistics corresponding to the removal of any effect in the current model or the addition of any effect to the current model. Choose the addition or removal that produced this minimum value, provided that this minimum is lower than the SBC statistic of the current model.

```
selection=stepwise(select=SBC drop=COMPETITIVE stop=VALIDATE)
```

This requests stepwise selection based on the SBC criterion with steps treated competitively and where stopping is based on the average square error over the validation data. At any step, SBC statistics corresponding 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. If this average square error is greater than the average square error on the validation data prior to this addition or deletion, then the algorithm terminates at this prior model.

Least Angle Regression (LAR)

Least angle regression 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 where 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 between these three predictors and the predicted response moves in this direction until a fourth predictor joins the set having the same correlation with the current residual. This process continues until all predictors are in the model.

As with other selection methods, the issue of when to stop the selection process is crucial. You can specify a criterion to use to choose among the models at each step with the CHOOSE= option. You can also specify a stopping criterion with the STOP= option. See the "Criteria Used in Model Selection Methods" section on page 44 for details and Table 6 for the formulae for evaluating these criteria. These formula 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 suggested in Efron et. al (2004) uses the LAR algorithm to select the set of covariates in the model at any step, but 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 SELECTION=LAR.

Lasso Selection (LASSO)

LASSO (Least Absolute Shrinkage and Selection Operator) selection arises from a constrained form of ordinary least squares where the sum of the absolute values of the regression coefficients is constrained to be smaller than a specified parameter. More precisely let $X = (x_1, x_2, \ldots, x_m)$ denote the matrix of covariates and let y denote the response, where the x_is 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 $\beta = (\beta_1, \beta_2, \ldots, \beta_m)$ are the solution to the constrained optimization problem

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

Provided that the LASSO parameter t is small enough, some of the regression coefficients will be exactly zero. 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 where 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 SELECTION=LASSO is used in PROC GLMSELECT. 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 with the other selection methods supported by PROC GLMSELECT, you can specify a criterion to choose among the models at each step of the LASSO algorithm with the CHOOSE= option. You can also specify a stopping criterion with the STOP= option. See the discussion in the "Forward Selection (FORWARD)" section on page 35 for additional details. The model degrees of freedom PROC GLMSELECT uses at any step of the LASSO is 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 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 SELECTION=LASSO.

Model Selection Issues

Many authors caution against the use of "automatic variable selection" methods and describe pitfalls that plague many such methods. For example, Harrell (2001) states that "Stepwise variable selection has been a very popular technique for many years, but if this procedure had just been proposed as a statistical method, it would most likely be rejected because it violates every principle of statistical estimation and hypothesis testing." He lists and discusses several of these issues and cites a variety of studies that highlight these problems. He also notes that many of these issues are not restricted to stepwise selection, but impact forward selection and backward elimination, as well as methods based on all-subset selection.

In their introductory chapter, Burnham and Anderson (2002) discuss many issues involved in model selection. They too strongly warn against "data dredging," which they describe as "the process of analyzing data with few or no a priori questions, by subjectively and iteratively searching the data for patterns and 'significance'." However, Burnham and Anderson also discuss the desirability of finding parsimonious models. They note that using "full models" that contain many insignificant predictors might avoid some of the inferential problems arising in models with automatically selected variables but will lead to overfitting the particular sample data and produce a model that performs poorly in predicting data not used in training the model.

One problem in the traditional implementations of forward, backward, and stepwise selection methods is that they are based on sequential testing with specified entry

(SLE) and stay (SLS) significance levels. However, it is known that the "F-to-enter" and "F-to-delete" statistics do not follow an F-distribution (Draper, Guttman, and Kanemasu 1971). Hence the SLE and SLS values cannot reliably be viewed as probabilities. One way to address this difficulty is to replace hypothesis testing as a means of selecting a model with information criteria or out-of-sample prediction criteria. While Harrell (2001) points out that information criteria were developed for comparing only prespecified models, Burnham and Anderson (2002) note that AIC criteria have routinely been used for several decades for performing model selection in time series analysis.

Problems also arise when the selected model is interpreted as if it were prespecified. There is a "selection bias" in the parameter estimates that is discussed in detail in Miller (2002). This bias occurs because a parameter is more likely to be selected if it is above its expected value than if it is below its expected value. Furthermore, because multiple comparisons are made in obtaining the selected model, the p-values obtained for the selected model are not valid. When a single best model is selected then inference is conditional on that model. Model averaging approaches provide a way to make more stable inferences based on a set of models. Methods for doing this are presented in Burnham and Anderson (2002) and Bayesian approaches are discussed in Raftery, Madigan, and Hoeting (1997).

Despite these difficulties, careful and informed use of variable selection methods still has its place in modern data analysis. For example, Foster and Stine (2004) use a modified version of stepwise selection to build a predictive model for bankruptcy from over 67,000 possible predictors and show that this yields a model whose predictions compare favorably with other recently developed data mining tools. In particular, when the goal is prediction rather than estimation or hypothesis testing, variable selection with careful use of validation to limit both under and over fitting is often a useful starting point of model development.

Criteria Used in Model Selection Methods

PROC GLMSELECT supports a variety of fit statistics that you can specify as criteria for the CHOOSE=, SELECT=, and STOP= options on the MODEL statement. The following statistics are available:

ADJRSQ	the Adjusted R-square statistic (Darlington 1968; Judge et al. 1985)
AIC	the Akaike Information Criterion (Darlington 1968; Judge et al. 1985)
AICC	the Corrected Akaike Information Criterion (Hurvich and Tsai 1991)
BIC	the Sawa Bayesian Information Criterion (Sawa 1978; Judge et al. 1985)
CP	the Mallow C(p) statistic (Mallows 1973; Hocking 1976)
PRESS	the predicted residual sum of squares statistic

SBC — the Schwarz Bayesian Information Criterion (Schwarz 1978; Judge et al. 1985)

SL — the significance level of the F statistic used to assess an effect's contribution to the fit when it is added to or removed from a model

VALIDATE | VALIDATEASE — the average square error over the validation data

Table 6. Formulas and Definitions for Model Fit Summary Statistics

MODEL Option or Statistic	Definition or Formula
n	the number of observations
p	the number of parameters including the intercept
$\hat{\sigma}^2$	the estimate of pure error variance from fitting the full model
SST	the total sum of squares corrected for the mean for the dependent variable
SSE	the error sum of squares
ASE	$\frac{\text{SSE}}{n}$
MSE	$\frac{\text{SSE}}{n-p}$
R^2	$1-\frac{\text{SSE}}{\text{SST}}$
ADJRSQ	$1-\frac{(n-1)(1-R^2)}{n-p}$
AIC	$n\ln\left(\frac{\text{SSE}}{n}\right)+2p$
AICC	$1+\ln\left(\frac{\text{SSE}}{n}\right)+\frac{2(p+1)}{n-p-2}$
BIC	$n\ln\left(\frac{\text{SSE}}{n}\right)+2(p+2)q-2q^2$ where $q=\frac{n\hat{\sigma}^2}{\text{SSE}}$
CP (C_p)	$\frac{\text{SSE}}{\hat{\sigma}^2_n}+2p-n$
PRESS	$\sum_{i=1}^{n}\frac{r_i^2}{(1-h_i)^2}$ where $r_i=$ the residual at observation i and $h_i=$ the leverage of observation $i=\mathbf{x}_i(\mathbf{X}'\mathbf{X})^{-}\mathbf{x}_i'$
RMSE	$\sqrt{\text{MSE}}$
SBC	$n\ln\left(\frac{\text{SSE}}{n}\right)+p\ln(n)$

CLASS Variable Parameterization

Consider a model with one CLASS variable A with four levels, 1, 2, 5, and 7. Details of the possible choices for the PARAM= option follow.

EFFECT Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of -1. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Effect Coding			
	Design Matrix		
A	A1	A2	A5
1	1	0	0
2	0	1	0
5	0	0	1
7	−1	−1	−1

Parameter estimates of CLASS main effects using the effect coding scheme estimate the difference in the effect of each nonreference level compared to the average effect over all four levels.

GLM As in PROC GLM, four columns are created to indicate group membership. The design matrix columns for A are as follows.

GLM Coding				
	Design Matrix			
A	A1	A2	A5	A7
1	1	0	0	0
2	0	1	0	0
5	0	0	1	0
7	0	0	0	1

Parameter estimates of CLASS main effects using the GLM coding scheme estimate the difference in the effects of each level compared to the last level.

ORDINAL

THERMOMETER Three columns are created to indicate group membership of the higher levels of the effect. For the first level of the effect (which for A is 1), all three dummy variables have a value of 0. The design matrix columns for A are as follows.

Ordinal Coding			
	Design Matrix		
A	A2	A5	A7
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

The first level of the effect is a control or baseline level. Parameter estimates of CLASS main effects using the ORDINAL coding scheme estimate the effect on the response as the ordinal factor is set to each succeeding level. When the parameters for an ordinal main effect have the same sign, the response effect is monotonic across the levels.

POLYNOMIAL

POLY

Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, ... according to their sorting order. The design matrix columns for A are as follows.

Polynomial Coding			
	Design Matrix		
A	**APOLY1**	**APOLY2**	**APOLY3**
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE

REF

Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of 0. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Reference Coding			
	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects using the reference coding scheme estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

ORTHEFFECT

The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=EFFECT. The design matrix columns for A are as follows.

Orthogonal Effect Coding			
	Design Matrix		
A	AOEFF1	AOEFF2	AOEFF3
1	1.41421	−0.81650	−0.57735
2	0.00000	1.63299	−0.57735
5	0.00000	0.00000	1.73205
7	−1.41421	−0.81649	−0.57735

ORTHORDINAL

ORTHOTHERM The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=ORDINAL. The design matrix columns for A are as follows.

Orthogonal Ordinal Coding			
	Design Matrix		
A	AOORD1	AOORD2	AOORD3
1	−1.73205	0.00000	0.00000
2	0.57735	−1.63299	0.00000
5	0.57735	0.81650	−1.41421
7	0.57735	0.81650	1.41421

ORTHPOLY The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=POLY. The design matrix columns for A are as follows.

Orthogonal Polynomial Coding			
	Design Matrix		
A	AOPOLY1	AOPOLY2	AOPOLY5
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=REFERENCE. The design matrix columns for A are as follows.

Orthogonal Reference Coding			
	Design Matrix		
A	AOREF1	AOREF2	AOREF3
1	1.73205	0.00000	0.00000
2	−0.57735	1.63299	0.00000
5	−0.57735	−0.81650	1.41421
7	−0.57735	−0.81650	−1.41421

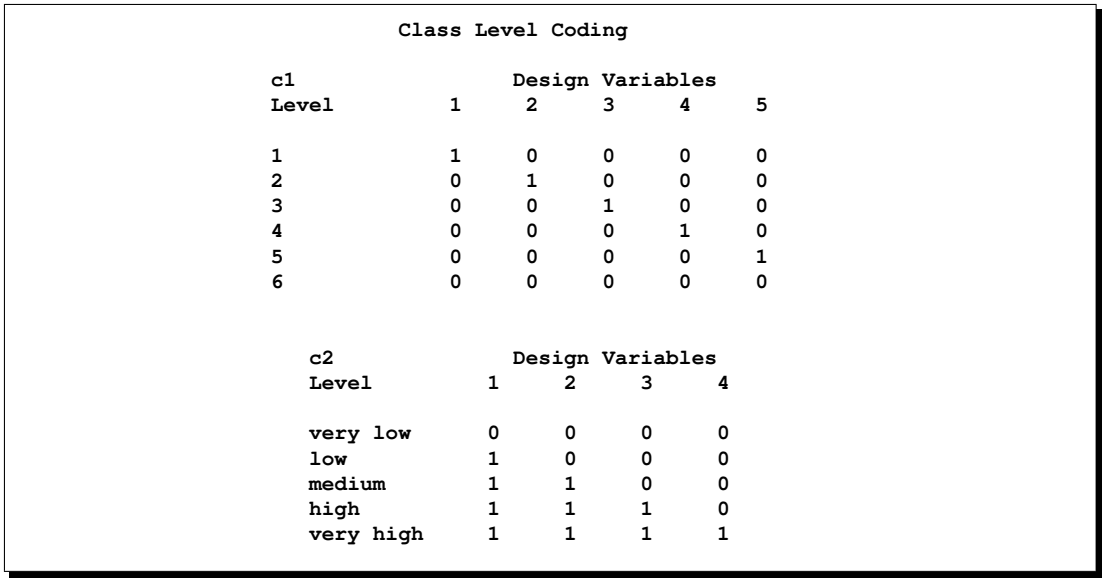
The following example illustrates several features of the CLASS statement.

```
data codingExample;
   drop i;

   do i=1 to 1000;
     c1 = 1 + mod(i,6);
     if      i < 50  then c2 = 'very low ';
     else if i < 250 then c2 = 'low';
     else if i < 500 then c2 = 'medium';
     else if i < 800 then c2 = 'high';
     else                 c2 = 'very high';
     x1 = ranuni(1);
     x2 = ranuni(1);
     y = x1 + 10*(c1=3) +5*(c1=5) +rannor(1);
     output;
   end;
run;

proc glmselect data=codingExample;
   class c1(param=ref split) c2(param=ordinal order=data) /
         delimiter = ',' showcoding;
   model y = c1 c2 x1 x2/orderselect;
run;
```

```
                    The GLMSELECT Procedure

                    Class Level Information

      Class      Levels      Values

      c1              6 *    1,2,3,4,5,6
      c2              5      very low,low,medium,high,very high

                  * Associated Parameters Split
```

Figure 11. Class Level Information

The "Class Level Information" table shown in Figure 11 is produced by default whenever you specify a CLASS statement. Note that because the levels of the variable "c2" contain embedded blanks, the DELIMITER="," option has been specified. The SHOWCODING option requests the display of the "Class Level Coding" tables shown in Figure 12. An ordinal parametrization is used for "c2" because its levels have a natural order. Furthermore, because these levels appear in their natural order in the data, you can preserve this order by specifying the ORDER=*data* option.



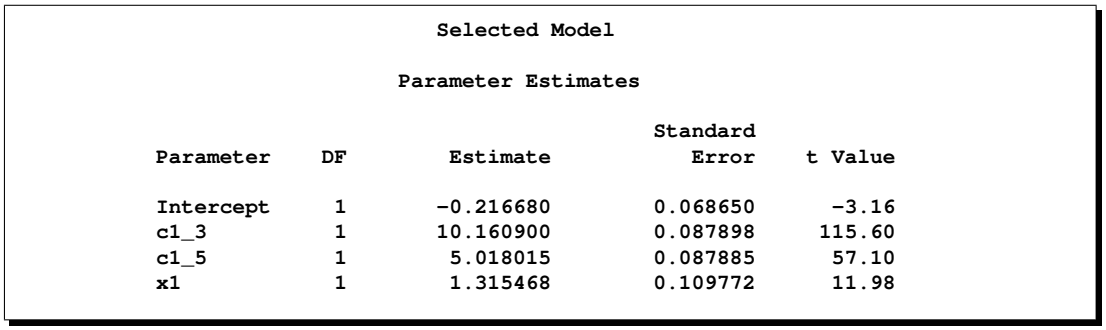
Class Level Coding

c1 Level	Design Variables 1	2	3	4	5
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1
6	0	0	0	0	0

c2 Level	Design Variables 1	2	3	4
very low	0	0	0	0
low	1	0	0	0
medium	1	1	0	0
high	1	1	1	0
very high	1	1	1	1

Figure 12. Class Level Coding

The SPLIT option has been specified for the class variable "c1." This permits the parameters associated with the effect "c1" to enter or leave the model individually. The "Parameter Estimates" table shown in Figure 13 shows that for this example the parameters corresponding to only levels 3 and 5 of "c1" are in the selected model. Finally, note that the ORDERSELECT option on the MODEL statement specifies that the parameters are displayed in the order they first entered the model.



Selected Model

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value
Intercept	1	−0.216680	0.068650	−3.16
c1_3	1	10.160900	0.087898	115.60
c1_5	1	5.018015	0.087885	57.10
x1	1	1.315468	0.109772	11.98

Figure 13. Parameter Estimates

Macro Variables Containing Selected Models

Often you may want to perform post-selection analysis using other SAS procedures. To facilitate this, PROC GLMSELECT saves the list of selected effects in a macro variable. This list does not explicitly include the intercept so that you can use it on the MODEL statement of other SAS/STAT regression procedures.

The following table describes the macro variables that PROC GLMSELECT creates. Note that when BY processing is used, one macro variable, indexed by the BY group

number, is created for each BY group.

Macro Variable	Description
No BY processing	
_GLSIND1	Selected model
BY processing	
_GLSNUMBYS	Number of BY groups
_GLSIND1	Selected model for BY group 1
_GLSIND2	Selected model for BY group 2
...	

You can use the macro variable _GLSIND as a synonym for _GLSIND1. If you do not use BY processing, _GLSNUMBYS is still defined and has the value 1.

To aid in associating indexed macro variables with the appropriate observations when BY processing is used, PROC GLMSELECT creates a variable _BY_ in the output data set specified on an OUTPUT statement (see the "OUTPUT Statement" section on page 30) that tags observations with an index that matches the index of the appropriate macro variable.

The following code creates a data set with two BY groups and runs PROC GLMSELECT to select a model for each BY group.

```
data one(drop=i j);
  array x{5} x1-x5;
  do i=1 to 1000;
    classVar = mod(i,4)+1;
    do j=1 to 5;
      x{j} = ranuni(1);
    end;
    if i<400 then do;
         byVar = 'group 1';
         y     = 3*classVar+7*x2+5*x2*x5+rannor(1);
       end;
    else do;
         byVar = 'group 2';
         y     = 2*classVar+x5+rannor(1);
       end;
    output;
  end;
run;


proc glmselect data=one;
  by     byVar;
  class  classVar;
  model  y = classVar x1|x2|x3|x4|x5 @2 /
                 selection=stepwise(stop=aicc);
  output out=glmselectOutput;
run;
```

This PROC GLMSELECT step produces three macro variables:

Macro Variable	Value	Description
_GLSNUMBYS	2	Number of BY groups
_GLSIND1	classVar x2 x2*x5	Selected model for the first BY group
_GLSIND2	classVar x5	Selected model for the second BY group

You can now leverage these macro variables and the output data set created by PROC GLMSELECT to perform post-selection analyses that match the selected models with the appropriate BY group observations. For example, the following code creates and runs a macro that uses PROC GLM to perform LSMeans analyses.

```
%macro LSMeansAnalysis;
   %do i=1 %to &_GLSNUMBYS;

      title1  "Analysis Using the Selected Model for BY group number &i";
      title2 "Selected Effects: &&_GLSIND&i";

      ods select LSMeans;
      proc glm data=glmselectOutput(where = (_BY_ = &i));
         class classVar;
         model y = &&_GLSIND&i;
         lsmeans classVar;
      run;quit;

   %end;
%mend;

%LSMeansAnalysis;
```

The LSMeans analysis output from PROC GLM is shown in Figure 14.

```
Analysis Using the Selected Model for BY group number 1
          Selected Effects: classVar x2 x2*x5

                    The GLM Procedure
                   Least Squares Means

                  class
                  Var          y LSMEAN

                  1           7.8832052
                  2          10.9528618
                  3          13.9412216
                  4          16.7929355


Analysis Using the Selected Model for BY group number 2
              Selected Effects: classVar x5

                    The GLM Procedure
                   Least Squares Means

                  class
                  Var          y LSMEAN

                  1          2.46805014
                  2          4.52102826
                  3          6.53369479
                  4          8.49354763
```

Figure 14. LS Means Analyses for Selected Models

Building the SSCP Matrix

Traditional implementations of FORWARD and STEPWISE selection methods start by computing the augmented cross-product matrix for all the specified effects. This initial cross-product matrix is updated as effects enter or leave the current model by sweeping the columns corresponding to the parameters of the entering or departing effects. Building the starting cross-product matrix can be done with a single pass through the data and requires $O(m^2)$ storage and $O(n^2m)$ work, where n is the number of observations and m is the number of parameters. If k selection steps are done then the total work sweeping effects in and out of the model is $O(km^2)$. When $n >> m$ then the work required is dominated by the time spent forming the cross-product matrix. However, when m is large (tens of thousands), then just storing the cross-product matrix becomes intractable even though the number of selected parameters may be small. Note also that when interactions of classification effects are considered, then the number of parameters considered can be large, even though the number of effects considered is much smaller.

When the number of selected parameters is smaller than the total number of parameters, it turns out that many of the cross-products are not needed in the selection process. Let y denote the dependent variable and suppose that at some step of the selection process that X denotes the $n \times p$ design matrix columns corresponding to the currently selected model. Let $Z = Z_1, Z_2, \ldots, Z_{m-p}$ denote the design matrix

columns corresponding to the $m - p$ effects not yet in the model. Then in order to compute the reduction in the residual sum of squares when Z_j is added to to the model, the only additional cross-products needed are $Z_j'y$, $Z_j'X$, and $Z_j'Z_j$. Note that it is not necessary to compute any of $Z_j'Z_i$ with $i \neq j$ and if $p << m$, and this yields a substantial saving in both memory required and computational work. Note, however, that this strategy does require a pass through the data at any step where adding an effect to the model is considered.

PROC GLMSELECT supports both of these strategies for building the cross-product matrix. You can choose which of these strategies to use by specifying the BUILDSSCP=FULL or BUILDSSCP=INCREMENTAL option on the PERFORMANCE statement. If you request BACKWARD selection, then the full SSCP matrix is required. Similarly, if you request the BIC or CP criterion as the SELECT=, CHOOSE=, or STOP= criterion, or if you request the display one or both of these criteria with the STATS=BIC, STATS=CP, or STATS=ALL option, then the full model needs to be computed. If you do not specify the BUILDSSCP= option, then PROC GLMSELECT switches to the incremental strategy if the number of effects is greater than one hundred. This default strategy is designed to give good performance when the number of selected parameters is less than about twenty percent of the total number of parameters. Hence if you choose options that you know will cause the selected model to contain a significantly higher percentage of the total number of candidate parameters, then you should consider specifying BUILDSSCP=FULL. Conversely if you specify less than 100 effects on the model statement but where many of these effects have a large number of associated parameters, then specifying BUILSSCP=INCREMENTAL might result in improved performance.

Using Validation and Test Data

When you have sufficient data, you can subdivide your data into three parts called the training, validation, and test data. During the selection process, models are fit on the training data, and the prediction error for the models so obtained is found using the validation data. This prediction error on the validation data can be used to decide when to terminate the selection process or to decide what effects to include as the selection process proceeds. Finally, once a selected model has been obtained, the test set can be used to assess how the selected model generalizes on data that played no role in determining selected model.

In some cases you may want to use only training and test data. For example, you may decide to use an information criterion to decide what effects to include and when to terminate the selection process. In this case no validation data are required but test data can still be useful in assessing the predictive performance of the selected model. In other cases you may decide to use validation data during the selection process but forgo assessing the selected model on test data. Hastie, Tibshirani, and Friedman (2001) note that it is difficult to give a general rule on how many observations you should assign to each role. They note that a typical split might be 50% for training, and 25% each for validation and testing.

PROC GLMSELECT provides several methods for partitioning data into training, validation, and test data. You can provide data for each role in separate data sets that

you specify with the DATA=, TESTDATA=, and VALDATA= options on the PROC GLMSELECT procedure. An alternative method is to use a PARTITION statement to logically subdivide the DATA= data set into separate roles. You can name the fractions of the data that you want to reserve as test data and validation data. For example, specifying

```
proc glmselect data=inData;
  partition fraction(test=0.25 validate=0.25);
  ...
run;
```

randomly subdivides the "inData" data set reserving 50% for training, and 25% each for validation and testing.

In some case you may need to exercise more control on the partitioning of the input data set. You can do this by naming a variable in the input data set as well as a formatted value of that variable that correspond to each role. For example, specifying

```
proc glmselect data=inData;
  partition roleVar=group(test='group 1' train='group 2')
  ...
run;
```

assigns all roles observations in the "inData" data set based the value of the variable named group in that data set. Observations where the value of group is 'group 1' are assigned for testing and those with value 'group 2' are assigned to training. All other observations are ignored.

You can also combine the use of the PARTITION statement with named data sets for specifying data roles. For example,

```
proc glmselect data=inData testData=inTest;
  partition fraction(validate=0.4);
  ...
run;
```

reserves 40% of the "inData" data set for validation and uses the remaining 60% for training. Data for testing is supplied in the "inTest" data set. Note that in this case, because you have supplied a TESTDATA= data set, you cannot reserve additional observations for testing with the PARTITION statement.

When you use a PARTITION statement, the output data set created with an OUTPUT statement contains a character variable _ROLE_ whose values "TRAIN," "TEST," or "VALIDATE" indicate the role of each observation. _ROLE_ is blank for observations that were not assigned to any of these three roles. When the input data set specified on theDATA= option on the PROC GLMSELECT statements contains an _ROLE_ variable and no PARTITION statement is used, and TESTDATA= and VALDATA= are not specified, then the _ROLE_ variable is used to define the roles

of each observation. This is useful when you want to rerun PROC GLMSELECT but use the same data partitioning as in a previous PROC GLMSELECT step. For example,

```
proc glmselect data=inData;
  partition fraction(test=0.5);
  model y=x1-x10/selection=forward;
  output out=outDataForward;
run;

proc glmselect data=outDataForward;
  model y=x1-x10/selection=backward;
run;
```

uses the same data for testing and training in both PROC GLMSELECT steps.

When you have reserved observations for training, validation, and testing, a model fit on the training data is scored on the validation and test data, and the average squared error, denoted by ASE, is computed separately for each of these subsets. The ASE for each data role is the error sum of squares for observations in that role divided by the number of observations in that role.

Using the Validation ASE as the STOP= Criterion

If you have provided observations for validation, then you can specify STOP=VALIDATE as a suboption on the SELECTION= option on the MODEL statement. At step k of the selection process, the best candidate effect to enter or leave the current model is determined. Note that here "best candidate" means the effect that gives the best value of the SELECT= criterion that need not be based on the validation data. The validation ASE for the model with this candidate effect added is computed. If this validation ASE is greater than the validation ASE for the model at step k, then the selection process terminates at step k.

Using the Validation ASE as the CHOOSE= Criterion

When you specify the CHOOSE=VALIDATE suboption of the SELECTION= option on the MODEL statement, then the validation ASE is computed for the models at each step of the selection process. The model at the first step yielding the smallest validation ASE is selected.

Using the Validation ASE as the SELECT= Criterion

You request the validation ASE as the selection criterion by specifying the SELECT=VALIDATE suboption of the SELECTION= option on the MODEL statement. At step k of the selection process, the validation ASE is computed for each model where a candidate for entry is added or candidate for removal is dropped. The selected candidate for entry or removal is the one that yields a model with the minimal validation ASE.

Cross Validation

Deciding when to stop a selection method is a crucial issue in performing effect selection. Predictive performance of candidate models on data not used in fitting the model is one approach supported by PROC GLMSELECT for addressing this problem (see the "Using Validation and Test Data" section on page 54). However, in some cases, you might not have sufficient data to create a sizable training set and a validation set that represent the predictive population well. In these cases, cross validation is an attractive alternative for estimating prediction error.

In k-fold cross validation, the data are split into k roughly equally sized parts. One of these parts is held out for validation and the model is fit on the remaining $k-1$ parts. This fitted model is used to compute the predicted residual sum of squares on the omitted part and this process is repeated for each of k parts. The sum of the k predicted residual sum of squares so obtained is the estimate of the prediction error that is denoted by CVPRESS. Note that computing the CVPRESS statistic for k-fold cross validation requires fitting k different models, and so the work and memory requirements increase linearly with the number of cross validation folds.

You can use CVMETHOD= option on the MODEL statement to specify the method for splitting the data into k parts. CVMETHOD=BLOCK(k) requests that the k parts be made of blocks of floor(n/k) or floor$(n/k)+1$ successive observations, where n is the number of observations. CVMETHOD=SPLIT(k) requests that parts consist of observations $\{1, k+1, 2k+1, 3k+1, \ldots\}$, $\{2, k+2, 2k+2, 3k+2, \ldots\}$, . . . , $\{k, 2k, 3k, \ldots\}$. CVMETHOD=RANDOM(k) partitions the data in random subsets each with roughly floor(n/k) observations. Finally, you can use the formatted value of an input data set variable to define the parts by specifying CVMETHOD=*variable*. This last partitioning method is useful in cases where you need to exercise extra control over how the data are partioned by taking into account factors such as important but rare observations that you want to "spread out" across the various parts.

You can request details of the CVPRESS computations by specifying the CVDETAILS= option on the MODEL statement. When you use cross validation, the output data set created with an OUTPUT statement contains an integer valued variable, _CVINDEX_, whose values indicate the subset to which an observation is assigned.

The widely used special case of n-fold cross validation when you have n observations is known as *leave-one-out* cross validation. In this case, each omitted part consists of one observation, and CVPRESS statistic can be efficiently obtained without refitting the model n times. In this case, the CVPRESS statistic is denoted simply by PRESS and is given by

$$\text{PRESS} = \sum_{i=1}^{n} \left(\frac{r_i}{1-h_i} \right)^2$$

where r_i is the residual and h_i is the leverage of the ith observation. You can request *leave-one-cross* validation by specifying PRESS instead of CV with the options SELECT=, CHOOSE=, and STOP= on the model statement. For example, if the

number of observations in the data set is one hundred, then the following two PROC GLMSELECT steps are mathematically equivalent, but the second step is computed much more efficiently.

```
proc glmselect;
    model y=x1-x10/selection=forward(stop=CV) cvMethod=split(100);
run;

proc glmselect;
    model y=x1-x10/selection=forward(stop=PRESS);
run;
```

Hastie, Tibshirani, and Friedman (2001) include a discussion about choosing the cross validation fold. They note as an estimator of true prediction error, cross validation tends to have decreasing bias but increasing variance as the number of folds increases. They recommend five- or tenfold cross validation as a good compromise. By default, PROC GLMSELECT uses CVMETHOD=RANDOM(5) for cross validation.

Using Cross Validation as the STOP= Criterion

You request cross validation as the stopping criterion by specifying the STOP=CV suboption of the SELECTION= option on the MODEL statement. At step k of the selection process, the best candidate effect to enter or leave the current model is determined. Note that here "best candidate" means the effect that gives the best value of the SELECT= criterion that need not be the CV criterion. The CVPRESS score for the model with this candidate effect added or removed is determined. If this CVPRESS score is greater than the CVPRESS score for the model at step k then the selection process terminates at step k.

Using Cross Validation as the CHOOSE= Criterion

When you specify the CHOOSE=CV suboption of the SELECTION= option on the MODEL statement, then the CVPRESS score is computed for the models at each step of the selection process. The model at the first step yielding the smallest CVPRESS score is selected.

Using Cross Validation as the SELECT= Criterion

You request cross validation as the selection criterion by specifying the SELECT=CV suboption of the SELECTION= option on the MODEL statement. At step k of the selection process, the CVPRESS score is computed for each model where a candidate for entry is added or candidate for removal is dropped. The selected candidate for entry or removal is the one that yields a model with the minimal CVPRESS score. Note that at each step of the selection process, this requires forming the CVPRESS statistic for all possible candidate models at the next step. Since forming the CVPRESS statistic for k-fold requires fitting k models, using cross validation as the selection criterion is computationally very demanding as compared to using other selection criteria.

Parallel BY Group Computation

The BY group processing in PROC GLMSELECT is multithreaded enabling parallel processing of BY groups when more than one processor is available. Before effect selection begins, PROC GLMSELECT preprocesses the entire input data set and writes the preprocessed data to one or more utility files. When you specify BY group processing and there are multiple CPUs available, then one utility file is created for each processor and each BY group is assigned to one of these utility files. Once this preprocessing phase is complete, each utility file is read and all the BY groups it contains are processed in its own thread. Finally, once model selection has been completed for each thread, then the results are displayed sequentially from a single thread.

You can control the use of threading in PROC GLMSELECT and other multithreaded procedures using the SAS system options THREADS | NOTHREADS and CPUCOUNT=. You can override these SAS system options by specifying THREADS | NOTHREADS and CPUCOUNT= on the PERFORMANCE statement. Note that if NOTHREADS is in effect, or CPUCOUNT=1, or no BY processing is used, then PROC GLMSELECT will use singly threaded code. If BY groups are to be processed in multiple threads, then the memory required is proportional to the number of threads used. PROC GLMSELECT tries to predict how much memory is required for model selection and reduces the number of threads it uses if there is insufficient memory for simultaneously processing more threads. You can find out the number of threads actually used by PROC GLMSELECT in the "Performance Settings" table that you request using the DETAILS option on the PERFORMANCE statement.

The speedup you will obtain processing BY groups in parallel depends on several factors. Firstly, if the time required perform model selection on each BY group is small, then the multithreading overhead might swamp any gains obtained by parallel processing. Another limiting factor is the I/O speed. When PROC GLMSELECT builds cross-product matrices incrementally (see the "Building the SSCP Matrix" section on page 53) then data in each BY group is read multiple times. If there is insufficient I/O bandwidth, then the parallel processing can stall as CPUs wait on I/O. Optimal results will be obtained if each of the utility files used by PROC GLMSELECT is assigned to its own I/O controller. The locations of utility files can be controlled by using the SAS system option UTILLOC which must be specified when SAS is invoked. For information about the UTILLOC option and for additional information on parallel processing in SAS, refer to the chapter on "Support for Parallel Processing" in *SAS Language Reference: Concepts.*

Displayed Output

The following sections describe the displayed output produced by PROC GLMSELECT. The output is organized into various tables, which are discussed in the order of appearance. Note that the contents of a table may change depending on the options you specify.

Model Information

The "Model Information" table displays basic information about the data sets and the settings used to control effect selection. These settings include:

- the selection method
- the criteria used to select effects, stop the selection, and choose the selected model
- the effect hierarchy enforced

For ODS purposes, the name of the "Model Information" table is "ModelInfo."

Performance Settings

The "Performance Settings" table displays settings that impact performance. These settings include whether threading is enabled and the number of CPUs available as well as the method used to build the cross-product matrices. This table is displayed only if you specify the DETAILS option on the PERFORMANCE statement. For ODS purposes, the name of the "Performance Settings" table is "PerfSettings."

Number of Observations

The "Number of Observations" table displays the number of observations read from the input data set and the number of observations used in the analysis. If you specify a FREQ statement, the table also displays the sum of frequencies read and used. If you use a PARTITION statement the table also displays the number of observations used for each data role. If you specify TESTDATA= or VALDATA= data sets on PROC GLMSELECT statement, then "Number of Observations" tables are also produced for these data sets. For ODS purposes, the name of the "Number of Observations" table is "NObs."

Class Level Information

The "Class Level Information" table lists the levels of every variable specified in the CLASS statement. For ODS purposes, the name of the "Class Level Information" table is "ClassLevelInfo"

Class Level Coding

The "Class Level Coding" table shows the coding used for variables specified in the CLASS statement. For ODS purposes, the name of the "Class Level Coding" table is "ClassLevelCoding"

Dimensions

The "Dimensions" table displays information about the number of effects and the number of parameters from which the selected model is chosen. If you use split classification variables, then this table also includes the number of effects after splitting is taken into account. For ODS purposes, the name of the "Dimensions" table is "Dimensions."

Candidates

The "Candidates" table displays the effect names and values of the criterion used to select entering or departing effects at each step of the selection process. The effects are displayed in sorted order from best to worst of the selection criterion. You request this table with the DETAILS= option on the MODEL statement. For ODS purposes, the name of the "Candidates" table is "Candidates."

Selection Summary

The "Selection Summary" table displays details about the sequence of steps of the selection process. For each step, the effect that was entered or dropped is displayed along with the statistics used to select the effect, stop the selection, and choose the selected model. You can request that additional statistics be displayed with the STATS= option on the MODEL statement. For all criteria that you can use for model selection, the steps at which the optimal values of these criteria occur are also indicated. For ODS purposes, the name of the "Selection Summary" table is "SelectionSummary."

Stop Reason

The "Stop Reason" table displays the reason why the selection stopped. To facilitate programmatic use of this table, an integer code is assigned to each reason and is included if you output this table using an ODS OUTPUT statement. The reasons and their associated codes follow:

Code	Stop Reason
1	maximum number of steps done
2	specified number of steps done
3	specified number of effects in model
4	stopping criterion at local optimum
5	model is an exact fit
6	all entering effects are linearly dependent on those in the model
7	all effects are in the model
8	all effects have been dropped
9	requested full least-squares fit completed
10	stepwise selection is cycling
11	dropping any effect does not improve the selection criterion
12	no effects are significant at the specified SLE or SLS levels
13	adding or dropping any effect does not improve the selection criterion

For ODS purposes, the name of the "Stop Reason" table is "Stop Reason."

Stop Details

The "Stop Details" table compares the optimal value of the stopping criterion at the final model with how it would change if the best candidate effect were to enter or leave the model. For ODS purposes, the name of the "Stop Details" table is "StopDetails."

Selected Effects

The "Selected Effects" table displays a string containing the list of effects in the selected model. For ODS purposes, the name of the "Selected Effects" table is "SelectedEffects."

ANOVA

The "ANOVA" table displays an analysis of variance for the selected model. This table includes

- the Source of the variation, Model for the fitted regression, Error for the residual error, and C Total for the total variation after correcting for the mean. The Uncorrected Total Variation is produced when the NOINT option is used.
- the degrees of freedom (DF) associated with the source
- the Sum of Squares for the term
- the Mean Square, the sum of squares divided by the degrees of freedom
- the F Value for testing the hypothesis that all parameters are zero except for the intercept. This is formed by dividing the mean square for Model by the mean square for Error.
- the Prob>F, the probability of getting a greater F statistic than that observed if the hypothesis is true. Note that these p-values are displayed only if you specify the "SHOWPVALUES" option on the MODEL statement. These p-values are generally liberal because they are not adjusted for the fact that the terms in the model have been selected.

You can request "ANOVA" tables for the models at each step of the selection process with the DETAILS= option on the MODEL statement. For ODS purposes, the name of the "ANOVA" table is "ANOVA."

Fit Statistics

The "Fit Statistics" table displays fit statistics for the selected model. The statistics displayed include:

- Root MSE is an estimate of the standard deviation of the error term. It is calculated as the square root of the mean square error.
- Dep Mean is the sample mean of the dependent variable.
- R-square is a measure between 0 and 1 that indicates the portion of the (corrected) total variation that is attributed to the fit rather than left to residual error. It is calculated as SS(Model) divided by SS(Total). It is also called the *coefficient of determination*. It is the square of the multiple correlation; in other words, the square of the correlation between the dependent variable and the predicted values.

- Adj R-Sq, the adjusted R^2, is a version of R^2 that has been adjusted for degrees of freedom. It is calculated as

$$\bar{R}^2 = 1 - \frac{(n-i)(1-R^2)}{n-p}$$

  where i is equal to 1 if there is an intercept and 0 otherwise; n is the number of observations used to fit the model; and p is the number of parameters in the model.
- Fit criteria AIC, AICC, BIC, CP, and PRESS if they are used in the selection process or are requested with the STATS= option. See the "Criteria Used in Model Selection Methods" section on page 44 for details and Table 6 for the formulae for evaluating these criteria.
- The CVPRESS statistic when cross validation is used in the selection process. See The "Cross Validation" section on page 57 for details.
- The average square errors (ASE) on the training, validation and test data. See the "Using Validation and Test Data" section on page 54 for details.

You can request "Fit Statistics" tables for the models at each step of the selection process with the DETAILS= option on the MODEL statement. For ODS purposes, the name of the "Fit Statistics" table is "FitStatistics."

Cross Validation Details

The "Cross Validation Details" table displays by fold

- the fold number
- the number of observations used fitting
- the number of observations omitted
- the predicted residual sum of squares on the omitted observations

You can request this table with the CVDETAILS= option on the MODEL statement whenever cross validation is used in the selection process. This table is displayed for the selected model but you can request this table at each step of the selection process using the DETAILS= option on the MODEL statement. For ODS purposes, the name of the "Cross Validation Details" table is "CVDetails."

Parameter Estimates

The "Parameter Estimates" table displays the parameters in the selected model and their estimates. The information displayed for each parameter in the selected model includes

- the parameter label that includes the effect name and level information for effects containing classification variables.

- the degrees of freedom (DF) for the parameter. There is one degree of freedom unless the model is not full rank.
- the parameter estimate
- the standard error, the estimate of the standard deviation of the parameter estimate
- T for H0: Parameter=0, the t test that the parameter is zero. This is computed as the parameter estimate divided by the standard Error.
- the Prob > |T|, the probability that a t statistic would obtain a greater absolute value than that observed given that the true parameter is zero. This is the two-tailed significance probability. Note that these p-values are displayed only if you specify the "SHOWPVALUES" option on the MODEL statement. These p-values are generally liberal because they are not adjusted for the fact that the terms in the model have been selected.

If cross validation is used in the selection process, then you can request that estimates of the parameters for each cross validation fold are included in the "Parameter Estimates" table by using the CVDETAILS= option on the MODEL statement. You can request "Parameter Estimates" tables for the models at each step of the selection process with the DETAILS= option on the MODEL statement. For ODS purposes, the name of the "Parameter Estimates" table is "ParameterEstimates."

Score Information

For each SCORE statement, the "Score Information" table displays the names of the score input and output data sets, and the number of observations that were read and that were successfully scored. For ODS purposes, the name of the "Score Information" table is "ScoreInfo."

Timing Breakdown

The "Timing Breakdown" table displays a broad breakdown of where time was spent in the PROC GLMSELECT step. This table is displayed only if you specify the DETAILS option on the PERFORMANCE statement. If multithreaded BY group processing is employed, then the number of threads used for the various phases of the computation is displayed. For ODS purposes, the name of the "Timing Breakdown" table is "Timing."

ODS Table Names

PROC GLMSELECT assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 7.

For more information on ODS, see Chapter 15, "Using the Output Delivery System." (*SAS/STAT User's Guide*)

Table 7. ODS Tables Produced by PROC GLMSELECT

ODS Table Name	Description	Statement	Option
ANOVA *	selected model ANOVA table	MODEL	default
Candidates	entry/removal effect ranking	MODEL	DETAILS=
ClassLevelCoding	classification variable coding	CLASS	SHOWCODING
ClassLevelInfo	classification variable levels	CLASS	default
CVDetails	cross validation PRESS by fold	MODEL	CVDETAILS=
Dimensions	number of effects and parameters	MODEL	default
FitStatistics *	selected model fit statistics	MODEL	default
ModelInfo	model information	MODEL	default
NObs	number of observations	MODEL	default
ParameterEstimates *	selected model parameter estimates	MODEL	default
PerfSettings	performance settings	PERFORMANCE	DETAILS
ScoreInfo	score request information	SCORE	default
SelectedEffects	list of selected effects	MODEL	default
SelectionSummary	selection summary	MODEL	default
StopDetails	stopping criterion details	MODEL	default
StopReason	reason why selection stopped	MODEL	default
Timing	timing details	PERFORMANCE	DETAILS

* You can request ANOVA, FitStatistics, and ParameterEstimates at each step with the DETAILS= option on the MODEL statement.

ODS Graph Names

PROC GLMSELECT assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 8.

To request these graphs you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*).

Table 8. ODS Graphics Produced by PROC GLMSELECT

ODS Graph Name	Plot Description	PLOTS Option
AdjRSqPlot	Adjusted R-square by step	CRITERIONPANEL(UNPACK)
AICPlot	Akaike Information Criterion by step	CRITERIONPANEL(UNPACK)
ASEPlot	Average square errors by step	ASE
BICPlot	Sawa's Bayesian Information Criterion by step	CRITERIONPANEL(UNPACK)
CandidatesPlot	SELECT criterion by effect	CANDIDATES
ChooseCriterionPlot	CHOOSE criterion by step	COEFFICIENTS(UNPACK)
CoefficientPanel	Coefficients and CHOOSE criterion by step	COEFFICIENTS

Table 8. (continued)

ODS Graph Name	Plot Description	PLOTS= Option
CoefficientPlot	Coefficients by step	COEFFICIENTS(UNPACK)
CPPlot	Mallows' C(p) by step	CRITERIONPANEL(UNPACK)
CriterionPanel	Fit criteria by step	CRITERIONPANEL
CVPRESSPlot	Cross Validation Predicted RSS by step	CRITERIONPANEL(UNPACK)
PRESSPlot	Predicted RSS by step	CRITERIONPANEL(UNPACK)
SBCPlot	Schwarz Bayesian Information Criterion by step	CRITERIONPANEL(UNPACK)
ValidateASEPlot	Average square error on validation data by step	CRITERIONPANEL(UNPACK)

Examples

Example 1. Modeling Baseball Salaries Using Performance Statistics

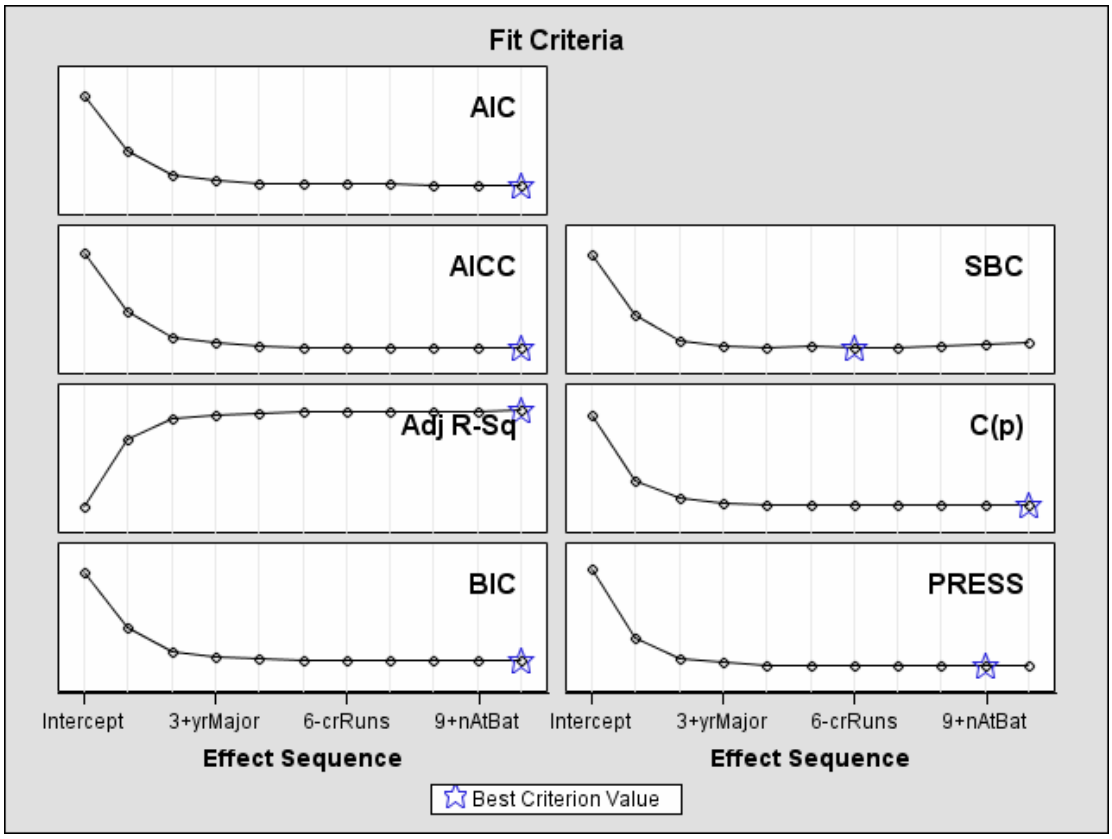
This example continues the investigation of the baseball data set introduced in the "Getting Started" section on page 4. In the Getting Started example, the default stepwise selection method based on the SBC criterion was used to select a model. In this example, model selection using other information criteria and out-of-sample prediction criteria is explored.

PROC GLMSELECT provides several selection algorithms that you can customize by specifying criteria for selecting effects, stopping the selection process, and choosing a model from the sequence of models at each step. For more details on the criteria available, see the section "Criteria Used in Model Selection Methods" on page 44. The SELECT=SL suboption of the SELECTION= option on the MODEL statement in following code requests the traditional hypothesis test-based stepwise selection approach, where effects in the model that are not significant at the stay significance level (SLS) are candidates for removal and effects not yet in the model whose addition is significant at the entry significance level (SLE) are candidates for addition to the model.

```
ods html;
ods graphics on;

proc glmselect data=baseball plot=CriterionPanel;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
                  / selection=stepwise(select=SL) stats=all;
run;

ods graphics off;
ods html close;
```

The default SLE and SLS values of 0.15 may not be appropriate for these data. One way to investigate alternative ways to stop the selection process is to assess the sequence of models in terms of model fit statistics. The STATS=ALL option on the model statement requests that all model fit statistics for assessing the sequence of models of the selection process be displayed. To help in the interpretation of the selection process, you can use graphics supported by PROC GLMSELECT. You enable these graphical displays by specifying the ODS GRAPHICS statement. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*). With ODS graphics enabled, the PLOTS=CRITRERIONPANEL option on the PROC GLMSELECT statement produces the "Criterion Panel" shown in Output 1.1.

Output 1.1. Criterion Panel



You can see in Output 1.1 that this stepwise selection selection process would stop at an earlier step if you use "Schwarz Bayesian Information Criterion" (SBC) or "Predicted Residual Sum of Squares" (PRESS) to assess the selected models as stepwise selection progresses. You can use the CHOOSE= suboption of the SELECTION= option on the MODEL statement to specify the criterion you want to use to select among the evaluated models. The following code uses the PRESS statistic to choose among the models evaluated during the stepwise selection.

```
proc glmselect data=baseball;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
                  / selection=stepwise(select=SL choose=PRESS);
run;
```

Note that the selected model is the model at step 9. By default, PROC GLMSELECT displays the selected model, ANOVA and fit statistics, and parameter estimates for the selected model. These are shown in Output 1.2.

Output 1.2. Details of Selected Model

The GLMSELECT Procedure
Selected Model

The selected model, based on PRESS, is the model at Step 9.

Effects: Intercept nAtBat nHits nBB yrMajor crHits division nOuts

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	7	124.67715	17.81102	55.07
Error	255	82.47658	0.32344	
Corrected Total	262	207.15373		

Root MSE	0.56872
Dependent Mean	5.92722
R-Square	0.6019
Adj R-Sq	0.5909
AIC	-288.98522
AICC	-0.08849
PRESS	88.55275
SBC	-260.40799

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value
Intercept	1	4.176133	0.150539	27.74
nAtBat	1	-0.001468	0.000946	-1.55
nHits	1	0.011078	0.002983	3.71
nBB	1	0.007226	0.002115	3.42
yrMajor	1	0.070056	0.018911	3.70
crHits	1	0.000247	0.000143	1.72
division East	1	0.143082	0.070972	2.02
division West	0	0	.	.
nOuts	1	0.000241	0.000134	1.81

Even though the model that is chosen to give the smallest value of the PRESS statistic is the model at step 9, the stepwise selection process continues to the step where the

stopping condition based on entry and stay significance levels is met. If you use the PRESS statistic as the stopping criterion, the stepwise selection process stops at step 9. This ability to stop at the first extremum of the criterion you specify can significantly reduce the amount of computation done, especially in the cases where you are selecting from a large number of effects. The following code requests stopping based on the PRESS statistic. The "Stop Reason" and Stop Details tables are shown in Output 1.3.

```
ods html;
ods graphics on;

proc glmselect data=baseball plot=Coefficients;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
                  / selection=stepwise(select=SL stop=PRESS);
run;

ods graphics off;
ods html close;
```

Output 1.3. Stopping Based on PRESS

The GLMSELECT Procedure

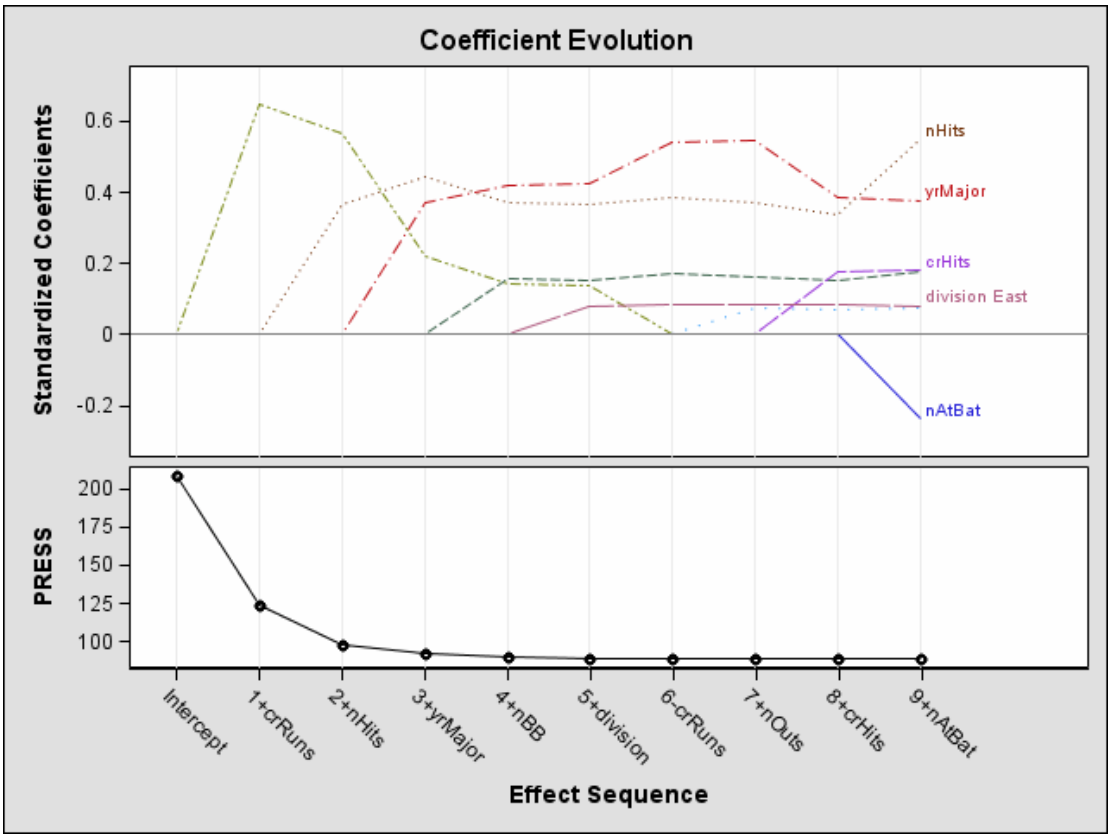
Selection stopped at a local minimum of the PRESS criterion.

Stop Details

Candidate For	Effect	Candidate PRESS		Compare PRESS
Entry	crBB	88.6321	>	88.5528
Removal	nAtBat	88.6866	>	88.5528

The PLOTS=COEFFICIENTS specification on the PROC GLMSELECT statement requests a plot that enables you to visualize the selection process.

Output 1.4. Coefficient Evolution Plot



In this plot, standardized coefficients of all the effects selected at some step of the stepwise method are plotted as a function of the step number. This enables you to assess relative importance of the effects selected at any step of the selection process as well as providing information as to when effects entered the model. The lower plot in the panel shows how the criterion used to choose the selected model changes as effects enter or leave the model.

Model selection is often done in order to obtain a parsimonious model that can be used for prediction on new data. An ever present danger is that of selecting a model that overfits the "training" data used in the fitting process, yielding a model with poor predictive performance. Using cross validation is one way to assess the predictive performance of the model. Using k-fold cross validation, the training data are subdivided into k parts and at each step of the selection process, models are obtained on each of the k subsets of the data obtained by omitting one of these parts. The cross validation predicted residual sum of squares, denoted CV PRESS, is obtained by summing the squares of the residuals when each of these submodels is scored on the data data omitted in fitting the submodel. Note that the PRESS statistic corresponds to the the special case of "leave one out" cross validation.

In the preceding example, the PRESS statistic was used to choose among models that were chosen based on entry and stay significance levels. In the following example the

SELECT=CVPRESS suboption of the SELECTION= option on the MODEL statement requests that the CV PRESS statistic itself is used as the selection criterion. The DROP=COMPETITIVE suboption requests that additions and deletions be considered simultaneously when deciding whether to add or remove an effect. At any step, the CV PRESS statistic for all models obtained by deleting one effect from the model or adding one effect to the model are computed. Among these models, the one yielding the smallest value of the CV PRESS statistic is selected and the process is repeated from this model. The stepwise selection terminates if all additions or deletions increase the CV PRESS statistic. The CVMETHOD=SPLIT(5) option requests 5-fold cross validation with the five subsets consisting of observations $\{1, 6, 11, \ldots\}$, $\{2, 7, 12, \ldots\}$, and so on.

```
ods html;
ods graphics on;

proc glmselect data=baseball plot=Candidates;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
                  / selection=stepwise(select=CV drop=competitive)
                    cvMethod=split(5);
run;

ods graphics off;
ods html close;
```

The "Selection Summary" table is shown in Output 1.5. By comparing Output 1.5 and Figure 3 you can see that the sequence of models produced is different from the sequence when the stepwise selection is based on the SBC statistic.

Output 1.5. Stepwise Selection Based on Cross Validation

The GLMSELECT Procedure

Stepwise Selection Summary

Step	Effect Entered	Effect Removed	Number Effects In	Number Parms In	CV PRESS
0	Intercept		1	1	208.9638
1	crRuns		2	2	122.5755
2	nHits		3	3	96.3949
3	yrMajor		4	4	92.2117
4	nBB		5	5	89.5242
5		crRuns	4	4	88.6917
6	league		5	5	88.0417
7	nError		6	6	87.3170
8	division		7	7	87.2147
9	nHome		8	8	87.0960*

* Optimal Value Of Criterion

If you have sufficient data, another way you can assess the predictive performance of your model is to reserve part of your data for testing your model. You score the model obtained using the training data on the test data and assess the predictive performance on this data that had no role in the selection process. You can also reserve part of your data to validate the model you obtain in the training process. Note that the validation data are not used in obtaining the coefficients of the model, but are used to decide on when to stop the selection process to limit overfitting.

PROC GLMSELECT enables you to partition your data into disjoint subsets for training validation and testing roles. This partitioning can be done using random proportions of the data or you can designate a variable in your data set that defines which observations are used for each role. See the section "PARTITION Statement" on page 32 for more details.

The following code randomly partitions the baseball data set using 50% for training, 30% for validation, and 20% for testing. The model selected at each step is scored on the validation data and the average residual sums of squares (ASE) is evaluated. The model yielding the lowest ASE on the validation data is selected. The ASE on the test data is also evaluated, but this data plays no role the selection process. Note that a seed for the pseudo-random number generator is specified on the PROC GLMSELECT statement.

```
ods html;
ods graphics on;

proc glmselect data=baseball plots=(CriterionPanel ASE) seed=1;
  partition fraction(validate=0.3 test=0.2);
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                yrMajor crAtBat crHits crHome crRuns crRbi
                crBB league division nOuts nAssts nError
/ selection=forward(choose=validate stop=10);
run;


ods graphics off;
ods html close;
```

Output 1.6. Number of Observations Table

The GLMSELECT Procedure

Number of Observations Read	322
Number of Observations Used	263
Number of Observations Used for Training	132
Number of Observations Used for Validation	80
Number of Observations Used for Testing	51

Output 1.6 shows the "Number of Observations" table. You can see that of the 263 observations that were used in the analysis, 132 (50.2%) observations were used for model training, 80 (30.4%) for model validation, and 51 (19.4%) for model testing.

Output 1.7. Selection Summary and Stop Reason

Forward Selection Summary

Step	Effect Entered	Number Effects In	Number Parms In	SBC	ASE
0	Intercept	1	1	-30.8531	0.7628
1	crRuns	2	2	-93.9367	0.4558
2	nHits	3	3	-126.2647	0.3439
3	yrMajor	4	4	-128.7570	0.3252
4	nBB	5	5	-132.2409*	0.3052
5	division	6	6	-130.7794	0.2974
6	nOuts	7	7	-128.5897	0.2914
7	nRBI	8	8	-125.7825	0.2868
8	nHome	9	9	-124.7709	0.2786
9	nAtBat	10	10	-121.3767	0.2754

* Optimal Value Of Criterion

Forward Selection Summary

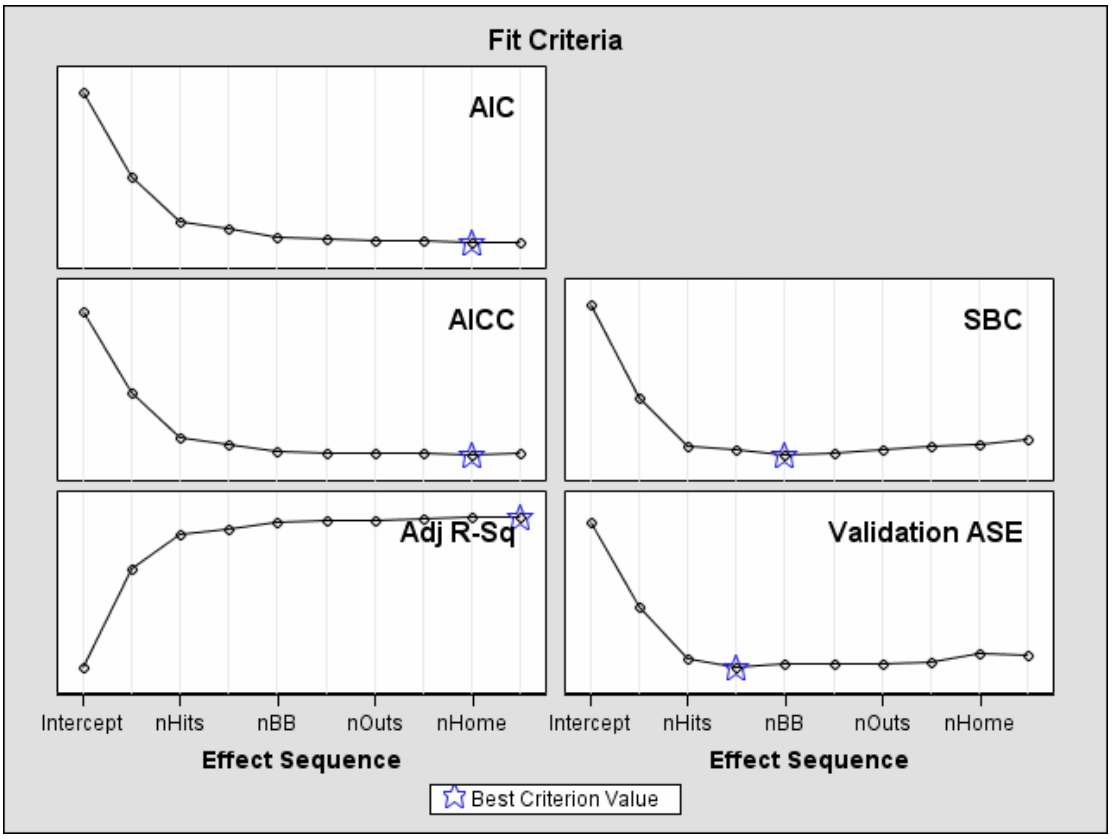
Step	Effect Entered	Validation ASE	Test ASE
0	Intercept	0.7843	0.8818
1	crRuns	0.4947	0.4210
2	nHits	0.3248	0.4697
3	yrMajor	0.2920*	0.4614
4	nBB	0.3065	0.4297
5	division	0.3050	0.4218
6	nOuts	0.3028	0.4186
7	nRBI	0.3097	0.4489
8	nHome	0.3383	0.4533
9	nAtBat	0.3337	0.4580

* Optimal Value Of Criterion

Selection stopped at the first model containing the specified number of effects (10).

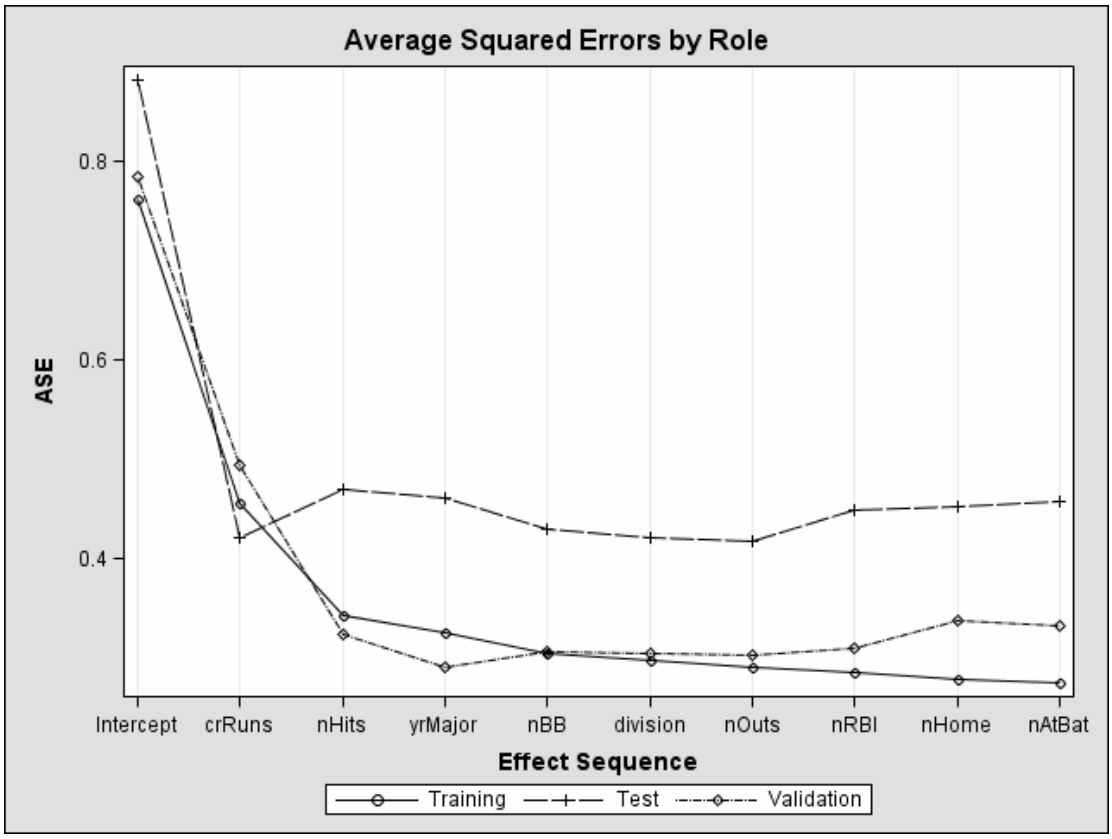
Output 1.7 shows the "Selection Summary" table and the "Stop Reason" table. The forward selection stops at step 9 as this model at this step contains 10 effects and so satisfies the stopping criterion requested with the "STOP=10" suboption. However, the selected model is the model at step 3, where the validation ASE, the CHOOSE= criterion, achieves its minimum.

Output 1.8. Criterion Panel



The "Criterion Panel" in Output 1.8 shows how the various criteria evolved as the stepwise selection method proceeded. Note that other than the ASE evaluated on the validation data, these criteria are evaluated on the training data.

Output 1.9. Average Square Errors by Role



Finally, the "ASE plot" shown in Output 1.9 shows how the average square error evolves on the training, validation, and test data. Note that while the ASE on the training data continued decreasing as the selection steps proceeded, the ASE on the test and validation data behave more erratically.

LASSO selection, pioneered by Tibshirani (1996), is a constrained least-squares method that can be viewed as a stepwise-like method where effects enter and leave the model sequentially. You can find additional details about the LASSO method in the section "Lasso Selection (LASSO)" on page 42. Note that when classification effects are used with LASSO, the design matrix columns for all effects containing classification variables can enter or leave the model individually. The following code performs LASSO selection for the baseball data. The "Selection Summary" table is shown in Output 1.10.

```
ods html;
ods graphics on;

proc glmselect data=baseball plot=CriterionPanel;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
```

```
                / selection=lasso(choose=CP steps=20);
run;

ods graphics off;
ods html close;
```

Output 1.10. Selection Summary for LASSO Selection

The GLMSELECT Procedure

LASSO Selection Summary

Step	Effect Entered	Effect Removed	Number Effects In	CP
0	Intercept		1	375.9275
1	crRuns		2	328.6492
2	crHits		3	239.5392
3	nHits		4	134.0374
4	nBB		5	111.6638
5	crRbi		6	81.7296
6	yrMajor		7	75.0428
7	nRBI		8	30.4494
8	division_East		9	29.9913
9	nOuts		10	25.1656
10		crRuns	9	18.7295
11		crRbi	8	15.1683
12	nError		9	16.6233
13	nHome		10	16.3741
14	league_National		11	14.8794
15		nRBI	10	8.8477*
16	crBB		11	9.2242
17	crRuns		12	10.7608
18	nAtBat		13	11.6266
19	nAssts		14	11.8572
20	crAtBat		15	13.4020

* Optimal Value Of Criterion

LASSO Selection Summary

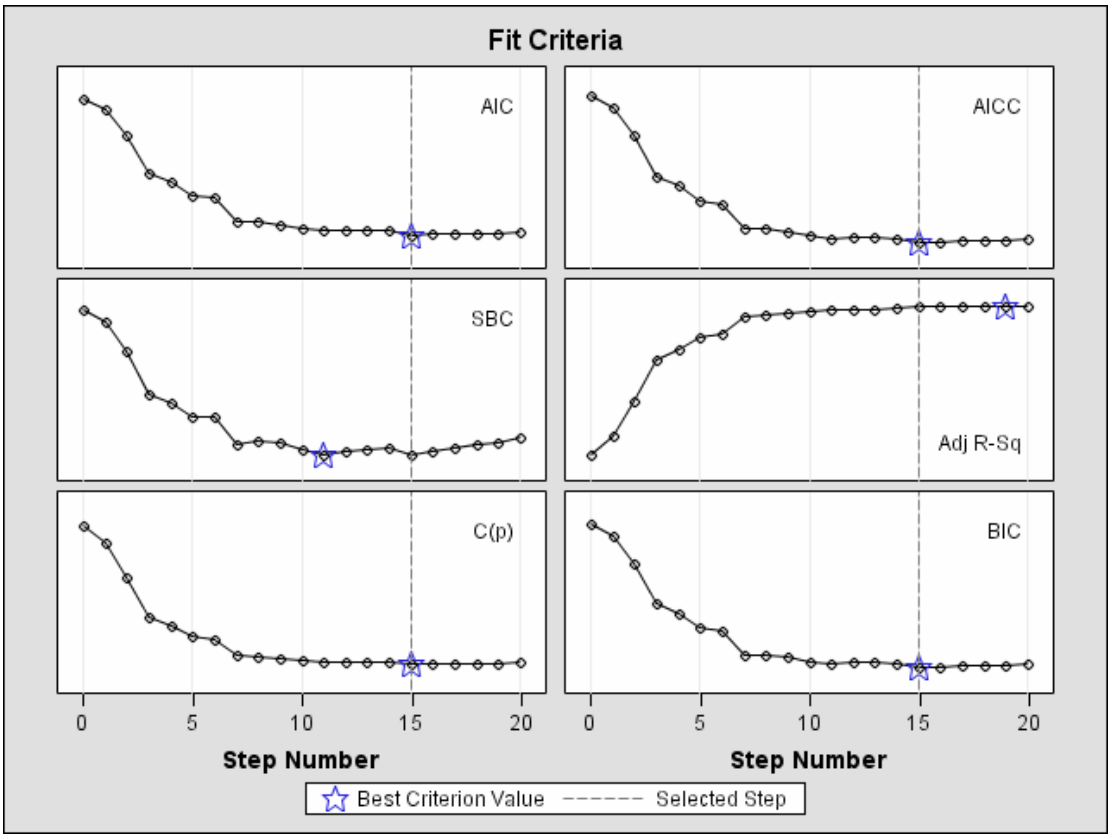
Step	Effect Entered	Effect Removed	SBC
0	Intercept		-57.2041
1	crRuns		-72.8102
2	crHits		-111.5450
3	nHits		-170.1414
4	nBB		-181.5853
5	crRbi		-200.1160
6	yrMajor		-201.5014
7	nRBI		-236.7565
8	division_East		-233.5248
9	nOuts		-234.5631
10		crRuns	-244.5223
11		crRbi	-251.6560*
12	nError		-246.6311
13	nHome		-243.3298
14	league_National		-241.3253
15		nRBI	-251.0743
16	crBB		-247.2030
17	crRuns		-242.1184
18	nAtBat		-237.7433
19	nAssts		-234.0495
20	crAtBat		-228.9628

* Optimal Value Of Criterion

Selection stopped at the specified number of steps (20).

Note that effects enter and leave sequentially. In this example, the STEPS= suboption of the SELECTION= option specifies that 20 steps of LASSO selection are done. You can see how the various model fit statistics evolved in Output 1.11.

Output 1.11. Criterion Panel



The CHOOSE=CP suboption specifies that the selected model is the model at step 15 that yields the optimal value of Mallow's C(p) statistic. Details of this selected model are shown in Output 1.12

Output 1.12. Selected Model

Selected Model

The selected model, based on C(p), is the model at Step 15.

Effects: Intercept nHits nHome nBB yrMajor crHits league_National division_East nOuts nError

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	9	125.24302	13.91589	42.98
Error	253	81.91071	0.32376	
Corrected Total	262	207.15373		

Root MSE	0.56900
Dependent Mean	5.92722
R-Square	0.6046
Adj R-Sq	0.5905
AIC	-286.79589
AICC	-0.07887
BIC	-283.91417
C(p)	8.84767
SBC	-251.07435

Parameter Estimates

Parameter	DF	Estimate
Intercept	1	4.124629
nHits	1	0.006942
nHome	1	0.002785
nBB	1	0.005727
yrMajor	1	0.067054
crHits	1	0.000249
league_National	1	0.079607
division_East	1	0.134723
nOuts	1	0.000183
nError	1	-0.007213

Example 2. Using Validation and Cross Validation

This example shows how you can use both test set and cross validation to monitor and control variable selection. It also demonstrates the use of split classification variables.

The following code produces analysis and test data sets. Note that the same code is used to generate the observations that are randomly assigned for analysis and test roles in the ratio of approximately two to one.

```
data analysisData testData;
   drop i j c3Num;
   length c3$ 7;
```

```
   array x{20} x1-x20;

   do i=1 to 1500;
      do j=1 to 20;
         x{j} = ranuni(1);
      end;

      c1 = 1 + mod(i,8);
      c2 = ranbin(1,3,.6);

      if      i < 50   then do; c3 = 'tiny';     c3Num=1;end;
      else if i < 250  then do; c3 = 'small';    c3Num=1;end;
      else if i < 600  then do; c3 = 'average';  c3Num=2;end;
      else if i < 1200 then do; c3 = 'big';      c3Num=3;end;
      else                  do; c3 = 'huge';     c3Num=5;end;

      y = 10 + x1 + 2*x5 + 3*x10 + 4*x20  + 3*x1*x7 + 8*x6*x7
             + 5*(c1=3)*c3Num + 8*(c1=7)  + 5*rannor(1);

      if ranuni(1) < 2/3 then output analysisData;
                         else output testData;
  end;
run;
```

Suppose you suspect that the dependent variable depends on both main effects and two-way interactions. You can use the following code to select a model.

```
ods html;
ods graphics on;

proc glmselect data=analysisData testdata=testData
               seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot);

   partition fraction(validate=0.5);

   class c1 c2 c3(order=data);

   model y =  c1|c2|c3|x1|x2|x3|x4|x5|x5|x6|x7|x8|x9|x10
             |x11|x12|x13|x14|x15|x16|x17|x18|x19|x20 @2
           / selection=stepwise(choose = validate
                                select = sl)
             hierarchy=single stb;
run;

ods graphics off;
ods html close;
```

Note that a TESTDATA= data set is named on the PROC GLMSELECT statement and that a PARTITION statement is used to randomly assign half the observations in the analysis data set for model validation and the rest for model training. You find details about the number of observation used for each role in the "Number of Observations" tables shown in Output 2.1

Output 2.1. Number of Observations Tables

```
                    The GLMSELECT Procedure

            Observation Profile for Analysis Data

Number of Observations Read                          1010
Number of Observations Used                          1010
Number of Observations Used for Training              510
Number of Observations Used for Validation            500


              Observation Profile for Test Data

          Number of Observations Read          490
          Number of Observations Used          490
```

Output 2.2. Class Level Information and Problem Dimensions

```
                  Class Level Information

Class       Levels    Values

c1               8    1 2 3 4 5 6 7 8
c2               4    0 1 2 3
c3               5    tiny small average big huge


                     Dimensions

                Number of Effects       278
                Number of Parameters    661
```

The "Class Level Information" and "Dimensions" table are shown in Output 2.2. The "Dimensions" table shows that at each step of the selection process, 278 effects are considered as candidates for entry or removal. Since several of these effects have multilevel classification variables as members, there are 661 parameters.

The model statement options request stepwise selection with the default entry and stay significance levels used for both selecting entering and departing effects and stopping the selection method. The CHOOSE=VALIDATE suboption specifies that the selected model is chosen to minimize the predicted residual sum of squares when the models at each step are scored on the observations reserved for validation. The HIERARCHY=SINGLE option specifies that interactions can enter the model only if the corresponding main effects are already in the model, and that main effects cannot be dropped from the model if an interaction with such an effect is in the model. These settings are listed in the "Model Information" table shown in Output 2.3

Output 2.3. Model Information

Data Set	WORK.ANALYSISDATA
Test Data Set	WORK.TESTDATA
Dependent Variable	y
Selection Method	Stepwise
Select Criterion	Significance Level
Stop Criterion	Significance Level
Choose Criterion	Validation ASE
Entry Significance Level (SLE)	0.15
Stay Significance Level (SLS)	0.15
Effect Hierarchy Enforced	Single
Random Number Seed	1

The "Stop Reason" and "Stop Details" tables are shown in Output 2.4. Note that because the STOP= suboption of the SELECTION= option was not explicitly specified, the stopping criterion used is the selection criterion, namely significance level.

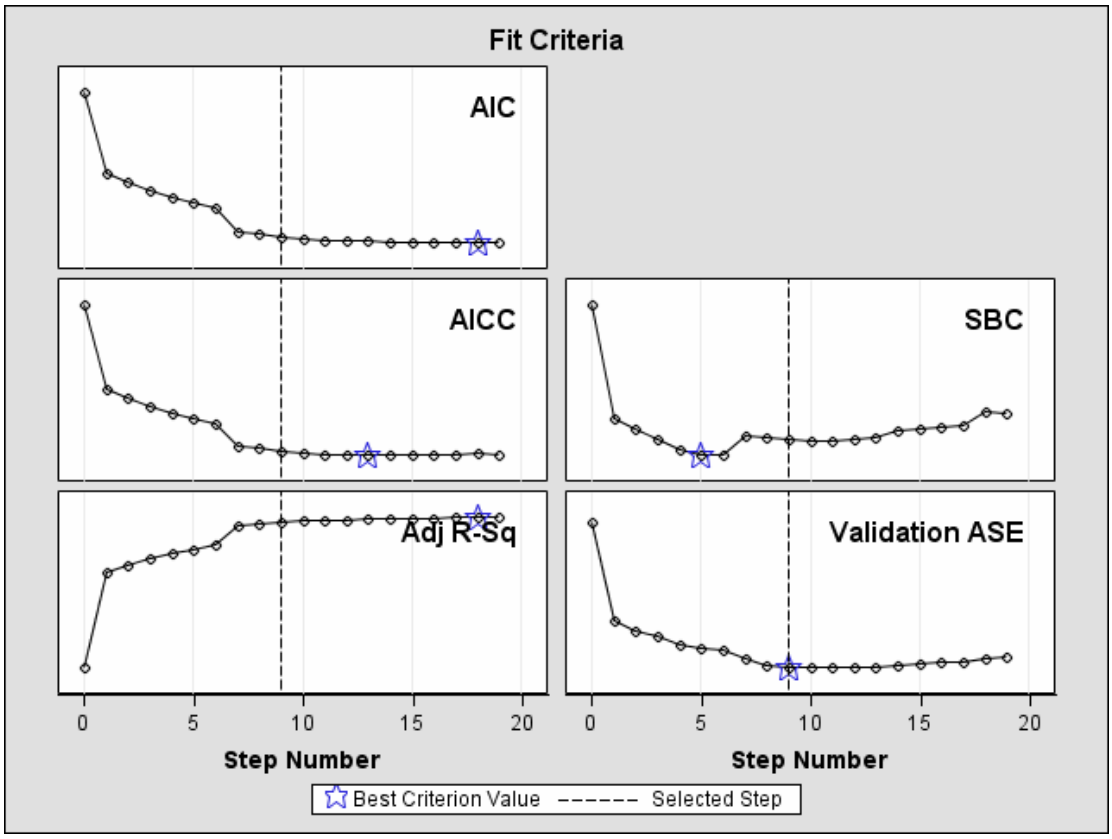
Output 2.4. Stop Details

Selection stopped because the candidate for entry has SLE > 0.15 and the candidate for removal has SLS < 0.15.

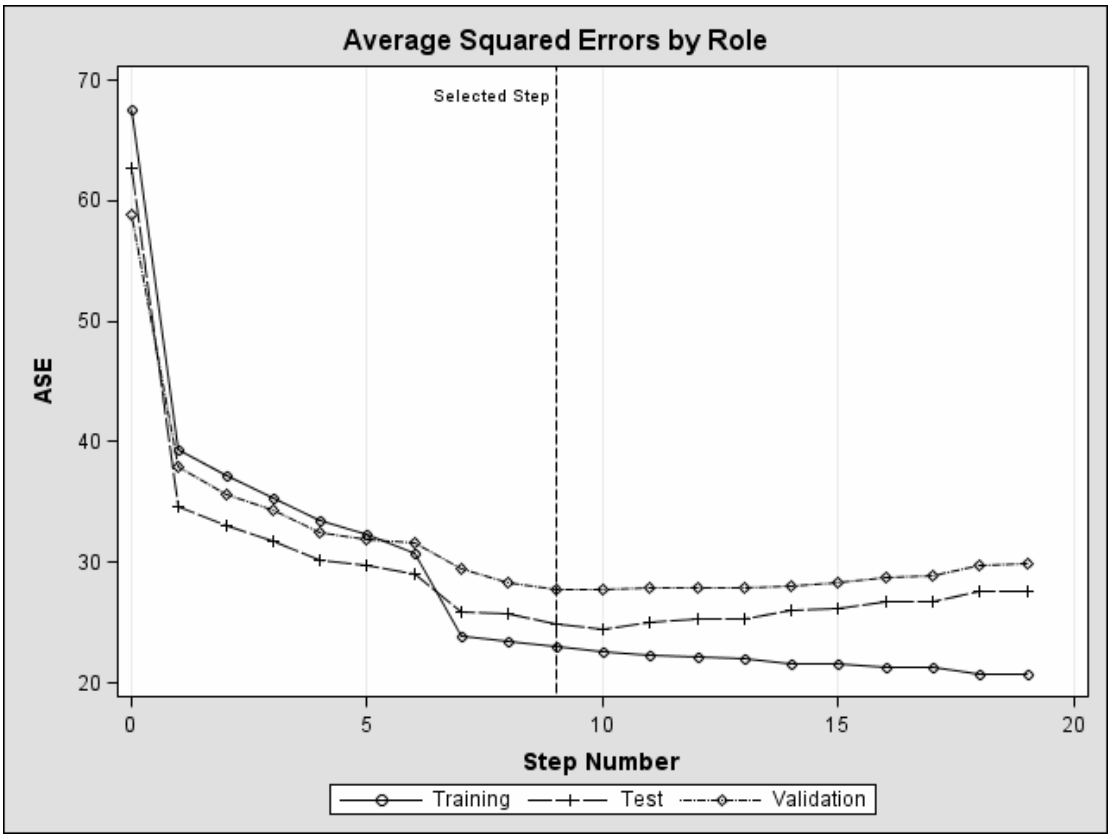
Stop Details

Candidate For	Effect	Candidate Significance		Compare Significance	
Entry	x2*x5	0.1742	>	0.1500	(SLE)
Removal	x5*x10	0.0534	<	0.1500	(SLS)

Output 2.5. Criterion Panel



Output 2.6. Average Squared Errors



The "Criterion Panel" in Output 2.5 shows how the various fit criteria evolved as the stepwise selection method proceeded. Note that other than the ASE evaluated on the validation data, these criteria are evaluated on the training data. You see that the minimum of the validation ASE occurs at step 9 and hence the model at this step is selected. Output 2.6 shows how the average squared error (ASE) evolved on the training, validation, and test data. Note that while the ASE on the training data decreases monotonically, the errors on both the validation and test data start increasing beyond step nine. This indicates that models after step 9 are beginning to overfit the training data.

Output 2.7. Selected Model Details

Selected Model

The selected model, based on Validation ASE, is the model at Step 9.

Effects: Intercept c1 c3 c1*c3 x1 x5 x6 x7 x10 x20

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	44	22723	516.43621	20.49
Error	465	11722	25.20856	
Corrected Total	509	34445		

Root MSE	5.02081
Dependent Mean	21.09705
R-Square	0.6597
Adj R-Sq	0.6275
AIC	1688.75319
AICC	4.33351
SBC	1879.30167
ASE (Train)	22.98427
ASE (Validate)	27.71105
ASE (Test)	24.82947

Output 2.8. Parameter Estimates

Selected Model

Parameter Estimates

Parameter		DF	Estimate	Standardized Estimate	Standard Error	t Value
Intercept		1	6.867831	0	1.524446	4.51
c1	1	1	0.226602	0.008272	2.022069	0.11
c1	2	1	-1.189623	-0.048587	1.687644	-0.70
c1	3	1	25.968930	1.080808	1.693593	15.33
c1	4	1	1.431767	0.054892	1.903011	0.75
c1	5	1	1.972622	0.073854	1.664189	1.19
c1	6	1	-0.094796	-0.004063	1.898700	-0.05
c1	7	1	5.971432	0.250037	1.846102	3.23
c1	8	0	0	0	.	.
c3	tiny	1	-2.919282	-0.072169	2.756295	-1.06
c3	small	1	-4.635843	-0.184338	2.218541	-2.09
c3	average	1	0.736805	0.038247	1.793059	0.41
c3	big	1	-1.078463	-0.063580	1.518927	-0.71
c3	huge	0	0	0	.	.
c1*c3	1 tiny	1	-2.449964	-0.018632	4.829146	-0.51
c1*c3	1 small	1	5.265031	0.069078	3.470382	1.52
c1*c3	1 average	1	-3.489735	-0.064365	2.850381	-1.22
c1*c3	1 big	1	0.725263	0.017929	2.516502	0.29
c1*c3	1 huge	0	0	0	.	.
c1*c3	2 tiny	1	5.455122	0.050760	4.209507	1.30
c1*c3	2 small	1	7.439196	0.131499	2.982411	2.49
c1*c3	2 average	1	-0.739606	-0.014705	2.568876	-0.29
c1*c3	2 big	1	3.179351	0.078598	2.247611	1.41
c1*c3	2 huge	0	0	0	.	.
c1*c3	3 tiny	1	-19.266847	-0.230989	3.784029	-5.09
c1*c3	3 small	1	-15.578909	-0.204399	3.266216	-4.77
c1*c3	3 average	1	-18.119398	-0.395770	2.529578	-7.16
c1*c3	3 big	1	-10.650012	-0.279796	2.205331	-4.83
c1*c3	3 huge	0	0	0	.	.
c1*c3	4 tiny	0	0	0	.	.
c1*c3	4 small	1	4.432753	0.047581	3.677008	1.21
c1*c3	4 average	1	-3.976295	-0.091632	2.625564	-1.51
c1*c3	4 big	1	-1.306998	-0.033003	2.401064	-0.54
c1*c3	4 huge	0	0	0	.	.
c1*c3	5 tiny	1	6.714186	0.062475	4.199457	1.60
c1*c3	5 small	1	1.565637	0.022165	3.182856	0.49
c1*c3	5 average	1	-4.286085	-0.068668	2.749142	-1.56
c1*c3	5 big	1	-2.046468	-0.045949	2.282735	-0.90
c1*c3	5 huge	0	0	0	.	.
c1*c3	6 tiny	1	5.135111	0.039052	4.754845	1.08
c1*c3	6 small	1	4.442898	0.081945	3.079524	1.44
c1*c3	6 average	1	-2.287870	-0.056559	2.601384	-0.88
c1*c3	6 big	1	1.598086	0.043542	2.354326	0.68
c1*c3	6 huge	0	0	0	.	.
c1*c3	7 tiny	1	1.108451	0.010314	4.267509	0.26
c1*c3	7 small	1	7.441059	0.119214	3.135404	2.37
c1*c3	7 average	1	1.796483	0.038106	2.630570	0.68
c1*c3	7 big	1	3.324160	0.095173	2.303369	1.44
c1*c3	7 huge	0	0	0	.	.
c1*c3	8 tiny	0	0	0	.	.
c1*c3	8 small	0	0	0	.	.
c1*c3	8 average	0	0	0	.	.
c1*c3	8 big	0	0	0	.	.
c1*c3	8 huge	0	0	0	.	.
x1		1	2.713527	0.091530	0.836942	3.24
x5		1	2.810341	0.098303	0.816290	3.44
x6		1	4.837022	0.167394	0.810402	5.97
x7		1	5.844394	0.207035	0.793775	7.36
x10		1	2.463916	0.087712	0.794599	3.10
x20		1	4.385924	0.156155	0.787766	5.57

Output 2.7 shows the "Selected Effects," "ANOVA," and "Fit Statistics" tables for the selected model. Output 2.8 shows the "Parameter Estimates" table. The magnitudes of the standardized estimates and the t statistics of the parameters of the effect "c1" reveal that only levels "3" and "7" of this effect contribute appreciably to the model. This suggests that a more parsimonious model with similar or better predictive power might be obtained if parameters corresponding to the levels of "c1" are allowed to enter or leave the model independently. You request this with the SPLIT option on the CLASS statement as shown in the following code:

```
ods html;
ods graphics on;

proc glmselect data=analysisData testdata=testData
               seed=1 plots(stepAxis=number)=all;

   partition fraction(validate=0.5);

   class c1(split) c2 c3(order=data);

   model y =  c1|c2|c3|x1|x2|x3|x4|x5|x5|x6|x7|x8|x9|x10
             |x11|x12|x13|x14|x15|x16|x17|x18|x19|x20 @2
           / selection=stepwise(stop   = validate
                                select = sl)
             hierarchy=single;

   output out=outData;
run;

ods graphics off;
ods html close;
```

Output 2.9. Class Level Information and Problem Dimensions

```
                    The GLMSELECT Procedure

                    Class Level Information

          Class      Levels      Values

          c1              8 *    1 2 3 4 5 6 7 8
          c2              4      0 1 2 3
          c3              5      tiny small average big huge

                    * Associated Parameters Split


                          Dimensions

                Number of Effects                278
                Number of Effects after Splits   439
                Number of Parameters             661
```

The "Class Level Information" and "Dimensions" table are shown in Output 2.9. The "Dimensions" table shows that while the model statement specifies 278 effects, after splitting the parameters corresponding to the levels of "c1", there are 439 split effects that are considered for entry or removal at each step of the selection process. Note that the total number of parameters considered is not affected by the split option.

Output 2.10. Stop Details

Selection stopped at a local minimum of the residual sum of squares of the validation data.

Stop Details

Candidate For	Effect	Candidate Validation ASE		Compare Validation ASE
Entry	x18	25.9851	>	25.7462
Removal	x6*x7	25.7611	>	25.7462

The "Stop Reason" and "Stop Details" tables are shown in Output 2.10. Since the validation ASE is specified as the stopping criterion, the selection stops at step 11 where the validation ASE achieves a local minimum and the model at this step is the selected model.

Output 2.11. Details of the Selected Model

Selected Model

The selected model is the model at the last step (Step 11).

Effects: Intercept c1_3 c1_7 c3 c1_3*c3 x1 x5 x6 x7 x6*x7 x10 x20

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	17	22111	1300.63200	51.88
Error	492	12334	25.06998	
Corrected Total	509	34445		

Root MSE	5.00699
Dependent Mean	21.09705
R-Square	0.6419
Adj R-Sq	0.6295
AIC	1660.72685
AICC	4.26329
SBC	1736.94624
ASE (Train)	24.18515
ASE (Validate)	25.74617
ASE (Test)	22.57297

You find details of the selected model in Output 2.11. The list of selected effects confirm that parameters corresponding to levels "3" and "7" only of "c1" are in the selected model. Notice that the selected model with classification variable "c1" split contains 18 parameters whereas the selected model without splitting "c1" has 45 parameters. Furthermore, by comparing the "Fit Statistics" tables in Output 2.7 and Output 2.11, you see that this more parsimonious model has smaller prediction errors on both the validation and test data.

When you use a PARTITION statement to subdivide the analysis data set, an output data set created with the OUTPUT statement contains a variable named "_ROLE_" that shows the role each observation was assigned to. See the "OUTPUT Statement" section on page 30 and the "Using Validation and Test Data" section on page 54 for additional details. The first five observations of the output data set created by the preceding code are shown in Output 2.12.

Output 2.12. Output Data Set with _ROLE_ Variable

Obs	c3	x1	x2	x3	x4	x5	x6	x7	x8
1	tiny	0.18496	0.97009	0.39982	0.25940	0.92160	0.96928	0.54298	0.53169
2	tiny	0.47579	0.84499	0.63452	0.59036	0.58258	0.37701	0.72836	0.50660
3	tiny	0.51132	0.43320	0.17611	0.66504	0.40482	0.12455	0.45349	0.19955
4	tiny	0.42071	0.07174	0.35849	0.71143	0.18985	0.14797	0.56184	0.27011
5	tiny	0.42137	0.03798	0.27081	0.42773	0.82010	0.84345	0.87691	0.26722

Obs	x9	x10	x11	x12	x13	x14	x15	x16	x17
1	0.04979	0.06657	0.81932	0.52387	0.85339	0.06718	0.95702	0.29719	0.27261
2	0.93121	0.92912	0.58966	0.29722	0.39104	0.47243	0.67953	0.16809	0.16653
3	0.57484	0.73847	0.43981	0.04937	0.52238	0.34337	0.02271	0.71289	0.93706
4	0.32520	0.56918	0.04259	0.43921	0.91744	0.52584	0.73182	0.90522	0.57600
5	0.30602	0.39705	0.34905	0.76593	0.54340	0.61257	0.55291	0.73591	0.37186

Obs	x18	x19	x20	c1	c2	y	_ROLE_	p_y
1	0.68993	0.97676	0.22651	2	1	11.4391	VALIDATE	18.5069
2	0.87110	0.29879	0.93464	3	1	31.4596	TRAIN	26.2188
3	0.44599	0.94694	0.71290	4	3	16.4294	VALIDATE	17.0979
4	0.18794	0.33133	0.69887	5	3	15.4815	VALIDATE	16.1567
5	0.64565	0.55718	0.87504	6	2	26.0023	TRAIN	24.6358

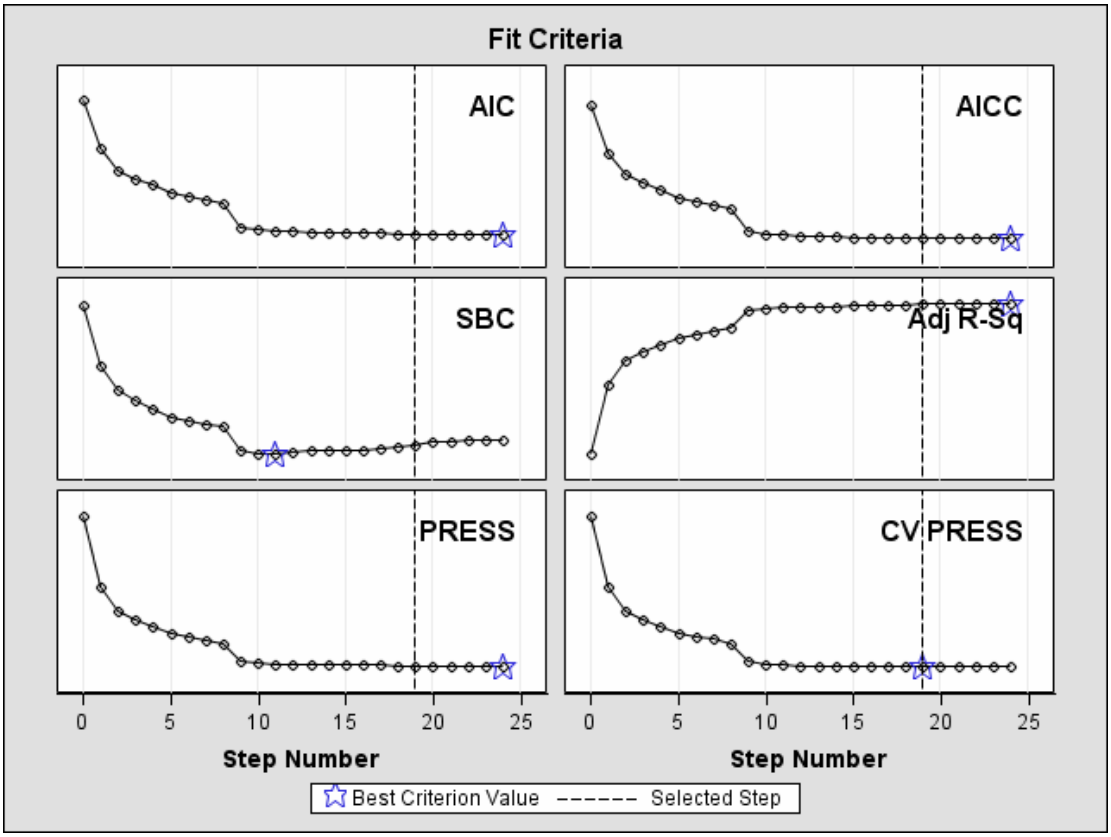
Cross validation is often used to assess the predictive performance of a model, especially for when you do not have enough observations for test set validation. See the "Cross Validation" section on page 57 for further details. The following code provides an example where cross validation is used as the CHOOSE= criterion.

```
ods html;
ods graphics on;

proc glmselect data=analysisData testdata=testData
               plots(stepAxis=number)=(criterionPanel ASEPlot);

   class c1(split) c2 c3(order=data);
```

```
   model y =  c1|c2|c3|x1|x2|x3|x4|x5|x5|x6|x7|x8|x9|x10
             |x11|x12|x13|x14|x15|x16|x17|x18|x19|x20 @2
           / selection = stepwise(choose = cv
                                  select = sl)
             stats     = press
             cvMethod  = split(5)
             cvDetails = all
             hierarchy = single;

    output out=outData;
run;

ods graphics off;
ods html close;
```

The CVMETHOD=SPLIT(5) option on the model statement requests 5-fold cross validation with the five subsets consisting of observations $\{1, 6, 11, \ldots\}$, $\{2, 7, 12, \ldots\}$, and so on. The STATS=PRESS option requests that the leave-one-out cross validation predicted residual sum of squares (PRESS) is also computed and displayed at each step, even though this statistic is not used in the selection process.

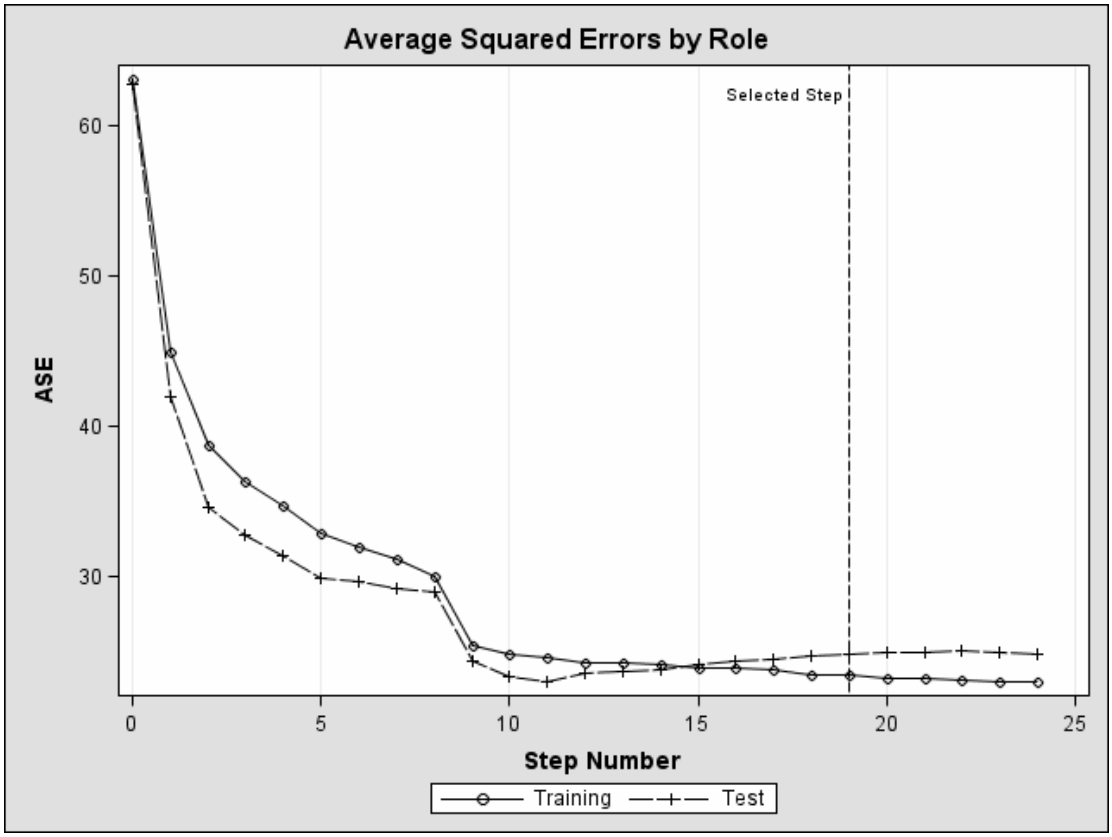
Output 2.13. Criterion Panel



Output 2.13 shows how several fit statistics evolved as the selection process pro-

gressed. The 5-fold CV PRESS statistic achieves its minimum at step 19. Note that this gives a larger model than was selected when the stopping criterion was determined using validation data. Furthermore, you see that the PRESS statistic has not achieved its minimum within 25 steps, and so an even larger model would have been selected based on "leave-one-out" cross validation.

Output 2.14. Average Squared Error Plot



Output 2.14 shows how the average squared errors compares on the test and training data. Note that the ASE error on the test data achieves a local minimum at step 11 and is already slowly increasing at step 19, which corresponds to the selected model.

Output 2.15. Breakdown of CV Press Statistic by Fold

The GLMSELECT Procedure
Selected Model

Cross Validation Details

Index	---Observations--- Fitted	Left Out	CV PRESS
1	808	202	5059.7375
2	808	202	4278.9115
3	808	202	5598.0354
4	808	202	4950.1750
5	808	202	5528.1846
Total			25293.5024

Output 2.16. Cross Validation Parameter Estimates

Parameter Estimates

Parameter		Cross Validation Estimates 1	2	3	4	5
Intercept		10.7617	10.1200	9.0254	13.4164	12.3352
c1_3		28.2715	27.2977	27.0696	28.6835	27.8070
c1_7		7.6530	7.6445	7.9257	7.4217	7.6862
c3	tiny	-3.1103	-4.4041	-5.1793	-8.4131	-7.2096
c3	small	2.2039	1.5447	1.0121	-0.3998	1.4927
c3	average	0.3021	-1.3939	-1.2201	-3.3407	-2.1467
c3	big	-0.9621	-1.2439	-1.6092	-3.7666	-3.4389
c3	huge	0	0	0	0	0
c1_3*c3	tiny	-21.9104	-21.7840	-22.0173	-22.6066	-21.9791
c1_3*c3	small	-20.8196	-20.2725	-19.5850	-20.4515	-20.7586
c1_3*c3	average	-16.8500	-15.1509	-15.0134	-15.3851	-13.4339
c1_3*c3	big	-12.7212	-12.1554	-12.0354	-12.3282	-13.0174
c1_3*c3	huge	0	0	0	0	0
x1		0.9238	1.7286	2.5976	-0.2488	1.2093
x1*c3	tiny	-1.5819	-1.1748	-3.2523	-1.7016	-2.7624
x1*c3	small	-3.7669	-3.2984	-2.9755	-1.8738	-4.0167
x1*c3	average	2.2253	2.4489	1.5675	4.0948	2.0159
x1*c3	big	0.9222	0.5330	0.7960	2.6061	1.2694
x1*c3	huge	0	0	0	0	0
x5		-1.3562	0.5639	0.3022	-0.4700	-2.5063
x6		-0.9165	-3.2944	-1.2163	-2.2063	-0.5696
x7		5.2295	5.3015	6.2526	4.1770	5.8364
x6*x7		6.4211	7.5644	6.1182	7.0020	5.8730
x10		1.9591	1.4932	0.7196	0.6504	-0.3989
x5*x10		3.6058	1.7274	4.3447	2.4388	3.8967
x15		-0.0079	0.6896	1.6811	0.0136	0.1799
x15*c1_3		-3.5022	-2.7963	-2.6003	-4.2355	-4.7546
x7*x15		-5.1438	-5.8878	-5.9465	-3.6155	-5.3337
x18		-2.1347	-1.5656	-2.4226	-4.0592	-1.4985
x18*c3	tiny	2.2988	1.1931	2.6491	6.1615	5.6204
x18*c3	small	4.6033	3.2359	4.4183	5.5923	1.7270
x18*c3	average	-2.3712	-2.5392	-0.6361	-1.1729	-1.6481
x18*c3	big	2.3160	1.4654	2.7683	3.0487	2.5768
x18*c3	huge	0	0	0	0	0
x6*x18		3.0716	4.2036	4.1354	4.9196	2.7165
x20		4.1229	4.5773	4.5774	4.6555	4.2655

The CVDETAILS=ALL option on the MODEL statement requests the "CV Details" table shown in Output 2.15 and the cross validation parameter estimates that are included in the "Parameter Estimates" table shown in Output 2.16. For each cross validation index, the predicted residual sum of squares on the observations omitted is shown in the "CV Details" table and the parameter estimates of the corresponding model are included in the "Parameter Estimate"s table. By default, these details are shown for the selected model, but you can request this information at every step with the DETAILS= option on the MODEL statement. You use the "_CVINDEX_" variable in the output data set shown in Output 2.17 to find out which observations in the analysis data are omitted for each cross validation fold.

Output 2.17. First Eight Observations in the Output Data Set

Obs	c3	x1	x2	x3	x4	x5	x6	x7	x8
1	tiny	0.18496	0.97009	0.39982	0.25940	0.92160	0.96928	0.54298	0.53169
2	tiny	0.47579	0.84499	0.63452	0.59036	0.58258	0.37701	0.72836	0.50660
3	tiny	0.51132	0.43320	0.17611	0.66504	0.40482	0.12455	0.45349	0.19955
4	tiny	0.42071	0.07174	0.35849	0.71143	0.18985	0.14797	0.56184	0.27011
5	tiny	0.42137	0.03798	0.27081	0.42773	0.82010	0.84345	0.87691	0.26722
6	tiny	0.81722	0.65822	0.02947	0.85339	0.36285	0.37732	0.51054	0.71194
7	tiny	0.19480	0.81673	0.08548	0.18376	0.33264	0.70558	0.92761	0.29642
8	tiny	0.04403	0.51697	0.68884	0.45333	0.83565	0.29745	0.40325	0.95684

Obs	x9	x10	x11	x12	x13	x14	x15	x16	x17
1	0.04979	0.06657	0.81932	0.52387	0.85339	0.06718	0.95702	0.29719	0.27261
2	0.93121	0.92912	0.58966	0.29722	0.39104	0.47243	0.67953	0.16809	0.16653
3	0.57484	0.73847	0.43981	0.04937	0.52238	0.34337	0.02271	0.71289	0.93706
4	0.32520	0.56918	0.04259	0.43921	0.91744	0.52584	0.73182	0.90522	0.57600
5	0.30602	0.39705	0.34905	0.76593	0.54340	0.61257	0.55291	0.73591	0.37186
6	0.37533	0.22954	0.68621	0.55243	0.58182	0.17472	0.04610	0.64380	0.64545
7	0.22404	0.14719	0.59064	0.46326	0.41860	0.25631	0.23045	0.08034	0.43559
8	0.42194	0.78079	0.33106	0.17210	0.91056	0.26897	0.95602	0.13720	0.27190

Obs	x18	x19	x20	c1	c2	y	_CVINDEX_	p_y
1	0.68993	0.97676	0.22651	2	1	11.4391	1	18.1474
2	0.87110	0.29879	0.93464	3	1	31.4596	2	24.7930
3	0.44599	0.94694	0.71290	4	3	16.4294	3	16.5752
4	0.18794	0.33133	0.69887	5	3	15.4815	4	14.7605
5	0.64565	0.55718	0.87504	6	2	26.0023	5	24.7479
6	0.09317	0.62008	0.07845	7	1	16.6503	1	21.4444
7	0.67020	0.42272	0.49827	1	1	14.0342	2	20.9661
8	0.55692	0.65825	0.68465	2	3	14.9830	3	17.5644

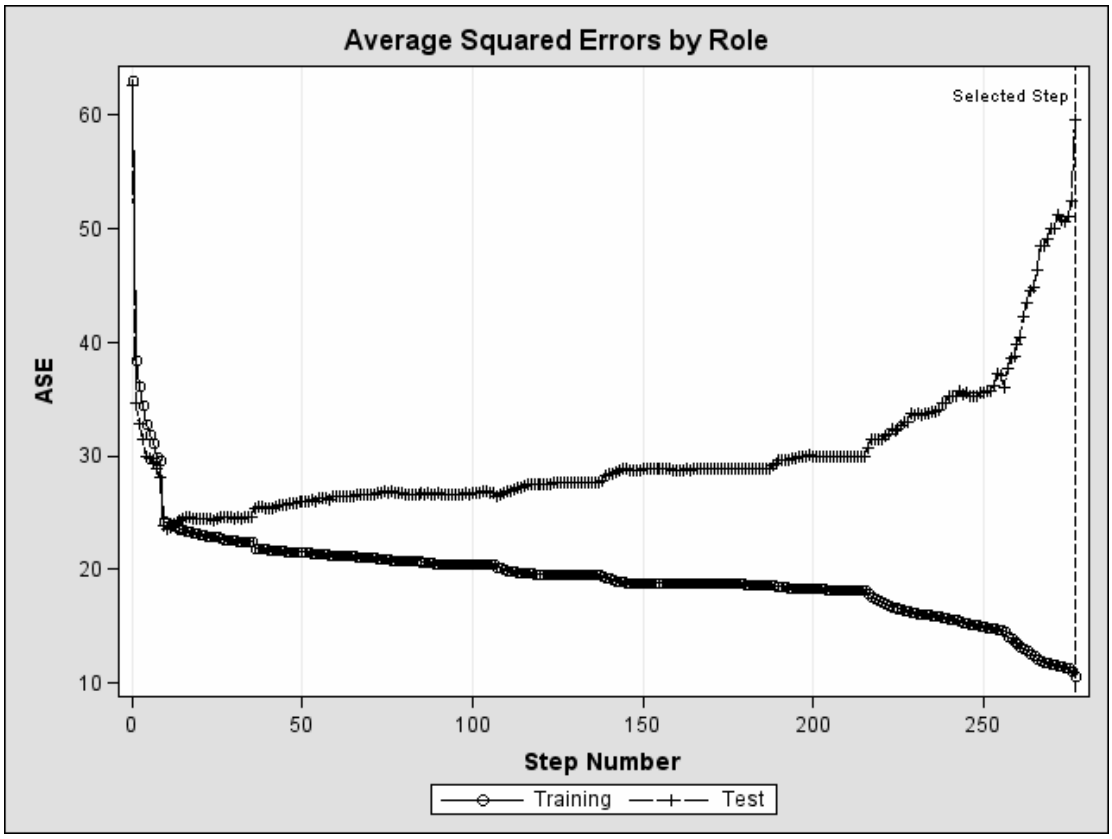
This example demonstrates the usefulness of effect selection when you suspect that interactions of effects are needed to explain the variation in your dependent variable. Ideally, a priori knowledge should be used decide what interactions to allow but in some cases this information might not be available. Simply fitting a least-squares model allowing all interactions produces a model that overfits your data and generalizes very poorly.

This following code uses forward selection with selection based on the SBC criterion which is the default selection criterion. At each step, the effect whose addition to the model yields the smallest SBC value is added. The STOP=NONE suboption specifies that this process continues even when the SBC statistic grows whenever an effect is added and so terminates at a full least-squares model. The BUILDSSCP=FULL option is specified on a PERFORMANCE statement, as building the SSCP matrix incrementally is counterproductive in this case. See the "Building the SSCP Matrix" section on page 53 for details. Note that if all you are interested in is a full least-squares model then it is much more efficient to simply specify SELECTION=NONE on the MODEL statement. However, in this example the aim is to add effects in roughly increasing order of explanatory power.

```
ods html;
ods graphics on;
```

```
proc glmselect data=analysisData testdata=testData plots=ASEPlot;
   class c1 c2 c3(order=data);

   model y =  c1|c2|c3|x1|x2|x3|x4|x5|x5|x6|x7|x8|x9|x10
             |x11|x12|x13|x14|x15|x16|x17|x18|x19|x20 @2
           / selection=forward(stop=none)
             hierarchy=single;

   performance buildSSCP = full;
run;

ods graphics off;
ods html close;
```

Output 2.18. Average Squared Error Plot



The ASE plot shown in Output 2.18 clearly demonstrates the danger in overfitting the training data. As more insignificant effects are added to the model, the growth in test set ASE shows how the the predictions produced by the resulting models worsen. This decline is particularly rapid in the latter stages of the forward selection as using the SBC criterion results in insignificant effects with lots of parameters being added after insignificant effects with fewer parameters.

References

Burnham, K.P., and Anderson, D.R. (2002), *Model Selection and Multimodel Inference, Second Edition,* New York: Springer-Verlag, Inc.

Collier Books (1987), *The 1987 Baseball Encyclopedia Update,* New York: Macmillan Publishing Company.

Darlington, R.B. (1968), "Multiple Regression in Psychological Research and Practice," *Psychological Bulletin*, 69, 161–182.

Draper, N.R., Guttman, I., and Kanemasu, H. (1971), "The Distribution of Certain Regression Statistics," *Biometrika*, 58, 295–298.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression (with discussion)," *Annals of Statistics*, 32, 407–499.

Foster, D.P. and Stine, R.A. (2004), "Variable Selection in Data Mining: Building a Predictive Model for Bankruptcy," *Journal of the American Statistical Association*, 99, 303–313.

Harrell, F.E. (2001), *Regression Modeling Strategies,* New York: Springer-Verlag, Inc.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning,* New York: Springer-Verlag, Inc.

Hocking, R.R. (1976), "The Analysis and Selection of Variables in Linear Regression," *Biometrics*, 32, 1–50.

Hurvich, C.M. and Tsai, C-L. (1991), "Bias of the Corrected AIC Criterion for Underfitted Regression and Time Series Models," *Biometrika*, 78, 499–509.

Judge, G.G., Griffiths, W.E., Hill, R.C., Lutkepohl, H., and Lee, T.C. (1985), *The Theory and Practice of Econometrics,* Second Edition, New York: John Wiley & Sons, Inc.

Mallows, C.L. (1967), "Choosing a Subset Regression," unpublished report, Bell Telephone Laboratories.

Mallows, C.L. (1973), "Some Comments on C_p," *Technometrics*, 15, 661–675.

Miller A. (2002), *Subset Selection in Regression, Second Edition*, Chapman & Hall/CRC.

Osborne, M., Presnell, B., and Turlach, B. (2000), "A New Approach to Variable Selection in Least Squares Problems," *IMA Journal of Numerical Analysis*, 20, 389–404.

Raftery, A.E., Madigan, D., and Hoeting, J.A. (1997), "Bayesian Model Averaging for Linear Regression Models," *Journal of the American Statistical Association*, 92, 179-191.

Sawa, T. (1978), "Information Criteria for Discriminating Among Alternative Regression Models," *Econometrica*, 46, 1273–1282.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Sports Illustrated (April 20, 1987).

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society Series B*, 58, 267–288.

Subject Index

N

O

P

S

T

V

Syntax Index

H

I

K

L

M

N

O

P

R

S

T

V

W