
# SAS/STAT® 14.3 User's Guide
# The HPGENSELECT Procedure

This document is an individual chapter from *SAS/STAT® 14.3 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS/STAT® 14.3 User's Guide*. Cary, NC: SAS Institute Inc.

**SAS/STAT® 14.3 User's Guide**



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

September 2017

# Chapter 54
# The HPGENSELECT Procedure

## Contents

# Overview: HPGENSELECT Procedure

The HPGENSELECT procedure is a high-performance procedure that provides model fitting and model building for generalized linear models. It fits models for standard distributions in the exponential family, such as the normal, Poisson, and Tweedie distributions. In addition, PROC HPGENSELECT fits multinomial models for ordinal and nominal responses, and it fits zero-inflated Poisson and negative binomial models for count data. For all these models, the HPGENSELECT procedure provides forward, backward, and stepwise variable selection.

PROC HPGENSELECT runs in either single-machine mode or distributed mode.

**NOTE:** Distributed mode requires SAS High-Performance Statistics.

## PROC HPGENSELECT Features

The HPGENSELECT procedure does the following:

- estimates the parameters of a generalized linear regression model by using maximum likelihood techniques
- provides model-building syntax in the CLASS statement and the effect-based MODEL statement, which are familiar from SAS/STAT procedures (in particular, the GLM, GENMOD, LOGISTIC, GLIMMIX, and MIXED procedures)
- enables you to split classification effects into individual components by using the SPLIT option in the CLASS statement
- permits any degree of interaction effects that involve classification and continuous variables
- provides multiple link functions
- provides models for zero-inflated count data
- provides cumulative link modeling for ordinal data and generalized logit modeling for unordered multinomial data
- enables model building (variable selection) through the SELECTION statement
- provides a WEIGHT statement for weighted analysis
- provides a FREQ statement for grouped analysis
- provides an OUTPUT statement to produce a data set that has predicted values and other observation-wise statistics

Because the HPGENSELECT procedure is a high-performance analytical procedure, it also does the following:

- enables you to run in distributed mode on a cluster of machines that distribute the data and the computations
- enables you to run in single-machine mode on the server where SAS is installed
- exploits all the available cores and concurrent threads, regardless of execution mode

For more information, see the section "Processing Modes" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

## PROC HPGENSELECT Contrasted with PROC GENMOD

This section contrasts the HPGENSELECT procedure with the GENMOD procedure in SAS/STAT software.

The CLASS statement in the HPGENSELECT procedure permits two parameterizations: GLM parameterization and a reference parameterization. In contrast to the LOGISTIC, GENMOD, and other procedures that permit multiple parameterizations, the HPGENSELECT procedure does not mix parameterizations across the variables in the CLASS statement. In other words, all classification variables have the same parameterization, and this parameterization is either GLM parameterization or reference parameterization. The CLASS statement also enables you to split an effect that involves a classification variable into multiple effects that correspond to individual levels of the classification variable.

The default optimization technique used by the HPGENSELECT procedure is a modification of the Newton-Raphson algorithm with a ridged Hessian. You can choose different optimization techniques (including first-order methods that do not require a crossproducts matrix or Hessian) by specifying the TECHNIQUE= option in the PROC HPGENSELECT statement.

As in the GENMOD procedure, the default parameterization of CLASS variables in the HPGENSELECT procedure is GLM parameterization. You can change the parameterization by specifying the PARAM= option in the CLASS statement.

The GENMOD procedure offers a wide variety of postfitting analyses, such as contrasts, estimates, tests of model effects, and least squares means. The HPGENSELECT procedure is limited in postfitting functionality because it is primarily designed for large-data tasks, such as predictive model building, model fitting, and scoring.

# Getting Started: HPGENSELECT Procedure

This example illustrates how you can use PROC HPGENSELECT to perform Poisson regression for count data. The following DATA step contains 100 observations for a count response variable (Y), a continuous variable (Total) to be used in a later analysis, and five categorical variables (C1–C5), each of which has four numerical levels:

```
data getStarted;
  input C1-C5 Y Total;
  datalines;
0 3 1 1 3 2 28.361
2 3 0 3 1 2 39.831
1 3 2 2 2 1 17.133
1 2 0 0 3 2 12.769
0 2 1 0 1 1 29.464
0 2 1 0 2 1 4.152
1 2 1 0 1 0 0.000
0 2 1 1 2 1 20.199
1 2 0 0 1 0 0.000
0 1 1 3 3 2 53.376
2 2 2 2 1 1 31.923
0 3 2 0 3 2 37.987
2 2 2 0 0 1 1.082
0 2 0 2 0 1 6.323
1 3 0 0 0 0 0.000
1 2 1 2 3 2 4.217
0 1 2 3 1 1 26.084
1 1 0 0 1 0 0.000
1 3 2 2 2 0 0.000
2 1 3 1 1 2 52.640
1 3 0 1 2 1 3.257
2 0 2 3 0 5 88.066
2 2 2 1 0 1 15.196
3 1 3 1 0 1 11.955
3 1 3 1 2 3 91.790
3 1 1 2 3 7 232.417
3 1 1 1 0 1 2.124
3 1 0 0 0 2 32.762
3 1 2 3 0 1 25.415
2 2 0 1 2 1 42.753
3 3 2 2 3 1 23.854
2 0 0 2 3 2 49.438
1 0 0 2 3 4 105.449
0 0 2 3 0 6 101.536
0 3 1 0 0 0 0.000
3 0 1 0 1 1 5.937
2 0 0 0 3 2 53.952
1 0 1 0 3 2 23.686
1 1 3 1 1 1 0.287
2 1 3 0 3 7 281.551
1 3 2 1 1 0 0.000
2 1 0 0 1 0 0.000
0 0 1 1 2 3 93.009
0 1 0 1 0 2 25.055
1 2 2 2 3 1 1.691
0 3 2 3 1 1 10.719
3 3 0 3 3 1 19.279
2 0 0 2 1 2 40.802
2 2 3 0 3 3 72.924
0 2 0 3 0 1 10.216
```

```
3 0 1 2 2 2 87.773
2 1 2 3 1 0 0.000
3 2 0 3 1 0 0.000
3 0 3 0 0 2 62.016
1 3 2 2 1 3 36.355
2 3 2 0 3 1 23.190
1 0 1 2 1 1 11.784
2 1 2 2 2 5 204.527
3 0 1 1 2 5 115.937
0 1 1 3 2 1 44.028
2 2 1 3 1 4 52.247
1 1 0 0 1 1 17.621
3 3 1 2 1 2 10.706
2 2 0 2 3 3 81.506
0 1 0 0 2 2 81.835
0 1 2 0 1 2 20.647
3 2 2 2 0 1 3.110
2 2 3 0 0 1 13.679
1 2 2 3 2 1 6.486
3 3 2 2 1 2 30.025
0 0 3 1 3 6 202.172
3 2 3 1 2 3 44.221
0 3 0 0 0 1 27.645
3 3 3 0 3 2 22.470
2 3 2 0 2 0 0.000
1 3 0 2 0 1 1.628
1 3 1 0 2 0 0.000
3 2 3 3 0 1 20.684
3 1 0 2 0 4 108.000
0 1 2 2 1 1 4.615
0 2 3 2 2 1 12.461
0 3 2 0 1 3 53.798
2 1 1 2 0 1 36.320
1 0 3 0 0 0 0.000
0 0 3 2 0 1 19.902
0 2 3 1 0 0 0.000
2 2 2 1 3 2 31.815
3 3 3 0 0 0 0.000
2 2 1 3 3 2 17.915
0 2 3 2 3 2 69.315
1 3 1 2 1 0 0.000
3 0 1 1 1 4 94.050
2 1 1 1 3 6 242.266
0 2 0 3 2 1 40.885
2 0 1 1 2 2 74.708
2 2 2 2 3 2 50.734
1 0 2 2 1 3 35.950
1 3 3 1 1 1 2.777
3 1 2 1 3 5 118.065
0 3 2 1 2 0 0.000
;
```

The following statements fit a log-linked Poisson model to these data by using classification effects for variables C1–C5:

```
proc hpgenselect data=getStarted;
   class C1-C5;
   model Y = C1-C5 /  Distribution=Poisson Link=Log;
run;
```

The default output from this analysis is presented in Figure 54.1 through Figure 54.8.

The "Performance Information" table in Figure 54.1 shows that the procedure executed in single-machine mode (that is, on the server where SAS is installed). When high-performance procedures run in single-machine mode, they use concurrently scheduled threads. In this case, four threads were used.

**Figure 54.1** Performance Information

**The HPGENSELECT Procedure**

| Performance Information | |
|---|---|
| **Execution Mode** | Single-Machine |
| **Number of Threads** | 4 |

Figure 54.2 displays the "Model Information" table. The variable Y is an integer-valued variable that is modeled by using a Poisson probability distribution, and the mean of Y is modeled by using a log link function. The HPGENSELECT procedure uses a Newton-Raphson algorithm to fit the model. The CLASS variables C1–C5 are parameterized by using GLM parameterization, which is the default.

**Figure 54.2** Model Information

| Model Information | |
|---|---|
| **Data Source** | WORK.GETSTARTED |
| **Response Variable** | Y |
| **Class Parameterization** | GLM |
| **Distribution** | Poisson |
| **Link Function** | Log |
| **Optimization Technique** | Newton-Raphson with Ridging |

Each of the CLASS variables C1–C5 has four unique formatted levels, which are displayed in the "Class Level Information" table in Figure 54.3.

**Figure 54.3** Class Level Information

**Class Level Information**

| Class | Levels | Values |
|---|---|---|
| C1 | 4 | 0 1 2 3 |
| C2 | 4 | 0 1 2 3 |
| C3 | 4 | 0 1 2 3 |
| C4 | 4 | 0 1 2 3 |
| C5 | 4 | 0 1 2 3 |

Figure 54.4 displays the "Number of Observations" table. All 100 observations in the data set are used in the analysis.

**Figure 54.4** Number of Observations

| | |
|---|---|
| **Number of Observations Read** | 100 |
| **Number of Observations Used** | 100 |

Figure 54.5 displays the "Dimensions" table for this model. This table summarizes some important sizes of various model components. For example, it shows that there are 21 columns in the design matrix $\mathbf{X}$: one column for the intercept and 20 columns for the effects that are associated with the classification variables C1–C5. However, the rank of the crossproducts matrix is only 16. Because the classification variables C1–C5 use GLM parameterization and because the model contains an intercept, there is one singularity in the crossproducts matrix of the model for each classification variable. Consequently, only 16 parameters enter the optimization.

**Figure 54.5** Dimensions in Poisson Regression

| Dimensions | |
|---|---|
| **Number of Effects** | 6 |
| **Number of Parameters** | 16 |
| **Columns in X** | 21 |

Figure 54.6 displays the final convergence status of the Newton-Raphson algorithm. The GCONV= relative convergence criterion is satisfied.

**Figure 54.6** Convergence Status

| |
|---|
| Convergence criterion (GCONV=1E-8) satisfied. |

The "Fit Statistics" table is shown in Figure 54.7. The –2 log likelihood at the converged estimates is 290.16169. You can use this value to compare the model to nested model alternatives by means of a likelihood-ratio test. To compare models that are not nested, information criteria such as AIC (Akaike's information criterion), AICC (Akaike's bias-corrected information criterion), and BIC (Schwarz Bayesian information criterion) are used. These criteria penalize the –2 log likelihood for the number of parameters.

**Figure 54.7** Fit Statistics

| Fit Statistics | |
|---|---|
| **-2 Log Likelihood** | 290.16 |
| **AIC (smaller is better)** | 322.16 |
| **AICC (smaller is better)** | 328.72 |
| **BIC (smaller is better)** | 363.84 |
| **Pearson Chi-Square** | 77.7694 |
| **Pearson Chi-Square/DF** | 0.9258 |

The "Parameter Estimates" table in Figure 54.8 shows that many parameters have fairly large *p*-values, indicating that one or more of the model effects might not be necessary.

**Figure 54.8** Parameter Estimates

| Parameter Estimates | | | | | |
|---|---|---|---|---|---|
| **Parameter** | **DF** | **Estimate** | **Standard Error** | **Chi-Square** | **Pr > ChiSq** |
| **Intercept** | 1 | 0.881903 | 0.382730 | 5.3095 | 0.0212 |
| **C1 0** | 1 | -0.196002 | 0.211482 | 0.8590 | 0.3540 |
| **C1 1** | 1 | -0.605161 | 0.263508 | 5.2742 | 0.0216 |
| **C1 2** | 1 | -0.068458 | 0.210776 | 0.1055 | 0.7453 |
| **C1 3** | 0 | 0 | . | . | . |
| **C2 0** | 1 | 0.961117 | 0.255485 | 14.1521 | 0.0002 |
| **C2 1** | 1 | 0.708188 | 0.246768 | 8.2360 | 0.0041 |
| **C2 2** | 1 | 0.161741 | 0.266365 | 0.3687 | 0.5437 |
| **C2 3** | 0 | 0 | . | . | . |
| **C3 0** | 1 | -0.227016 | 0.252561 | 0.8079 | 0.3687 |
| **C3 1** | 1 | -0.094775 | 0.229519 | 0.1705 | 0.6797 |
| **C3 2** | 1 | 0.044801 | 0.238127 | 0.0354 | 0.8508 |
| **C3 3** | 0 | 0 | . | . | . |
| **C4 0** | 1 | -0.280476 | 0.263589 | 1.1322 | 0.2873 |
| **C4 1** | 1 | 0.028157 | 0.249652 | 0.0127 | 0.9102 |
| **C4 2** | 1 | 0.047803 | 0.240378 | 0.0395 | 0.8424 |
| **C4 3** | 0 | 0 | . | . | . |
| **C5 0** | 1 | -0.817936 | 0.219901 | 13.8351 | 0.0002 |
| **C5 1** | 1 | -0.710596 | 0.206265 | 11.8684 | 0.0006 |
| **C5 2** | 1 | -0.602080 | 0.217724 | 7.6471 | 0.0057 |
| **C5 3** | 0 | 0 | . | . | . |

# Syntax: HPGENSELECT Procedure

The following statements are available in the HPGENSELECT procedure:

**PROC HPGENSELECT** < *options* > **;**
**BY** *variables* **;**
**CLASS** *variable* < **(***options***)** >. . . < *variable* < **(***options***)** > > < / *global-options* > **;**
**CODE** < *options* > **;**
**FREQ** *variable* **;**
**ID** *variables* **;**
**MODEL** *response*< **(***response-options***)** > **=** < *effects* > < / *model-options* > **;**
**MODEL** *events/trials*< **(***response-options***)** > **=** < *effects* > < / *model-options* > **;**
**OUTPUT** < **OUT=***SAS-data-set* >
< *keyword* < **=***name* > >. . .
< *keyword* < **=***name* > > < / *options* > **;**
**PARTITION** < *partition-options* > **;**
**PERFORMANCE** *performance-options* **;**
**RESTRICT** < *'label'* > *constraint-specification* < **,** . . . **,** *constraint-specification* >
< *operator* < *value* > > < / *option* > **;**
**SELECTION** *selection-options* **;**
**WEIGHT** *variable* **;**
**ZEROMODEL** < *effects* >< / *zeromodel-options* > **;**

The PROC HPGENSELECT statement and at least one MODEL statement are required. The CLASS statement can appear multiple times. If a CLASS statement is specified, it must precede the MODEL statements.

## PROC HPGENSELECT Statement

**PROC HPGENSELECT** < *options* > **;**

The PROC HPGENSELECT statement invokes the procedure. Table 54.1 summarizes the available options in the PROC HPGENSELECT statement by function. The options are then described fully in alphabetical order.

**Table 54.1** PROC HPGENSELECT Statement Options

| Option | Description |
|---|---|
| **Basic Options** | |
| ALPHA= | Specifies a global significance level |
| DATA= | Specifies the input data set |
| NAMELEN= | Limits the length of effect names |

**Table 54.1** *continued*

| Option | Description |
|---|---|
| **Output Options** | |
| CORR | Displays the "Parameter Estimates Correlation Matrix" table |
| COV | Displays the "Parameter Estimates Covariance Matrix" table |
| ITDETAILS | Displays the "Iteration History" table when no model selection is performed |
| ITSELECT | Displays the summarized "Iteration History" table when model selection is performed |
| ITSUMMARY | Displays the summarized "Iteration History" table when no model selection is performed |
| NOPRINT | Suppresses ODS output |
| NOCLPRINT | Limits or suppresses the display of classification variable levels |
| NOSTDERR | Suppresses computation of the covariance matrix and standard errors |
| **Optimization Options** | |
| ABSCONV= | Tunes the absolute function convergence criterion |
| ABSFCONV= | Tunes the absolute function difference convergence criterion |
| ABSGCONV= | Tunes the absolute gradient convergence criterion |
| FCONV= | Tunes the relative function difference convergence criterion |
| GCONV= | Tunes the relative gradient convergence criterion |
| INEST= | Specifies the SAS data set that contains starting values, bounds, and constraints for single parameters |
| LASSORHO= | Specifies the base regularization parameter for the LASSO method |
| LASSOSTEPS= | Specifies the maximum number of steps for the LASSO method |
| MAXITER= | Chooses the maximum number of iterations in any optimization |
| MAXFUNC= | Specifies the maximum number of function evaluations in any optimization |
| MAXTIME= | Specifies the upper limit of CPU time (in seconds) for any optimization |
| MINITER= | Specifies the minimum number of iterations in any optimization |
| NORMALIZE= | Specifies whether the objective function is normalized during optimization |
| OUTEST | Adds parameter names to the "Parameter Estimates" table |
| TECHNIQUE= | Selects the optimization technique |
| XCONV= | Tunes the relative parameter difference convergence criterion |
| **Tolerance Options** | |
| LASSOTOL= | Specifies the convergence criterion for the LASSO method |
| SINGCHOL= | Tunes the singularity criterion for Cholesky decompositions |
| SINGSWEEP= | Tunes the singularity criterion for the sweep operator |
| SINGULAR= | Tunes the general singularity criterion |
| **User-Defined Format Options** | |
| FMTLIBXML= | Specifies the file reference for a format stream |

You can specify the following *options* in the PROC HPGENSELECT statement.

**ABSCONV=***r*

**ABSTOL=***r*

specifies an absolute function convergence criterion. For minimization, termination requires $f(\boldsymbol{\psi}^{(k)}) \leq r$, where $\boldsymbol{\psi}$ is the vector of parameters in the optimization and $f(\cdot)$ is the objective function. The

default value of *r* is the negative square root of the largest double-precision value, which serves only as a protection against overflow.

**ABSFCONV=***r* **<** *n* **>**

**ABSFTOL=***r* **<** *n* **>**

specifies an absolute function difference convergence criterion. For all techniques except NMSIMP, termination requires a small change of the function value in successive iterations:

$$|f(\boldsymbol{\psi}^{(k-1)}) - f(\boldsymbol{\psi}^{(k)})| \leq r$$

Here, $\boldsymbol{\psi}$ denotes the vector of parameters that participate in the optimization, and $f(\cdot)$ is the objective function. The same formula is used for the NMSIMP technique, but $\boldsymbol{\psi}(k)$ is defined as the vertex that has the lowest function value and $\boldsymbol{\psi}^{(k-1)}$ is defined as the vertex that has the highest function value in the simplex. The optional integer value *n* specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated. The default value is $r = 0$.

**ABSGCONV=***r* **<** *n* **>**

**ABSGTOL=***r* **<** *n* **>**

specifies an absolute gradient convergence criterion. Termination requires the maximum absolute gradient element to be small:

$$\max_j |g_j(\boldsymbol{\psi}^{(k)})| \leq r$$

Here, $\boldsymbol{\psi}$ denotes the vector of parameters that participate in the optimization, and $g_j(\cdot)$ is the gradient of the objective function with respect to the *j*th parameter. This criterion is not used by the NMSIMP technique. The optional integer value *n* specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated. The default value is $r = 1E{-}8$.

**ALPHA=***number*

specifies a global significance level for the construction of confidence intervals. The confidence level is 1 – *number*. The value of *number* must be between 0 and 1; the default is 0.05. You can override this global significance level by specifying the ALPHA= option in the MODEL statement or the ALPHA= option in the OUTPUT statement.

**CORR**

creates the "Parameter Estimates Correlation Matrix" table. The correlation matrix is computed by normalizing the covariance matrix $\boldsymbol{\Sigma}$. That is, if $\sigma_{ij}$ is an element of $\boldsymbol{\Sigma}$, then the corresponding element of the correlation matrix is $\sigma_{ij}/\sigma_i\sigma_j$, where $\sigma_i = \sqrt{\sigma_{ii}}$.

**COV**

creates the "Parameter Estimates Covariance Matrix" table. The covariance matrix is computed as the inverse of the negative of the matrix of second derivatives of the log-likelihood function with respect to the model parameters (the Hessian matrix).

**DATA=***SAS-data-set*

names the input SAS data set for PROC HPGENSELECT to use. The default is the most recently created data set.

If the procedure executes in distributed mode, the input data are distributed to memory on the appliance nodes and analyzed in parallel, unless the data are already distributed in the appliance database. In that case the procedure reads the data alongside the distributed database. For information about the various execution modes, see the section "Processing Modes" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*); for information about the alongside-the-database model, see the section "Alongside-the-Database Execution" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*)

**FCONV=***r*<*n*>

**FTOL=***r*<*n*>

specifies a relative function difference convergence criterion. For all techniques except NMSIMP, termination requires a small relative change of the function value in successive iterations:

$$\frac{|f(\boldsymbol{\psi}^{(k)}) - f(\boldsymbol{\psi}^{(k-1)})|}{|f(\boldsymbol{\psi}^{(k-1)})|} \leq r$$

Here, $\boldsymbol{\psi}$ denotes the vector of parameters that participate in the optimization, and $f(\cdot)$ is the objective function. The same formula is used for the NMSIMP technique, but $\boldsymbol{\psi}^{(k)}$ is defined as the vertex that has the lowest function value, and $\boldsymbol{\psi}^{(k-1)}$ is defined as the vertex that has the highest function value in the simplex.

The optional integer value *n* specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated. The default value is $r = 2 \times \epsilon$, where $\epsilon$ is the machine precision.

**FMTLIBXML=***file-ref*

specifies the file reference for the XML stream that contains the user-defined format definitions. User-defined formats are handled differently in a distributed computing environment than they are in other SAS products. For information about how to generate an XML stream for your formats, see the section "Working with Formats" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

**GCONV=***r*<*n*>

**GTOL=***r*<*n*>

specifies a relative gradient convergence criterion. For all techniques except CONGRA and NMSIMP, termination requires that the normalized predicted function reduction be small:

$$\frac{\mathbf{g}(\boldsymbol{\psi}^{(k)})'[\mathbf{H}^{(k)}]^{-1}\mathbf{g}(\boldsymbol{\psi}^{(k)})}{|f(\boldsymbol{\psi}^{(k)})|} \leq r$$

Here, $\boldsymbol{\psi}$ denotes the vector of parameters that participate in the optimization, $f(\cdot)$ is the objective function, and $\mathbf{g}(\cdot)$ is the gradient. For the CONGRA technique (where a reliable Hessian estimate $\mathbf{H}$ is not available), the following criterion is used:

$$\frac{\| \mathbf{g}(\boldsymbol{\psi}^{(k)}) \|_2^2 \quad \| \mathbf{s}(\boldsymbol{\psi}^{(k)}) \|_2}{\| \mathbf{g}(\boldsymbol{\psi}^{(k)}) - \mathbf{g}(\boldsymbol{\psi}^{(k-1)}) \|_2 \, |f(\boldsymbol{\psi}^{(k)})|} \leq r$$

This criterion is not used by the NMSIMP technique. The optional integer value *n* specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated. The default value is *r*=1E–8.

**INEST=***SAS-data-set*

names the SAS data set that contains starting values for the parameters. Your data set must include the _TYPE_ variable, a character variable in which the value 'PARMS' indicates the observation that contains your starting values. The data set also includes a numeric variable for each parameter for which you are specifying a starting value; the name of this numeric variable is the parameter name. You can obtain parameter names by specifying the OUTEST option and by using the ODS OUTPUT statement to output the "Parameter Estimates" table into a data set; the parameter name is contained in the ParmName variable in this data set. If you do not specify a starting value for a parameter, it is set to 0. PROC HPGENSELECT uses only the first observation for which _TYPE_=PARMS, and it ignores BY variables. You can also specify single-parameter equality constraints by using a value of 'EQ' for the variable _TYPE_ to indicate the observation that contains your equality constraints, and similarly by using values for _TYPE_ of 'UB' for upper bounds and 'LB' for lower bounds on parameters.

**ITDETAILS**

adds to the "Iteration History" table the current values of the parameter estimates and their gradients. These quantities are reported only for parameters that participate in the optimization. This option is not available when you perform model selection.

**ITSELECT**

generates the "Iteration History" table when you perform a model selection.

**ITSUMMARY**

generates the "Iteration History" table. This option is not available when you perform model selection.

**LASSORHO=***r*

specifies the base regularization parameter for the LASSO model selection method. The regularization parameter for step $i$ is $r^i$.

**LASSOSTEPS=***n*

specifies the maximum number of steps for LASSO model selection.

**LASSOTOL=***r*

specifies the convergence tolerance for the optimization algorithm that solves for the LASSO parameter estimates at each step of LASSO model selection.

**MAXFUNC=***n*

**MAXFU=***n*

specifies the maximum number of function calls in the optimization process. The default values are as follows, depending on the optimization technique:

- TRUREG, NRRIDG, NEWRAP: $n = 125$
- QUANEW, DBLDOG: $n = 500$
- CONGRA: $n = 1{,}000$
- NMSIMP: $n = 3{,}000$

The optimization can terminate only after completing a full iteration. Therefore, the number of function calls that are actually performed can exceed *n*. You can choose the optimization technique by specifying the TECHNIQUE= option.

**MAXITER=***n*

**MAXIT=***n*

specifies the maximum number of iterations in the optimization process. The default values are as follows, depending on the optimization technique:

- TRUREG, NRRIDG, NEWRAP: $n = 50$
- QUANEW, DBLDOG: $n = 200$
- CONGRA: $n = 400$
- NMSIMP: $n = 1{,}000$

These default values also apply when *n* is specified as a missing value. You can choose the optimization technique by specifying the TECHNIQUE= option.

**MAXTIME=***r*

specifies an upper limit of *r* seconds of CPU time for the optimization process. The default value is the largest floating-point double representation of your computer. The time specified by this option is checked only once at the end of each iteration. Therefore, the actual running time can be longer than *r*.

**MINITER=***n*

**MINIT=***n*

specifies the minimum number of iterations. The default value is 0. If you request more iterations than are actually needed for convergence to a stationary point, the optimization algorithms might behave strangely. For example, the effect of rounding errors can prevent the algorithm from continuing for the required number of iterations.

**NAMELEN=***number*

specifies the length to which long effect names are shortened. The default and minimum value is 20.

**NOCLPRINT< =***number* **>**

suppresses the display of the "Class Level Information" table if you do not specify *number*. If you specify *number*, the values of the classification variables are displayed for only those variables whose number of levels is less than *number*. Specifying a *number* helps to reduce the size of the "Class Level Information" table if some classification variables have a large number of levels.

**NOPRINT**

suppresses the generation of ODS output.

**NORMALIZE=YES | NO**

specifies whether to normalize the objective function during optimization by the reciprocal of the frequency count of observations that are used in the analysis. This option affects the values that are reported in the "Iteration History" table. The results that are reported in the "Fit Statistics" are always displayed for the nonnormalized log-likelihood function. By default, NORMALIZE = NO.

**NOSTDERR**

suppresses the computation of the covariance matrix and the standard errors of the regression coefficients. When the model contains many variables (thousands), the inversion of the Hessian matrix to derive the covariance matrix and the standard errors of the regression coefficients can be time-consuming.

**OUTEST**

adds a column for the ParmName variable to the "Parameter Estimates" table. This column is not displayed, but you can use it to create a data set that you can specify in an INEST= option by first using the ODS OUTPUT statement to output the "Parameter Estimates" table and then submitting the following statements:

```
proc transpose data=parameterestimates out=inest label=_TYPE_;
   label Estimate=PARMS;
   var Estimate;
   id ParmName;
run;
```

**SINGCHOL=***number*

tunes the singularity criterion in Cholesky decompositions. The default is 1E4 times the machine epsilon; this product is approximately 1E–12 on most computers.

**SINGSWEEP=***number*

tunes the singularity criterion for sweep operations. The default is 1E4 times the machine epsilon; this product is approximately 1E–12 on most computers.

**SINGULAR=***number*

tunes the general singularity criterion that is applied in sweeps and inversions. The default is 1E4 times the machine epsilon; this product is approximately 1E–12 on most computers.

**TECHNIQUE=***keyword*

**TECH=***keyword*

specifies the optimization technique for obtaining maximum likelihood estimates. You can choose from the following techniques by specifying the appropriate *keyword*:

| | |
|---|---|
| CONGRA | performs a conjugate-gradient optimization. |
| DBLDOG | performs a version of double-dogleg optimization. |
| NEWRAP | performs a Newton-Raphson optimization with line search. |
| NMSIMP | performs a Nelder-Mead simplex optimization. |
| NONE | performs no optimization. |
| NRRIDG | performs a Newton-Raphson optimization with ridging. |
| QUANEW | performs a dual quasi-Newton optimization. |
| TRUREG | performs a trust-region optimization |

The default value is TECHNIQUE=NRRIDG, except for the Tweedie distribution, for which the default value is TECHNIQUE=QUANEW.

For more information, see the section "Choosing an Optimization Algorithm" on page 4344.

**XCONV=***r*< *n*>

**XTOL=***r*< *n*>

specifies the relative parameter convergence criterion. The termination criterion and the default value depend on the technique, as follows:

- For all techniques except NMSIMP, termination requires a small relative parameter change in subsequent iterations:

$$\frac{\max_j |\psi_j^{(k)} - \psi_j^{(k-1)}|}{\max(|\psi_j^{(k)}|, |\psi_j^{(k-1)}|)} \leq r$$

  By default, XCONV = 0.

- For the NMSIMP technique, the same formula is used, but $\psi_j^{(k)}$ is defined as the vertex with the lowest function value and $\psi_j^{(k-1)}$ is defined as the vertex with the highest function value in the simplex. The default value is $r$ = 1E–8.

The optional integer value *n* specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

## BY Statement

**BY** *variables* **;**

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

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

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the HPGENSELECT procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

BY statement processing is not supported when the HPGENSELECT procedure runs alongside the database or alongside the Hadoop Distributed File System (HDFS). These modes are used if the input data are stored in a database or HDFS and the grid host is the appliance that houses the data.

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

## CLASS Statement

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

The CLASS statement names the classification variables to be used as explanatory variables in the analysis. The CLASS statement must precede the MODEL statement. You can list the response variable for binary and multinomial models in the CLASS statement, but this is not necessary.

The CLASS statement is documented in the section "CLASS Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

The HPGENSELECT procedure additionally supports the following *global-option* in the CLASS statement:

**UPCASE**

uppercases the values of character-valued CLASS variables before levelizing them. For example, if the UPCASE option is in effect and a CLASS variable can take the values 'a', 'A', and 'b', then 'a' and 'A' represent the same level and the CLASS variable is treated as having only two values: 'A' and 'B'.

## CODE Statement

**CODE** <*options*> ;

The CODE statement writes SAS DATA step code for computing predicted values of the fitted model either to a file or to a catalog entry. This code can then be included in a DATA step to score new data.

Table 54.2 summarizes the *options* available in the CODE statement.

**Table 54.2** CODE Statement Options

| Option | Description |
|---|---|
| CATALOG= | Names the catalog entry where the generated code is saved |
| DUMMIES | Retains the dummy variables in the data set |
| ERROR | Computes the error function |
| FILE= | Names the file where the generated code is saved |
| FORMAT= | Specifies the numeric format for the regression coefficients |
| GROUP= | Specifies the group identifier for array names and statement labels |
| IMPUTE | Imputes predicted values for observations with missing or invalid covariates |
| LINESIZE= | Specifies the line size of the generated code |
| LOOKUP= | Specifies the algorithm for looking up CLASS levels |
| RESIDUAL | Computes residuals |

For details about the syntax of the CODE statement, see the section "CODE Statement" on page 399 in Chapter 19, "Shared Concepts and Topics."

## FREQ Statement

**FREQ** *variable* **;**

The *variable* in the FREQ statement identifies a numeric variable in the data set that contains the frequency of occurrence for each observation. PROC HPGENSELECT treats each observation as if it appeared $f$ times, where the frequency value $f$ is the value of the FREQ variable for the observation. If $f$ is not an integer, then $f$ is truncated to an integer. If $f$ is less than 1 or missing, the observation is not used in the analysis. When the FREQ statement is not specified, each observation is assigned a frequency of 1.

## ID Statement

**ID** *variables* **;**

The ID statement lists one or more variables from the input data set that are to be transferred to the output data set that is specified in the OUTPUT statement.

For more information, see the section "ID Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

## MODEL Statement

**MODEL** *response* < **(***response-options***)** > **=** < *effects* > < / *model-options* > **;**

**MODEL** *events* / *trials* **=** < *effects* > < / *model-options* > **;**

The MODEL statement defines the statistical model in terms of a *response* variable (the target) or an *events/trials* specification. You can also specify model effects that are constructed from variables in the input data set, and you can specify options. An intercept is included in the model by default. You can remove the intercept by specifying the NOINT option.

You can specify a single *response* variable that contains your interval, binary, ordinal, or nominal response values. When you have binomial data, you can specify the *events/trials* form of the response, where one variable contains the number of positive responses (or events) and another variable contains the number of trials. The values of both *events* and (*trials* – *events*) must be nonnegative, and the value of *trials* must be positive. If you specify a single *response* variable that is in a CLASS statement, then the response is assumed to be either binary or multinomial, depending on the number of levels.

For information about constructing the model effects, see the section "Specification and Parameterization of Model Effects" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

There are two sets of options in the MODEL statement. The *response-options* determine how the HPGENSELECT procedure models probabilities for binary and multinomial data. The *model-options* control other aspects of model formation and inference. Table 54.3 summarizes these options.

**Table 54.3** MODEL Statement Options

| Option | Description |
|---|---|
| **Response Variable Options for Binary and Multinomial Models** | |
| DESCENDING | Reverses the response categories |
| EVENT= | Specifies the event category |
| ORDER= | Specifies the sort order |
| REF= | Specifies the reference category |
| **Model Options** | |
| ALPHA= | Specifies the confidence level for confidence limits |
| CL | Requests confidence limits |
| DISPERSION \| PHI= | Specifies a fixed dispersion parameter |
| DISTRIBUTION \| DIST= | Specifies the response distribution |
| INCLUDE= | Includes effects in all models for model selection |
| INITIALPHI= | Specifies a starting value of the dispersion parameter |
| LINK= | Specifies the link function |
| NOCENTER | Requests that continuous main effects not be centered and scaled |
| NOINT | Suppresses the intercept |
| OFFSET= | Specifies the offset variable |
| SAMPLEFRAC= | Specifies the fraction of the data to be used to compute starting values for the Tweedie distribution |
| START= | Includes effects in the initial model for model selection |

## Response Variable Options

Response variable options determine how the HPGENSELECT procedure models probabilities for binary and multinomial data.

You can specify the following *response-options* by enclosing them in parentheses after the *response* or *trials* variable.

**DESCENDING**

**DESC**

reverses the order of the response categories. If both the DESCENDING and ORDER= options are specified, PROC HPGENSELECT orders the response categories according to the ORDER= option and then reverses that order.

**EVENT='***category***' | FIRST | LAST**

specifies the event category for the binary response model. PROC HPGENSELECT models the probability of the event category. The EVENT= option has no effect when there are more than two response categories.

You can specify the event *category* (formatted, if a format is applied) in quotes, or you can specify one of the following:

**FIRST**
designates the first ordered category as the event. This is the default.

**LAST**
designates the last ordered category as the event.

For example, the following statements specify that observations that have a formatted value of '1' represent events in the data. The probability modeled by the HPGENSELECT procedure is thus the probability that the variable def takes on the (formatted) value '1'.

```
proc hpgenselect data=MyData;
   class A B C;
   model def(event ='1') = A B C x1 x2 x3;
run;
```

**ORDER=DATA | FORMATTED | INTERNAL**

**ORDER=FREQ | FREQDATA | FREQFORMATTED | FREQINTERNAL**
specifies the sort order for the levels of the *response* variable. When ORDER=FORMATTED (the default) for numeric variables for which you have supplied no explicit format (that is, for which there is no corresponding FORMAT statement in the current PROC HPGENSELECT run or in the DATA step that created the data set), the levels are ordered by their internal (numeric) value. Table 54.4 shows the interpretation of the ORDER= option.

**Table 54.4** Sort Order

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

By default, ORDER=FORMATTED. For the FORMATTED and INTERNAL orders, the sort order is machine-dependent.

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

**REF='***category***' | FIRST | LAST**

specifies the reference category for the generalized logit model and the binary response model. For the generalized logit model, each logit contrasts a nonreference category with the reference category. For the binary response model, specifying one response category as the reference is the same as specifying the other response category as the event category. You can specify the reference *category* (formatted if a format is applied) in quotes, or you can specify one of the following:

**FIRST**

designates the first ordered category as the reference

**LAST**

designates the last ordered category as the reference. This is the default.

## Model Options

**ALPHA=***number*

requests that confidence intervals for each of the parameters that are requested by the CL option be constructed with confidence level 1–*number*. The value of *number* must be between 0 and 1; the default is 0.05.

**CL**

requests that confidence limits be constructed for each of the parameter estimates. The confidence level is 0.95 by default; this can be changed by specifying the ALPHA= option.

**DISPERSION=***number*

specifies a fixed dispersion parameter for those distributions that have a dispersion parameter. The dispersion parameter used in all computations is fixed at *number*, and not estimated.

**DISTRIBUTION=***keyword*

specifies the response distribution for the model. The *keywords* and the associated distributions are shown in Table 54.5.

**Table 54.5** Built-In Distribution Functions

| DISTRIBUTION= | Distribution Function |
|---|---|
| BINARY | Binary |
| BINOMIAL | Binary or binomial |
| GAMMA | Gamma |
| INVERSEGAUSSIAN \| IG | Inverse Gaussian |
| MULTINOMIAL \| MULT | Multinomial |
| NEGATIVEBINOMIAL \| NB | Negative binomial |
| NORMAL \| GAUSSIAN | Normal |
| POISSON | Poisson |
| TWEEDIE<(*Tweedie-options*)> | Tweedie |
| ZINB | Zero-inflated negative binomial |
| ZIP | Zero-inflated Poisson |

When DISTRIBUTION=TWEEDIE, you can specify the following *Tweedie-options*:

**INITIALP=**
specifies a starting value for iterative estimation of the Tweedie power parameter.

**OPTMETHOD=***Tweedie-optimization-option*
requests an optimization method for iterative estimation of the Tweedie model parameters. You can specify the following *Tweedie-optimization-options*:

**EQL**
requests that extended quasi-likelihood be used for a sample of the data, followed by extended quasi-likelihood for the full data. This is equivalent to the TWEEDIEEQL *Tweedie-option*.

**EQLLHOOD**
requests that extended quasi-likelihood be used for a sample of the data, followed by Tweedie log likelihood for the full data. This is the default method.

**FINALLHOOD**
requests a four-stage approach to estimating the Tweedie model parameters. The four stages are as follows:

1. extended quasi-likelihood for a sample of the data
2. Tweedie log likelihood for a sample of the data
3. extended quasi-likelihood for the full data
4. Tweedie log likelihood for the full data

**LHOOD**
requests that Tweedie log likelihood be used for a sample of the data, followed by Tweedie log likelihood for the full data.

**P=**
requests a fixed Tweedie power parameter.

**TWEEDIEEQL | EQL**
requests that extended quasi-likelihood be used instead of Tweedie log likelihood in parameter estimation.

If you do not specify a link function with the LINK= option, a default link function is used. The default link function for each distribution is shown in Table 54.6. For the binary and multinomial distributions, only the link functions shown in Table 54.6 are available. For the other distributions, you can use any link function shown in Table 54.7 by specifying the LINK= option. Other commonly used link functions for each distribution are shown in Table 54.6.

**Table 54.6** Default and Commonly Used Link Functions

| DISTRIBUTION= | Default Link Function | Other Commonly Used Link Functions |
|---|---|---|
| BINARY | Logit | Probit, complementary log-log, log-log |
| BINOMIAL | Logit | Probit, complementary log-log, log-log |
| GAMMA | Reciprocal | Log |
| INVERSEGAUSSIAN \| IG | Reciprocal square | Log |
| MULTINOMIAL \| MULT | Logit (ordinal) | Probit, complementary log-log, log-log |
| MULTINOMIAL \| MULT | Generalized logit (nominal) | |
| NEGATIVEBINOMIAL \| NB | Log | |
| NORMAL \| GAUSSIAN | Identity | Log |
| POISSON | Log | |
| TWEEDIE | Log | |
| ZINB | Log | |
| ZIP | Log | |

**INCLUDE=***n*

**INCLUDE=***single-effect*

**INCLUDE=(***effects***)**

forces effects to be included in all models. If you specify INCLUDE=*n*, then the first *n* effects that are listed in the MODEL statement are included in all models. If you specify INCLUDE=*single-effect* or if you specify a list of effects within parentheses, then the specified effects are forced into all models. The effects that you specify in this option must be explanatory effects that are specified in the MODEL statement before the slash (/).

**INITIALPHI=***number*

specifies a starting value for iterative maximum likelihood estimation of the dispersion parameter for distributions that have a dispersion parameter.

**LINK=***keyword*

specifies the link function for the model. The *keywords* and the associated link functions are shown in Table 54.7. Default and commonly used link functions for the available distributions are shown in Table 54.6.

**Table 54.7** Built-In Link Functions

| LINK= | Link Function | $g(\mu) = \eta =$ |
|---|---|---|
| CLOGLOG \| CLL | Complementary log-log | $\log(-\log(1-\mu))$ |
| GLOGIT \| GENLOGIT | Generalized logit | |
| IDENTITY \| ID | Identity | $\mu$ |
| INV \| RECIP | Reciprocal | $\frac{1}{\mu}$ |
| INV2 | Reciprocal square | $\frac{1}{\mu^2}$ |
| LOG | Logarithm | $\log(\mu)$ |
| LOGIT | Logit | $\log(\mu/(1-\mu))$ |
| LOGLOG | Log-log | $-\log(-\log(\mu))$ |
| PROBIT | Probit | $\Phi^{-1}(\mu)$ |

$\Phi^{-1}(\cdot)$ denotes the quantile function of the standard normal distribution.

If a multinomial response variable has more than two categories, the HPGENSELECT procedure fits a model by using a cumulative link function that is based on the specified link. However, if you specify LINK=GLOGIT, the procedure assumes a generalized logit model for nominal (unordered) data, regardless of the number of response categories.

**NOCENTER**
requests that continuous main effects not be centered and scaled internally. (Continuous main effects are centered and scaled by default to aid in computing maximum likelihood estimates.) Parameter estimates and related statistics are always reported on the original scale.

**NOINT**
requests that no intercept be included in the model. (An intercept is included by default.) The NOINT option is not available in multinomial models.

**OFFSET=***variable*
specifies a *variable* to be used as an offset to the linear predictor. An offset plays the role of an effect whose coefficient is known to be 1. The offset variable cannot appear in the CLASS statement or elsewhere in the MODEL statement. Observations that have missing values for the offset variable are excluded from the analysis.

**SAMPLEFRAC=***number*
specifies a fraction of the data to be used to determine starting values for iterative estimation of the parameters of a Tweedie model. The sampled data are used in an extended quasi-likelihood estimation of the model parameters. The estimated parameters are then used as starting values in a full maximum likelihood estimation of the model parameters that uses all of the data.

**START=***n*

**START=***single-effect*

**START=(***effects***)**
begins the selection process from the designated initial model for the FORWARD and STEPWISE selection methods. If you specify START=*n*, then the starting model includes the first *n* effects that are listed in the MODEL statement. If you specify START=*single-effect* or if you specify a list of

effects within parentheses, then the starting model includes those specified effects. The effects that you specify in the START= option must be explanatory effects that are specified in the MODEL statement before the slash (/). The START= option is not available when you specify METHOD=BACKWARD in the SELECTION statement.

## OUTPUT Statement

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

The OUTPUT statement creates a data set that contains observationwise statistics that are computed after the model is fitted. The variables in the input data set are *not* included in the output data set to avoid data duplication for large data sets; however, variables that are specified in the ID statement are included.

If the input data are in distributed form, where accessing data in a particular order cannot be guaranteed, the HPGENSELECT procedure copies the distribution or partition key to the output data set so that its contents can be joined with the input data.

The computation of the output statistics is based on the final parameter estimates. If the model fit does not converge, missing values are produced for the quantities that depend on the estimates.

When there are more than two response levels for multinomial data, values are computed only for variables that are named by the LINP and PREDICTED keywords; the other variables have missing values. These statistics are computed for every response category, and the automatic variable _LEVEL_ identifies the response category on which the computed values are based. If you also specify the OBSCAT option, then the observationwise statistics are computed only for the observed response category, as indicated by the value of the _LEVEL_ variable.

For observations in which only the response variable is missing, values of the XBETA and PREDICTED statistics are computed even though these observations do not affect the model fit. For zero-inflated models, ZBETA and PZERO are also computed. This practice enables predicted mean values or predicted probabilities to be computed for new observations.

You can specify the following syntax elements in the OUTPUT statement before the slash (/).

**OUT=***SAS-data-set*

**DATA=***SAS-data-set*

specifies the name of the output data set. If the OUT= (or DATA=) option is omitted, the procedure uses the DATA*n* convention to name the output data set.

*keyword* **< =***name* **>**

specifies a statistic to include in the output data set and optionally assigns a *name* to the variable. If you do not provide a *name*, the HPGENSELECT procedure assigns a default name based on the type of statistic requested.

You can specify the following *keywords* for adding statistics to the OUTPUT data set:

**ADJPEARSON<=*name*>**

**ADJPEARS<=*name*>**

**STDRESCHI<=*name*>**

requests the Pearson residual, adjusted to have unit variance. The adjusted Pearson residual is defined for the *i*th observation as $\frac{y_i-\mu_i}{\sqrt{\phi \mathrm{V}(\mu_i)(1-h_i)}}$, where $\mathrm{V}(\mu)$ is the response distribution variance function and $h_i$ is the leverage. The leverage $h_i$ of the *i*th observation is defined as the *i*th diagonal element of the hat matrix

$$\mathbf{H} = \mathbf{W}^{\frac{1}{2}}\mathbf{X}(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}^{\frac{1}{2}}$$

where $\mathbf{W}$ is the diagonal matrix whose *i*th diagonal is $w_{ei} = \frac{w_i}{\phi \mathrm{V}(\mu_i)(g'(\mu_i))^2}$, and $w_i$ is a prior weight specified in a WEIGHT statement or 1 if no WEIGHT statement is specified. For the negative binomial, $\phi \mathrm{V}(\mu_i)$ in the denominator is replaced with the distribution variance, in both the definition of the leverage and the adjusted residual.

This statistic is not computed for multinomial models, nor is it computed for zero-modified models.

If you do not specify a *name*, PROC HPGENSELECT assigns Adjusted_Pearson as the *name*.

**LINP<=*name*>**

**XBETA<=*name*>**

requests the linear predictor $\eta = \mathbf{x}'\boldsymbol{\beta}$.

If you do not specify a *name*, PROC HPGENSELECT assigns Xbeta as the *name*.

**LOWER<=name>**

requests a lower confidence limit for the predicted value. This statistic is not computed for generalized logit multinomial models or zero-modified models.

If you do not specify a *name*, PROC HPGENSELECT assigns Lower as the *name*.

**PEARSON<=*name*>**

**PEARS<=*name*>**

**RESCHI<=*name*>**

requests the Pearson residual, $\frac{y-\mu}{\sqrt{\mathrm{V}(\mu)}}$, where $\mu$ is the estimate of the predicted response mean and $\mathrm{V}(\mu)$ is the response distribution variance function. For the negative binomial defined in the section "Negative Binomial Distribution" on page 4334 and the zero-inflated models defined in the sections "Zero-Inflated Poisson Distribution" on page 4335 and "Zero-Inflated Negative Binomial Distribution" on page 4335, the distribution variance is used in place of $\mathrm{V}(\mu)$.

This statistic is not computed for multinomial models.

If you do not specify a *name*, PROC HPGENSELECT assigns Pearson as the *name*.

**PREDICTED<=*name*>**

**PRED<=*name*>**

**P<=*name*>**

requests predicted values for the response variable.

If you do not specify a *name*, PROC HPGENSELECT assigns Pred as the *name*.

**PZERO<=*name*>**
requests zero-inflation probabilities for zero-inflated models.

If you do not specify a *name*, PROC HPGENSELECT assigns Pzero as the *name*.

**RESIDUAL<=*name*>**

**RESID<=*name*>**

**R<=*name*>**
requests the raw residual, $y - \mu$, where $\mu$ is the estimate of the predicted mean. This statistic is not computed for multinomial models.

If you do not specify a *name*, PROC HPGENSELECT assigns Residual as the *name*.

**ROLE<=*name*>**
requests a numeric variable that indicates the role played by each observation in fitting the model. Table 54.8 shows the interpretation of this variable for each observation.

**Table 54.8** Role Interpretation

| Value | Observation Role |
|---|---|
| 0 | Not used |
| 1 | Training |
| 2 | Validation |
| 3 | Testing |

If you do not partition the input data by specifying a PARTITION statement, then the role variable value is 1 for observations that are used in fitting the model and 0 for observations that have at least one missing or invalid value for the response, regressors, frequency, or weight variable.

If you do not specify a *name*, PROC HPGENSELECT assigns Role as the *name*.

**UPPER<=*name*>**
requests an upper confidence limit for the predicted value. This statistic is not computed for generalized logit multinomial models or zero-modified models.

If you do not specify a *name*, PROC HPGENSELECT assigns Upper as the *name*.

**ZBETA<=*name*>**
requests the linear predictor for the zeros model in zero-modified models: $\kappa = \mathbf{z}'\boldsymbol{\gamma}$.

If you do not specify a *name*, PROC HPGENSELECT assigns Zbeta as the *name*.

You can specify the following *options* in the OUTPUT statement after the slash (/):

**ALPHA=*number***
specifies the significance level for the construction of confidence intervals in the OUTPUT data set. The confidence level is $1 -$ *number*.

**OBSCAT**
requests (for multinomial models) that observationwise statistics be produced only for the response level. If the OBSCAT option is not specified and the response variable has $J$ levels, then the following outputs are created: for cumulative link models, $J - 1$ records are output for every observation in the input data that corresponds to the $J - 1$ lower-ordered response categories; for generalized logit models, $J$ records are output that correspond to all $J$ response categories.

## PARTITION Statement

**PARTITION** < *partition-option* > **;**

The PARTITION statement specifies how observations in the input data set are to be logically partitioned into disjoint subsets for model training, validation, and testing. For more information, see the section "Using Validation and Test Data" on page 4342. 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 specify proportions to use for random assignment of observations for each role.

You can specify one of the following mutually exclusive *partition-options*:

**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 TEST=, TRAIN=, and VALIDATE= suboptions specify the formatted values of this variable that are used to assign observations roles. If you do not specify the TRAIN= suboption, then all observations whose role is not determined by the TEST= or VALIDATE= suboption are assigned to training.

**FRACTION(< TEST=***fraction* **> < VALIDATE=***fraction* **> < SEED=***number* **>)**
randomly assigns specified proportions of the observations in the input data set to the roles. You specify the proportions for testing and validation by 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 1 and the remaining fraction of the observations are assigned to the training role. The SEED= option specifies an integer that is used to start the pseudorandom number generator for random partitioning of data for training, testing, and validation. If you do not specify a seed, or if you specify a *number* less than or equal to 0, the seed is generated by reading the time of day from the computer's clock.

## PERFORMANCE Statement

**PERFORMANCE** < *performance-options* > **;**

You can use the PERFORMANCE statement to control whether the procedure executes in single-machine or distributed mode. The default is single-machine mode.

You can also use this statement to define performance parameters for multithreaded and distributed computing, and you can request details about performance results.

The PERFORMANCE statement is documented in the section "PERFORMANCE Statement" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

## RESTRICT Statement

**RESTRICT** < *'label'* > *constraint-specification* < **,** . . . **,** *constraint-specification* >
< *operator* < *value* > > < / *option* > **;**

The RESTRICT statement enables you to specify linear equality or inequality constraints among the parameters of a model. These restrictions are incorporated into the maximum likelihood analysis.

Following are reasons why you might want to place constraints and restrictions on the model parameters:

- to fix a parameter at a particular value
- to equate parameters in a model
- to impose order conditions on the parameters in a model
- to specify contrasts among the parameters that the fitted model should honor

A restriction is composed of a left-hand side and a right-hand side, separated by an operator. If you do not specify the operator and right-hand side, the restriction is assumed to be an equality constraint against zero. If you do not specify the right-hand side, the value is assumed to be zero.

You write an individual *constraint-specification* in (nearly) the same form as you specify estimable linear functions in the ESTIMATE statement of the GLM, MIXED, or GLIMMIX procedure. The *constraint-specification* takes the form

*model-effect value-list* < . . . *model-effect value-list* >

You must specify at least one *model-effect*, followed by one or more values in the *value-list*. The values in the list correspond to the multipliers of the corresponding parameter that is associated with the position in the model effect. If you specify more values in the *value-list* than the *model-effect* occupies in the model design matrix, the extra coefficients are ignored.

The following statements provide an example. Here, A is a CLASS variable that has three levels.

```
proc hpgenselect;
   class A;
   model y/n = A x /  dist=binomial;
   restrict A 1 0 -1;
   restrict x 2 >= 0.5;
run;
```

The linear predictor for this model can be written as

$$\eta = \beta_0 + \beta_1 A_1 + \beta_2 A_2 + \beta_3 A_3 + x\beta_4$$

where $A_k$ is the binary variable associated with the $k$th level of A.

The first RESTRICT statement specifies that the parameter estimates that are associated with the first and third levels of the A effect be identical. In terms of the linear predictor, the restriction can be written as

$$\beta_1 - \beta_3 = 0$$

Because, in the default GLM parameterization, $\beta_3 = 0$, the RESTRICT statement has the effect of setting $\beta_1 = 0$.

The second RESTRICT statement involves the regression parameter associated with the variable x and specifies that the parameter estimate satisfy $\beta_4 \geq 0.25$. In terms of the linear predictor, the restriction can be written as

$$2\beta_4 \geq \frac{1}{2}$$

PROC HPGENSELECT applies both of these restrictions when it computes the maximum likelihood estimates of the regression parameters of the model.

Zero-inflated models contain two components: a model for the mean of the underlying distribution and a model for the zero-inflation probability. To specify restrictions for effects in specific components of the model, separate the *constraint-specifications* by commas. The following statements provide an example:

```
proc hpgenselect data=b itdetails itselect cov;
   class  C;
   model B = C / dist=ZIP;
   zeromodel X;
   restrict Intercept 0, X 1 = 0;
run;
```

In this example, the model for the mean has a single regressor, which is specified by the CLASS variable C. The model for the zero-inflation probability has a continuous regressor X. The RESTRICT statement specifies that the parameter estimate associated with X be constrained to be 0. The *Intercept 0 constraint-specification* serves as a placeholder and has no effect on the model for the mean. You must include this *model-effect value-list* pair in order to specify constraints on the zero-inflation part of the model. You can use any *model-effect* in the model for the mean in place of *Intercept*. For example, the following statement has the same effect, because C is in the model for the mean:

```
   restrict C 0, X 1 = 0;
```

The generalized logit model for a nominal multinomial response consists of a regression model for each nonreference level of the response variable. To specify restrictions for effects in specific components of the model, you specify a *constraint-specification* for each component to which you want to apply constraints. You specify the *constraint-specifications* in the sort order of the response variable and separate them with commas. You must specify a null *constraint-specification* with a *value-list* set to zero for each component model that has a lower response variable sort order than the one to which you want to apply constraints. The following statements provide an example. In this example, a generalized logit regression model is fit to the categorical response variable Y, with four levels. The generalized logit model consists of a regression model with a CLASS regressor Visit and a continuous regressor Lage for each level of the response variable Y. The RESTRICT statements constrain the model to have identical values of the estimated regression coefficient for Lage for all three nonreference categories of Y; that is, a common-slopes model is fit. In the second RESTRICT statement, the *constraint-specification* of *Lage 0* is necessary as a placeholder and does not affect the regression coefficient of Lage for the first level of Y.

```
proc hpgenselect data=thallMult_hgen7809;
   class Visit / Param=Ref;
   model Y=Visit Lage/dist=Multinomial link=Glogit;
   restrict Lage 1 , Lage -1;
   restrict Lage 0 , Lage  1, Lage -1;
run;
```

You can use following operators to separate the left- and right-hand sides of the restriction: **=**, **>**, **<**, **>=**, **<=**.

Some distributions involve a dispersion parameter (the parameter $\phi$ in the expressions for the log likelihood), and in the case of the Tweedie distribution, a power parameter. You cannot use the RESTRICT statement to constrain either of these parameters. Instead, you can use the MODEL statement options PHI= to set the dispersion to a fixed value and P= to set the Tweedie power parameter to a fixed value.

You can specify the following *option* after a slash (/):

**DIVISOR=***value*
specifies a *value* by which all coefficients on the right-hand and left-hand sides of the restriction are divided.

## SELECTION Statement

**SELECTION** < *options* > **;**

The SELECTION statement performs model selection by examining whether effects should be added to or removed from the model according to rules that are defined by model selection methods. The statement is fully documented in the section "SELECTION Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

The HPGENSELECT procedure supports the following effect-selection methods in the SELECTION statement:

**METHOD=NONE** results in no model selection. This method fits the full model.

**METHOD=BACKWARD** performs backward elimination. This method starts with all effects in the model and deletes effects.

**METHOD=FORWARD** performs forward selection. This method starts with no effects in the model and adds effects.

**METHOD=LASSO** performs model selection by the group LASSO method. This method adds and removes effects by using a sequence of LASSO steps.

**METHOD=STEPWISE** performs stepwise regression. This method is similar to the FORWARD method except that effects already in the model do not necessarily stay there.

For methods other than LASSO, the only effect-selection criterion that the HPGENSELECT procedure supports is SELECT=SL, in which effects enter and leave the model based on an evaluation of the significance level. To determine the level of significance for each candidate effect, PROC HPGENSELECT calculates an approximate chi-square test statistic. The SELECT= option is not supported by the LASSO method.

You can specify the following criteria in the CHOOSE= option:

**AIC** specifies Akaike's information criterion (Akaike 1974).

**AICC** specifies a small-sample bias-corrected version of Akaike's information criterion as promoted in Hurvich and Tsai (1989) and Burnham and Anderson (1998), among others.

**BIC | SBC** specifies the Schwarz Bayesian criterion (Schwarz 1978).

**VALIDATE** specifies the average squared error (ASE) that is computed from validation data, if you specify validation data by using a PARTITION statement. For more information about ASE and partitioned data, see the section "Using Validation and Test Data" on page 4342

You can specify the following criteria in the STOP= option:

**SL** specifies the significance level of the test.

**AIC** specifies Akaike's information criterion (Akaike 1974).

**AICC** specifies a small-sample bias-corrected version of Akaike's information criterion as promoted in Hurvich and Tsai (1989) and Burnham and Anderson (1998), among others.

**BIC | SBC** specifies the Schwarz Bayesian criterion (Schwarz 1978).

If you specify METHOD=LASSO and you do not specify either the CHOOSE= or STOP= option, then the model in the last LASSO step is chosen as the selected model.

The calculation of the information criteria uses the following formulas, where $p$ denotes the number of effective parameters in the candidate model, $f$ denotes the number of frequencies used, and $l$ is the log likelihood evaluated at the converged estimates:

$$\mathrm{AIC} = -2l + 2p$$

$$\mathrm{AICC} = \begin{cases} -2l + 2pf/(f-p-1) & \text{when } f > p+2 \\ -2l + 2p(p+2) & \text{otherwise} \end{cases}$$

$$\mathrm{BIC} = -2l + p\log(f)$$

If you specify the PARTITION statement, then the AIC, AICC, BIC, and SL statistics are computed on the training data set; otherwise they are computed on the full data set.

When you specify one of the following DETAILS= options in the SELECTION statement, the HPGENSELECT procedure produces the indicated tables:

**DETAILS=SUMMARY** produces a summary table that shows which effect is added or removed at each step along with the $p$-value. The summary table is produced by default if you do not specify the DETAILS= option. This option has no effect when you use the LASSO method.

**DETAILS=STEPS** produces a table of selection details that displays fit statistics for the model at each step of the selection process and the approximate log $p$-value. The summary table that results from the DETAILS=SUMMARY option is also produced. This option has no effect when you use the LASSO method.

**DETAILS=ALL** for methods other than LASSO, produces all the tables that are produced when DETAILS=STEPS and also produces a table that displays the effect that is added or removed at each step along with the $p$-value, chi-square statistic, and fit statistics for the model. For the LASSO method, it

produces a table that displays the effects that are added or removed at each step; the LASSO regularization parameter; and the AIC, AICC, and BIC fit statistics.

## WEIGHT Statement

**WEIGHT** *variable* **;**

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

## ZEROMODEL Statement

**ZEROMODEL** < *effects* > < / *zeromodel-options* > **;**

The ZEROMODEL statement defines the statistical model for zero inflation probability in terms of model effects that are constructed from variables in the input data set. An intercept term is always included in the model.

You can specify the following *zeromodel-options*.

**INCLUDE=***n*

**INCLUDE=***single-effect*

**INCLUDE=(***effects***)**

forces effects to be included in all models for zero inflation for all selection methods. If you specify INCLUDE=*n*, then the first *n* effects that are listed in the ZEROMODEL statement are included in all models. If you specify INCLUDE=*single-effect* or if you specify a list of effects within parentheses, then the specified effects are forced into all models. The effects that you specify in the INCLUDE= option must be explanatory effects that are specified in the ZEROMODEL statement before the slash (/).

**LINK=***keyword*

specifies the link function for the zero inflation probability. The *keywords* and the associated link functions are shown in Table 54.9.

**Table 54.9** Built-In Link Functions for Zero Inflation Probability

| LINK= | Link Function | $g(\mu) = \eta =$ |
|---|---|---|
| CLOGLOG \| CLL | Complementary log-log | $\log(-\log(1-\mu))$ |
| LOGIT | Logit | $\log(\mu/(1-\mu))$ |
| LOGLOG | Log-log | $-\log(-\log(\mu))$ |
| PROBIT | Probit | $\Phi^{-1}(\mu)$ |

$\Phi^{-1}(\cdot)$ denotes the quantile function of the standard normal distribution.

**START=***n*

**START=***single-effect*

**START=(***effects***)**

begins the selection process from the designated initial zero inflation model for the FORWARD and STEPWISE selection methods. If you specify START=*n*, then the starting model includes the first *n* effects that are listed in the ZEROMODEL statement. If you specify START=*single-effect* or if you specify a list of effects within parentheses, then the starting model includes those specified effects. The effects that you specify in the START= option must be explanatory effects that are specified in the ZEROMODEL statement before the slash (/). The START= option is not available when you specify METHOD=BACKWARD in the SELECTION statement.

# Details: HPGENSELECT Procedure

## Missing Values

Any observation that has missing values for the response, frequency, weight, offset, or explanatory variables is excluded from the analysis; however, missing values are valid for response and explanatory variables that are specified in the MISSING option in the CLASS statement. Observations that have a nonpositive weight or a frequency less than 1 are also excluded.

The estimated linear predictor and the fitted probabilities are not computed for any observation that has missing offset or explanatory variable values. However, if only the response value is missing, the linear predictor and the fitted probabilities can be computed and output to a data set by using the OUTPUT statement.

## Exponential Family Distributions

Many of the probability distributions that the HPGENSELECT procedure fits are members of an exponential family of distributions, which have probability distributions that are expressed as follows for some functions $b$ and $c$ that determine the specific distribution:

$$f(y) = \exp\left\{\frac{y\theta - b(\theta)}{\phi} + c(y,\phi)\right\}$$

For fixed $\phi$, this is a one-parameter exponential family of distributions. The response variable can be discrete or continuous, so $f(y)$ represents either a probability mass function or a probability density function. A more useful parameterization of generalized linear models is by the mean and variance of the distribution:

$$\begin{aligned} \mathrm{E}(Y) &= b'(\theta) \\ \mathrm{Var}(Y) &= b''(\theta)\phi \end{aligned}$$

In generalized linear models, the mean $\mu$ of the response distribution is related to linear regression parameters through a link function,

$$g(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta}$$

for the $i$th observation, where $\mathbf{x}_i$ is a fixed known vector of explanatory variables and $\boldsymbol{\beta}$ is a vector of regression parameters. The HPGENSELECT procedure parameterizes models in terms of the regression parameters $\boldsymbol{\beta}$ and either the dispersion parameter $\phi$ or a parameter that is related to $\phi$, depending on the model. For exponential family models, the distribution variance is $\mathrm{Var}(Y) = \phi \mathrm{V}(\mu)$ where $\mathrm{V}(\mu)$ is a variance function that depends only on $\mu$.

The zero-inflated models and the multinomial models are not exponential family models, but they are closely related models that are useful and are included in the HPGENSELECT procedure.

## Response Distributions

The response distribution is the probability distribution of the response (target) variable. The HPGENSELECT procedure can fit data for the following distributions:

- binary distribution
- binomial distribution
- gamma distribution
- inverse Gaussian distribution
- multinomial distribution (ordinal and nominal)
- negative binomial distribution
- normal (Gaussian) distribution
- Poisson distribution
- Tweedie distribution
- zero-inflated negative binomial distribution
- zero-inflated Poisson distribution

Expressions for the probability distributions (probability density functions for continuous variables or probability mass functions for discrete variables) are shown in the section "Response Probability Distribution Functions" on page 4332. The expressions for the log-likelihood functions of these distributions are given in the section "Log-Likelihood Functions" on page 4336.

The binary (or Bernoulli) distribution is the elementary distribution of a discrete random variable that can take on two values that have probabilities $p$ and $1 - p$. Suppose the random variable is denoted $Y$ and

$$\Pr(Y = 1) = p$$
$$\Pr(Y = 0) = 1 - p$$

The value that is associated with probability $p$ is often termed the *event* or "success"; the complementary event is termed the *non-event* or "failure." A Bernoulli experiment is a random draw from a binary distribution and generates events with probability $p$.

If $Y_1, \cdots, Y_n$ are $n$ independent Bernoulli random variables, then their sum follows a binomial distribution. In other words, if $Y_i = 1$ denotes an event (success) in the $i$th Bernoulli trial, a binomial random variable is the number of events (successes) in $n$ independent Bernoulli trials. If you use the events/trials syntax in the MODEL statement and you specify the DISTRIBUTION=BINOMIAL option, the HPGENSELECT procedure fits the model as if the data had arisen from a binomial distribution. For example, the following statements fit a binomial regression model that has regressors x1 and x2. The variables e and t represent the events and trials, respectively, for the binomial distribution:

```
proc hpgenselect;
   model e/t = x1 x2 / distribution=Binomial;
run;
```

If the events/trials syntax is used, then both variables must be numeric and the value of the events variable cannot be less than 0 or exceed the value of the trials variable. A "Response Profile" table is not produced for binomial data, because the response variable is not subject to levelization.

The multinomial distribution is a generalization of the binary distribution and allows for more than two outcome categories. Because there are more than two possible outcomes for the multinomial distribution, the terminology of "successes," "failures," "events," and "non-events" no longer applies. For multinomial data, these outcomes are generically referred to as "categories" or levels.

Whenever the HPGENSELECT procedure determines that the response variable is listed in a CLASS statement and has more than two levels (unless the events/trials syntax is used), the procedure fits the model as if the data had arisen from a multinomial distribution. By default, it is then assumed that the response categories are ordered and a cumulative link model is fit by applying the default or specified link function. If the response categories are unordered, then you should fit a generalized logit model by choosing LINK=GLOGIT in the MODEL statement.

If the response variable is not listed in a CLASS statement and a response distribution is not specified in a DISTRIBUTION= option, then a normal distribution that uses the default or specified link function is assumed.

## Response Probability Distribution Functions

### Binary Distribution

$$
\begin{aligned}
f(y) &= \begin{cases} p & \text{for } y = 1 \\ 1-p & \text{for } y = 0 \end{cases} \\
\mathrm{E}(Y) &= p \\
\mathrm{Var}(Y) &= p(1-p)
\end{aligned}
$$

### Binomial Distribution

$$\begin{aligned} f(y) &= \left( \begin{array}{c} n \\ r \end{array} \right) \mu^r (1-\mu)^{n-r} \quad \text{for } y = \frac{r}{n},\ r = 0, 1, 2, \ldots, n \\ \mathrm{E}(Y) &= \mu \\ \mathrm{Var}(Y) &= \frac{\mu(1-\mu)}{n} \end{aligned}$$

### Gamma Distribution

$$\begin{aligned} f(y) &= \frac{1}{\Gamma(\nu) y} \left( \frac{y\nu}{\mu} \right)^{\nu} \exp\left( -\frac{y\nu}{\mu} \right) \quad \text{for } 0 < y < \infty \\ \phi &= \frac{1}{\nu} \\ \mathrm{E}(Y) &= \mu \\ \mathrm{Var}(Y) &= \frac{\mu^2}{\nu} \end{aligned}$$

For the gamma distribution, $\nu = \frac{1}{\phi}$ is the estimated dispersion parameter that is displayed in the output. The parameter $\nu$ is also sometimes called the gamma index parameter.

### Inverse Gaussian Distribution

$$\begin{aligned} f(y) &= \frac{1}{\sqrt{2\pi y^3 \sigma}} \exp\left[ -\frac{1}{2y} \left( \frac{y-\mu}{\mu\sigma} \right)^2 \right] \quad \text{for } 0 < y < \infty \\ \phi &= \sigma^2 \\ \mathrm{Var}(Y) &= \phi\mu^3 \end{aligned}$$

### Multinomial Distribution

$$f(y_1, y_2, \cdots, y_k) = \frac{m!}{y_1! y_2! \cdots y_k!} p_1^{y_1} p_2^{y_2} \cdots p_k^{y_k}$$

### Negative Binomial Distribution

$$\begin{aligned} f(y) &= \frac{\Gamma(y+1/k)}{\Gamma(y+1)\Gamma(1/k)}\frac{(k\mu)^y}{(1+k\mu)^{y+1/k}} \quad \text{for } y = 0,1,2,\ldots \\ \phi &= k \\ \mathrm{E}(Y) &= \mu \\ \mathrm{Var}(Y) &= \mu + \phi\mu^2 \end{aligned}$$

For the negative binomial distribution, $k$ is the estimated dispersion parameter that is displayed in the output.

### Normal Distribution

$$\begin{aligned} f(y) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right] \quad \text{for } -\infty < y < \infty \\ \phi &= \sigma^2 \\ \mathrm{E}(Y) &= \mu \\ \mathrm{Var}(Y) &= \phi \end{aligned}$$

### Poisson Distribution

$$\begin{aligned} f(y) &= \frac{\mu^y \mathrm{e}^{-\mu}}{y!} \quad \text{for } y = 0,1,2,\ldots \\ \mathrm{E}(Y) &= \mu \\ \mathrm{Var}(Y) &= \mu \end{aligned}$$

### Tweedie Distribution

The Tweedie model is a generalized linear model from the exponential family. The Tweedie distribution is characterized by three parameters: the mean parameter $\mu$, the dispersion $\phi$, and the power $p$. The variance of the distribution is $\phi\mu^p$. For values of $p$ in the range $1 < p < 2$, a Tweedie random variable can be represented as a Poisson sum of gamma distributed random variables. That is,

$$Y = \sum_{i=1}^{N} Y_i$$

where $N$ has a Poisson distribution that has mean $\lambda = \frac{\mu^{2-p}}{\phi(2-p)}$ and the $Y_i$s have independent, identical gamma distributions, each of which has an expected value $\mathrm{E}(Y_i) = \phi(2-p)\mu^{p-1}$ and an index parameter $\nu_i = \frac{2-p}{p-1}$.

In this case, $Y$ has a discrete mass at 0, $\Pr(Y = 0) = \Pr(N = 0) = \exp(-\lambda)$, and the probability density of $Y$ $f(y)$ is represented by an infinite series for $y > 0$. The HPGENSELECT procedure restricts the power parameter to satisfy $1.1 <= p$ for numerical stability in model fitting. The Tweedie distribution does not have a general closed form representation for all values of $p$. It can be characterized in terms of the distribution mean parameter $\mu$, dispersion parameter $\phi$, and power parameter $p$. For more information about the Tweedie distribution, see Frees (2010).

The distribution mean and variance are given by:

$$\begin{aligned} \mathrm{E}(Y) &= \mu \\ \mathrm{Var}(Y) &= \phi\mu^p \end{aligned}$$

## Zero-Inflated Negative Binomial Distribution

$$\begin{aligned} f(y) &= \begin{cases} \omega + (1-\omega)(1+k\lambda)^{-\frac{1}{k}} & \text{for } y = 0 \\ (1-\omega)\frac{\Gamma(y+1/k)}{\Gamma(y+1)\Gamma(1/k)}\frac{(k\lambda)^y}{(1+k\lambda)^{y+1/k}} & \text{for } y = 1, 2, \ldots \end{cases} \\ \phi &= k \\ \mu = \mathrm{E}(Y) &= (1-\omega)\lambda \\ \mathrm{Var}(Y) &= (1-\omega)\lambda(1+\omega\lambda+k\lambda) \\ &= \mu + \left(\frac{\omega}{1-\omega} + \frac{k}{1-\omega}\right)\mu^2 \end{aligned}$$

For the zero-inflated negative binomial distribution, $k$ is the estimated dispersion parameter that is displayed in the output.

## Zero-Inflated Poisson Distribution

$$\begin{aligned} f(y) &= \begin{cases} \omega + (1-\omega)\mathrm{e}^{-\lambda} & \text{for } y = 0 \\ (1-\omega)\frac{\lambda^y \mathrm{e}^{-\lambda}}{y!} & \text{for } y = 1, 2, \ldots \end{cases} \\ \mu = \mathrm{E}(Y) &= (1-\omega)\lambda \\ \mathrm{Var}(Y) &= (1-\omega)\lambda(1+\omega\lambda) \\ &= \mu + \frac{\omega}{1-\omega}\mu^2 \end{aligned}$$

## Log-Likelihood Functions

The HPGENSELECT procedure forms the log-likelihood functions of the various models as

$$L(\boldsymbol{\mu};\mathbf{y}) = \sum_{i=1}^{n} f_i\, l(\mu_i; y_i, w_i)$$

where $l(\mu_i; y_i, w_i)$ is the log-likelihood contribution of the $i$th observation that has weight $w_i$, and $f_i$ is the value of the frequency variable. For the determination of $w_i$ and $f_i$, see the WEIGHT and FREQ statements. The individual log likelihood contributions for the various distributions are as follows.

In the following, the mean parameter $\mu_i$ for each observation $i$ is related to the regression parameters $\boldsymbol{\beta}_i$ through the linear predictor $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$ by

$$\mu_i = g^{-1}(\eta_i)$$

where $g$ is the link function.

There are two link functions and linear predictors that are associated with zero-inflated Poisson and zero-inflated negative binomial distributions: one for the zero-inflation probability $\omega$, and another for the parameter $\lambda$, which is the Poisson or negative binomial mean if there is no zero-inflation. Each of these parameters is related to regression parameters through an individual link function,

$$\begin{aligned}
\eta_i &= \mathbf{x}_i'\boldsymbol{\beta} \\
\kappa_i &= \mathbf{z}_i'\boldsymbol{\gamma} \\
\lambda_i(\boldsymbol{\beta}) &= g^{-1}(\eta_i) \\
\omega_i(\boldsymbol{\gamma}) &= h^{-1}(\kappa_i)
\end{aligned}$$

where $h$ is one of the following link functions that are associated with binary data: complementary log-log, log-log, logit, or probit. These link functions are also shown in Table 54.9.

### Binary Distribution

The HPGENSELECT procedure computes the log-likelihood function $l(\mu_i(\boldsymbol{\beta}); y_i)$ for the $i$th binary observation as

$$\begin{aligned}
\eta_i &= \mathbf{x}_i'\boldsymbol{\beta} \\
\mu_i(\boldsymbol{\beta}) &= g^{-1}(\eta_i) \\
l(\mu_i(\boldsymbol{\beta}); y_i) &= y_i \log\{\mu_i\} + (1 - y_i)\log\{1 - \mu_i\}
\end{aligned}$$

Here, $\mu_i$ is the probability of an event, and the variable $y_i$ takes on the value 1 for an event and the value 0 for a non-event. The inverse link function $g^{-1}(\cdot)$ maps from the scale of the linear predictor $\eta_i$ to the scale of the mean. For example, for the logit link (the default),

$$\mu_i(\boldsymbol{\beta}) = \frac{\exp\{\eta_i\}}{1 + \exp\{\eta_i\}}$$

You can control which binary outcome in your data is modeled as the event by specifying the *response-options* in the MODEL statement, and you can choose the link function by specifying the LINK= option in the MODEL statement.

If a WEIGHT statement is specified and $w_i$ denotes the weight for the current observation, the log-likelihood function is computed as

$$l(\mu_i(\boldsymbol{\beta}); y_i, w_i) = w_i l(\mu_i(\boldsymbol{\beta}); y_i)$$

### Binomial Distribution

The HPGENSELECT procedure computes the log-likelihood function $l(\mu_i(\boldsymbol{\beta}); y_i)$ for the $i$th binomial observation as

$$\begin{aligned}
\eta_i &= \mathbf{x}_i'\boldsymbol{\beta} \\
\mu_i(\boldsymbol{\beta}) &= g^{-1}(\eta_i) \\
l(\mu_i(\boldsymbol{\beta}); y_i, w_i) &= w_i \left(y_i \log\{\mu_i\} + (n_i - y_i)\log\{1-\mu_i\}\right) \\
&\quad + w_i \left(\log\{\Gamma(n_i+1)\} - \log\{\Gamma(y_i+1)\} - \log\{\Gamma(n_i - y_i + 1)\}\right)
\end{aligned}$$

where $y_i$ and $n_i$ are the values of the events and trials of the $i$th observation, respectively. $\mu_i$ measures the probability of events (successes) in the underlying Bernoulli distribution whose aggregate follows the binomial distribution.

### Gamma Distribution

The HPGENSELECT procedure computes the log-likelihood function $l(\mu_i(\boldsymbol{\beta}); y_i)$ for the $i$th observation as

$$\begin{aligned}
\eta_i &= \mathbf{x}_i'\boldsymbol{\beta} \\
\mu_i(\boldsymbol{\beta}) &= g^{-1}(\eta_i) \\
l(\mu_i(\boldsymbol{\beta}); y_i, w_i) &= \frac{w_i}{\phi}\log\left(\frac{w_i y_i}{\phi\mu_i}\right) - \frac{w_i y_i}{\phi\mu_i} - \log(y_i) - \log\left(\Gamma\left(\frac{w_i}{\phi}\right)\right)
\end{aligned}$$

For the gamma distribution, $\nu = \frac{1}{\phi}$ is the estimated dispersion parameter that is displayed in the output.

### Inverse Gaussian Distribution

The HPGENSELECT procedure computes the log-likelihood function $l(\mu_i(\boldsymbol{\beta}); y_i)$ for the $i$th observation as

$$\begin{aligned}
\eta_i &= \mathbf{x}_i'\boldsymbol{\beta} \\
\mu_i(\boldsymbol{\beta}) &= g^{-1}(\eta_i) \\
l(\mu_i(\boldsymbol{\beta}); y_i, w_i) &= -\frac{1}{2}\left[\frac{w_i(y_i-\mu_i)^2}{y_i\mu^2\phi} + \log\left(\frac{\phi y_i^3}{w_i}\right) + \log(2\pi)\right]
\end{aligned}$$

where $\phi$ is the dispersion parameter.

### Multinomial Distribution

The multinomial distribution that is modeled by the HPGENSELECT procedure is a generalization of the binary distribution; it is the distribution of a single draw from a discrete distribution with $J$ possible values. The log-likelihood function for the $i$th observation is

$$l(\boldsymbol{\mu}_i; \mathbf{y}_i, w_i) = w_i \sum_{j=1}^{J} y_{ij} \log\{\mu_{ij}\}$$

In this expression, $J$ denotes the number of response categories (the number of possible outcomes) and $\mu_{ij}$ is the probability that the $i$th observation takes on the response value that is associated with category $j$. The category probabilities must satisfy

$$\sum_{j=1}^{J} \mu_j = 1$$

and the constraint is satisfied by modeling $J - 1$ categories. In models that have ordered response categories, the probabilities are expressed in cumulative form, so that the last category is redundant. In generalized logit models (multinomial models that have unordered categories), one category is chosen as the reference category and the linear predictor in the reference category is set to 0.

### Negative Binomial Distribution

The HPGENSELECT procedure computes the log-likelihood function $l(\mu_i(\boldsymbol{\beta}); y_i)$ for the $i$th observation as

$$\begin{aligned}
\eta_i &= \mathbf{x}_i'\boldsymbol{\beta} \\
\mu_i(\boldsymbol{\beta}) &= g^{-1}(\eta_i) \\
l(\mu_i(\boldsymbol{\beta}); y_i, w_i) &= y_i \log\left(\frac{k\mu}{w_i}\right) - (y_i + w_i/k)\log\left(1 + \frac{k\mu}{w_i}\right) + \log\left(\frac{\Gamma(y_i + w_i/k)}{\Gamma(y_i + 1)\Gamma(w_i/k)}\right)
\end{aligned}$$

where $k$ is the negative binomial dispersion parameter that is displayed in the output.

### Normal Distribution

The HPGENSELECT procedure computes the log-likelihood function $l(\mu_i(\boldsymbol{\beta}); y_i)$ for the $i$th observation as

$$\begin{aligned}
\eta_i &= \mathbf{x}_i'\boldsymbol{\beta} \\
\mu_i(\boldsymbol{\beta}) &= g^{-1}(\eta_i) \\
l(\mu_i(\boldsymbol{\beta}); y_i, w_i) &= -\frac{1}{2}\left[\frac{w_i(y_i - \mu_i)^2}{\phi} + \log\left(\frac{\phi}{w_i}\right) + \log(2\pi)\right]
\end{aligned}$$

where $\phi$ is the dispersion parameter.

### Poisson Distribution

The HPGENSELECT procedure computes the log-likelihood function $l(\mu_i(\boldsymbol{\beta}); y_i)$ for the $i$th observation as

$$\begin{aligned}
\eta_i &= \mathbf{x}_i'\boldsymbol{\beta} \\
\mu_i(\boldsymbol{\beta}) &= g^{-1}(\eta_i) \\
l(\mu_i(\boldsymbol{\beta}); y_i, w_i) &= w_i[y_i \log(\mu_i) - \mu_i - \log(y_i!)]
\end{aligned}$$

## Tweedie Distribution

The Tweedie distribution does not in general have a closed form log-likelihood function in terms of the mean, dispersion, and power parameters. The form of the log likelihood is

$$L(\boldsymbol{\mu};\mathbf{y}) = \sum_{i=1}^{n} f_i \, l(\mu_i; y_i, w_i)$$

where

$$l(\mu_i, y_i, w_i) = \log(f(y_i; \mu_i, p, \frac{\phi}{w_i}))$$

and $f(y, \mu, p, \phi)$ is the Tweedie probability distribution, which is described in the section "Tweedie Distribution" on page 4334. Evaluation of the Tweedie log likelihood for model fitting is performed numerically as described in Dunn and Smyth (2005, 2008).

### Quasi-likelihood

The extended quasi-likelihood (EQL) is constructed according to the definition of McCullagh and Nelder (1989, Chapter 9) as

$$Q_p(\mathbf{y}, \boldsymbol{\mu}, \phi, p) = \sum_i q(y_i, \mu_i, \phi, p)$$

where the contribution from an observation is

$$q(y_i, \mu_i, \phi, p) = -0.5 \log(2\pi \frac{\phi}{w_i} y_i^p) - w_i \left( \frac{y_i^{2-p} - (2-p) y_i \mu_i^{1-p} + (1-p)\mu_i^{2-p}}{(1-p)(1-p)} \right) / \phi$$

where $1 < p < 2$. This EQL is used in computing initial values for the iterative maximization of the Tweedie log likelihood, as specified using the OPTMETHOD= Tweedie option in Table 54.5. If you specify the OPTMETHOD=EQL *Tweedie-optimization-option* in Table 54.5, then the parameter estimates are computed by using the EQL instead of the log likelihood.

## Zero-Inflated Negative Binomial Distribution

The HPGENSELECT procedure computes the log-likelihood function $l(\lambda_i(\boldsymbol{\beta}), \omega_i(\boldsymbol{\gamma}); y_i)$ for the $i$th observation as

$$\begin{aligned}
\eta_i &= \mathbf{x}_i' \boldsymbol{\beta} \\
\kappa_i &= \mathbf{z}_i' \boldsymbol{\gamma} \\
\lambda_i(\boldsymbol{\beta}) &= g^{-1}(\eta_i) \\
\omega_i(\boldsymbol{\gamma}) &= h^{-1}(\kappa_i)
\end{aligned}$$

$$l(\mu_i(\boldsymbol{\beta}), \omega_i(\boldsymbol{\gamma}); y_i, w_i) = \begin{cases} \log[\omega_i + (1-\omega_i)(1+\frac{k}{w_i}\lambda)^{-\frac{w_i}{k}}] & y_i = 0 \\ \log(1-\omega_i) + y_i \log\left(\frac{k\lambda}{w_i}\right) & \\ -(y_i + \frac{w_i}{k}) \log\left(1 + \frac{k\lambda}{w_i}\right) & \\ + \log\left(\frac{\Gamma(y_i + \frac{w_i}{k})}{\Gamma(y_i+1)\Gamma(\frac{w_i}{k})}\right) & y_i > 0 \end{cases}$$

where $k$ is the zero-inflated negative binomial dispersion parameter that is displayed in the output.

### Zero-Inflated Poisson Distribution

The HPGENSELECT procedure computes the log-likelihood function $l(\lambda_i(\boldsymbol{\beta}), \omega_i(\boldsymbol{\gamma}); y_i)$ for the $i$th observation as

$$\begin{aligned}
\eta_i &= \mathbf{x}_i'\boldsymbol{\beta} \\
\kappa_i &= \mathbf{z}_i'\boldsymbol{\gamma} \\
\lambda_i(\boldsymbol{\beta}) &= g^{-1}(\eta_i) \\
\omega_i(\boldsymbol{\gamma}) &= h^{-1}(\kappa_i) \\
l(\mu_i(\boldsymbol{\beta}), \omega_i(\boldsymbol{\gamma}); y_i, w_i) &= \begin{cases} w_i \log[\omega_i + (1-\omega_i)\exp(-\lambda_i)] & y_i = 0 \\ w_i[\log(1-\omega_i) + y_i \log(\lambda_i) - \lambda_i - \log(y_i!)] & y_i > 0 \end{cases}
\end{aligned}$$

## The LASSO Method of Model Selection

### LASSO Selection

The HPGENSELECT procedure implements the group LASSO method, which is described in the section "Group LASSO Selection" on page 4341. This section provides some background about the LASSO method that you need in order to understand the group LASSO method.

LASSO (least absolute shrinkage and selection operator) selection arises from a constrained form of ordinary least squares regression in which the sum of the absolute values of the regression coefficients is constrained to be smaller than a specified parameter. More precisely, let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$ denote the matrix of covariates, and let $\mathbf{y}$ denote the response. Then for a given parameter $t$, the LASSO regression coefficients $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)$ are the solution to the constrained least squares problem

$$\min ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 \qquad \text{subject to} \qquad \sum_{j=1}^{m} |\beta_j| \le t$$

For generalized linear models, the LASSO regression coefficients $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)$ are the solution to the constrained optimization problem

$$\min\{-L(\boldsymbol{\mu}; \mathbf{y})\} \qquad \text{subject to} \qquad \sum_{j=1}^{m} |\beta_j| \le t$$

where $L$ is the log-likelihood function defined in the section "Log-Likelihood Functions" on page 4336.

Provided that the LASSO parameter $t$ is small enough, some of the regression coefficients will be exactly zero. Hence, you can view the LASSO method 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 for which the nonzero coefficients at each step correspond to selected parameters. For more information about the LASSO method, see, for example, Hastie, Tibshirani, and Friedman (2009).

### Group LASSO Selection

The group LASSO method, proposed by Yuan and Lin (2006), is a variant of LASSO that is specifically designed for models defined in terms of effects that have multiple degrees of freedom, such as the main effects of CLASS variables, and interactions between CLASS variables. If all effects in the model are continuous, then the group LASSO method is the same as the LASSO method.

Recall that LASSO selection depends on solving a constrained optimization problem of the form

$$\min\{-L(\boldsymbol{\mu};\mathbf{y})\} \qquad \text{subject to} \quad \sum_{j=1}^{m} |\beta_j| \le t$$

In this formulation, individual parameters can be included or excluded from the model independently, subject only to the overall constraint. In contrast, the group LASSO method uses a constraint that forces all parameters corresponding to the same effect to be included or excluded simultaneously. For a model that has $k$ effects, let $\beta_{G_j}$ be the group of linear coefficients that correspond to effect $j$ in the model. Then group LASSO depends on solving a constrained optimization problem of the form

$$\min\{-L(\boldsymbol{\mu};\mathbf{y})\} \qquad \text{subject to} \quad \sum_{j=1}^{k} \sqrt{|G_j|}||\beta_{G_j}|| \le t$$

where $|G_j|$ is the number of parameters that correspond to effect $j$, and $||\beta_{G_j}||$ denotes the Euclidean norm of the parameters $\beta_{G_j}$,

$$||\beta_{G_j}|| = \sqrt{\sum_{i=1}^{G_j} \beta_i^2}$$

That is, instead of constraining the sum of the absolute value of individual parameters, group LASSO constrains the Euclidean norm of groups of parameters, where groups are defined by effects.

You can write the group LASSO method in the equivalent Lagrangian form, which is an example of a penalized log-likelihood function:

$$\min\{-L(\boldsymbol{\mu};\mathbf{y})\} + \lambda \sum_{j=1}^{k} \sqrt{|G_j|}||\beta_{G_j}||$$

The weight $\sqrt{|G_j|}$ was suggested by Yuan and Lin (2006) in order to take the size of the group into consideration in group LASSO.

Unlike LASSO for linear models, group LASSO does not allow a piecewise linear constant solution path as generated by a LAR algorithm. Instead, the method proposed by Nesterov (2013) is adopted to solve the Lagrangian form of the group LASSO problem that corresponds to a prespecified regularization parameter $\lambda$. Nesterov's method is known to have an optimal convergence rate for first-order black box optimization. Because the optimal $\lambda$ is usually unknown, a series of regularization parameters $\rho, \rho^2, \rho^3, \ldots$ is employed, where $\rho$ is a positive value less than 1. You can specify $\rho$ by using the LASSORHO= option in the PROC HPGENSELECT statement; the default value is $\rho = 0.8$. In the $i$th step of group LASSO selection, the value that is used for $\lambda$ is $\rho^i$.

A unique feature of the group LASSO method is that it does not necessarily add or remove precisely one effect at each step of the process. This is different from the forward, stepwise, and backward selection methods.

As with the other selection methods that PROC HPGENSELECT supports, you can specify a criterion to choose among the models at each step of the group LASSO algorithm by using the CHOOSE= option in the SELECTION statement. You can also specify a stopping criterion by using the STOP= option in the SELECTION statement. If you do not specify either the CHOOSE= or STOP= option, the model at the last LASSO step is chosen as the selected model, and parameter estimates are reported for this model. If you request an output data set by using an OUTPUT statement, these parameter estimates are used to compute predicted values in the output data set.

For more information, see the discussion in the section "SELECTION Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

The model degrees of freedom that PROC HPGENSELECT uses at any step of the LASSO are simply the number of nonzero regression coefficients in the model at that step. Efron et al. (2004) cite empirical evidence for doing this but do not give any mathematical justification for this choice.

Some distributions involve a dispersion parameter (the parameter $\phi$ in the expressions for the log likelihood), and in the case of the Tweedie distribution, a power parameter. These parameters are not estimated by the LASSO optimization algorithm, and are set to either the default value or a value that you specify. You can use the MODEL statement options PHI= to set the dispersion to a fixed value and P= to set the Tweedie power parameter to a fixed value.

## Using Validation and Test Data

When you have sufficient data, you can divide your data into three parts, which are called the training, validation, and test data. For a single model fit or during the model selection process, models are fit and selected based on the training data. After a model has been fit, the validation and test sets can be used to assess how the selected model generalizes on data that played no role in selecting the model. For example, Hastie, Tibshirani, and Friedman (2009) advocate using validation data in the model selection process to determine which effects to include in each step and when to terminate the selection process. PROC HPGENSELECT does not currently use validation data in this way, so the validation and test data subsets are equivalent.

You can use validation and test data to score data that were not used in fitting the model. Statistics in an output data set that is created by an OUTPUT statement are computed for validation and test data, using the model fit based on the training data. You can use the ROLE option in an OUTPUT statement to add a variable (named Role by default) to an output data set to indicate the role played by each observation.

You use a PARTITION statement to logically divide the DATA= data set into separate roles. You can specify the fractions of the data that you want to reserve as test data and validation data. For example, the following statements randomly divide the inData data set, reserving 50% for training and 25% each for validation and testing:

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

In some cases you might need to exercise more control over the partitioning of the input data set. You can do this by naming both a variable in the input data set and a formatted value of that variable that corresponds to each role. For example, the following statements assign roles to the observations in the inData data set based on the value of the variable group in that data set. Observations whose value of Group is `'group 1'` are assigned for testing, and those whose value is `'group 2'` are assigned to training. All other observations are ignored.

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

When you have reserved observations for training, validation, and testing, a model that is fit on the training data is scored on the validation and test data, and fit statistics, including the average squared error (ASE), are computed separately for each of these subsets. The ASE for each data role is the sum of the squared differences between the responses and the predictions for observations in that role divided by the number of observations in that role.

## Computational Method: Multithreading

Threading refers to the organization of computational work into multiple tasks (processing units that can be scheduled by the operating system). A task is associated with a thread. Multithreading refers to the concurrent execution of threads. When multithreading is possible, substantial performance gains can be realized compared to sequential (single-threaded) execution.

The number of threads spawned by the HPGENSELECT procedure is determined by the number of CPUs on a machine and can be controlled in the following ways:

- You can specify the number of CPUs in the CPUCOUNT= SAS system option. For example, if you specify the following statement, the HPGENSELECT procedure determines threading as if it executed on a system that has four CPUs, regardless of the actual CPU count:

  ```
  options cpucount=4;
  ```

- You can specify the NTHREADS= option in the PERFORMANCE statement to control the number of threads. This specification overrides the CPUCOUNT= system option. Specify NTHREADS=1 to force single-threaded execution.

The number of threads per machine is displayed in the “Dimensions” table, which is part of the default output. The HPGENSELECT procedure allocates one thread per CPU by default.

The tasks that are multithreaded by the HPGENSELECT procedure are primarily defined by dividing the data that are processed on a single machine among the threads—that is, the HPGENSELECT procedure implements multithreading through a data-parallel model. For example, if the input data set has 1,000 observations and PROC HPGENSELECT is running with four threads, then 250 observations are associated with each thread. All operations that require access to the data are then multithreaded. These operations include the following:

- variable levelization
- effect levelization
- formation of the initial crossproducts matrix
- formation of approximate Hessian matrices for candidate evaluation during model selection
- objective function calculation
- gradient calculation
- Hessian calculation
- scoring of observations

In addition, operations on matrices such as sweeps can be multithreaded provided that the matrices are of sufficient size to realize performance benefits from managing multiple threads for the particular matrix operation.

## Choosing an Optimization Algorithm

### First- or Second-Order Algorithms

The factors that affect how you choose an optimization technique for a particular problem are complex. Although the default method works well for most problems, you might occasionally benefit from trying several different algorithms.

For many optimization problems, computing the gradient takes more computer time than computing the function value. Computing the Hessian sometimes takes *much* more computer time and memory than computing the gradient, especially when there are many decision variables. Unfortunately, optimization techniques that do not use some kind of Hessian approximation usually require many more iterations than techniques that do use a Hessian matrix; as a result, the total run time of these techniques is often longer. Techniques that do not use the Hessian also tend to be less reliable. For example, they can terminate more easily at stationary points than at global optima.

Table 54.10 shows which derivatives are required for each optimization technique.

**Table 54.10** Derivatives Required

| Algorithm | First-Order | Second-Order |
|---|---|---|
| TRUREG | x | x |
| NEWRAP | x | x |
| NRRIDG | x | x |
| QUANEW | x | - |
| DBLDOG | x | - |
| CONGRA | x | - |
| NMSIMP | - | - |

The second-derivative methods TRUREG, NEWRAP, and NRRIDG are best for small problems for which the Hessian matrix is not expensive to compute. Sometimes the NRRIDG algorithm can be faster than the TRUREG algorithm, but TRUREG can be more stable. The NRRIDG algorithm requires only one matrix with $p(p+1)/2$ double words; TRUREG and NEWRAP require two such matrices. Here, $p$ denotes the number of parameters in the optimization.

The first-derivative methods QUANEW and DBLDOG are best for medium-sized problems for which the objective function and the gradient can be evaluated much faster than the Hessian. In general, the QUANEW and DBLDOG algorithms require more iterations than TRUREG, NRRIDG, and NEWRAP, but each iteration can be much faster. The QUANEW and DBLDOG algorithms require only the gradient to update an approximate Hessian, and they require slightly less memory than TRUREG or NEWRAP.

The first-derivative method CONGRA is best for large problems for which the objective function and the gradient can be computed much faster than the Hessian and for which too much memory is required to store the (approximate) Hessian. In general, the CONGRA algorithm requires more iterations than QUANEW or DBLDOG, but each iteration can be much faster. Because CONGRA requires only a factor of $p$ double-word memory, many large applications can be solved only by CONGRA.

The no-derivative method NMSIMP is best for small problems for which derivatives are not continuous or are very difficult to compute.

Each optimization method uses one or more convergence criteria that determine when it has converged. An algorithm is considered to have converged when any one of the convergence criteria is satisfied. For example, under the default settings, the QUANEW algorithm converges if ABSGCONV <1E–5, FCONV $< 2 \times \epsilon$, or GCONV <1E–8.

By default, the HPGENSELECT procedure applies the NRRIDG algorithm because it can take advantage of multithreading in Hessian computations and inversions. If the number of parameters becomes large, specifying the TECHNIQUE=QUANEW option (which is a first-order method with good overall properties), is recommended.

## Algorithm Descriptions

The following subsections provide details about each optimization technique and follow the same order as Table 54.10.

### *Trust Region Optimization (TRUREG)*

The trust region method uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\psi}^{(k)})$; thus, it requires that the objective function $f(\boldsymbol{\psi})$ have continuous first- and second-order derivatives inside the feasible region.

The trust region method iteratively optimizes a quadratic approximation to the nonlinear objective function within a hyperelliptic trust region with radius $\Delta$ that constrains the step size that corresponds to the quality of the quadratic approximation. The trust region method is implemented based on Dennis, Gay, and Welsch (1981); Gay (1983); Moré and Sorensen (1983).

The trust region method performs well for small- to medium-sized problems, and it does not need many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, one of the dual quasi-Newton or conjugate gradient algorithms might be more efficient.

#### Newton-Raphson Optimization with Line Search (NEWRAP)

The NEWRAP technique uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\psi}^{(k)})$; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region.

If second-order derivatives are computed efficiently and precisely, the NEWRAP method can perform well for medium-sized to large problems, and it does not need many function, gradient, and Hessian calls.

This algorithm uses a pure Newton step when the Hessian is positive-definite and when the Newton step reduces the value of the objective function successfully. Otherwise, a combination of ridging and line search is performed to compute successful steps. If the Hessian is not positive-definite, a multiple of the identity matrix is added to the Hessian matrix to make it positive-definite (Eskow and Schnabel 1991).

In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The line-search method uses quadratic interpolation and cubic extrapolation.

#### Newton-Raphson Ridge Optimization (NRRIDG)

The NRRIDG technique uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\psi}^{(k)})$; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region.

This algorithm uses a pure Newton step when the Hessian is positive-definite and when the Newton step reduces the value of the objective function successfully. If at least one of these two conditions is not satisfied, a multiple of the identity matrix is added to the Hessian matrix.

Because the NRRIDG technique uses an orthogonal decomposition of the approximate Hessian, each iteration of NRRIDG can be slower than an iteration of the NEWRAP technique, which works with a Cholesky decomposition. However, NRRIDG usually requires fewer iterations than NEWRAP.

The NRRIDG method performs well for small- to medium-sized problems, and it does not require many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, one of the dual quasi-Newton or conjugate gradient algorithms might be more efficient.

#### Quasi-Newton Optimization (QUANEW)

The dual quasi-Newton method uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$, and it does not need to compute second-order derivatives because they are approximated. It works well for medium-sized to moderately large optimization problems, where the objective function and the gradient can be computed much faster than the Hessian. However, in general the QUANEW technique requires more iterations than the TRUREG, NEWRAP, and NRRIDG techniques, which compute second-order derivatives. The QUANEW technique provides an appropriate balance between the speed and stability that are required for most generalized linear model applications.

The QUANEW technique that is implemented by the HPGENSELECT procedure is the dual quasi-Newton algorithm, which updates the Cholesky factor of an approximate Hessian.

In each iteration, a line search is performed along the search direction to find an approximate optimum. The line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size $\alpha$ that satisfies the Goldstein conditions (Fletcher 1987). One of the Goldstein conditions can be violated if the feasible region defines an upper limit of the step size. Violating the left-side Goldstein condition can affect the positive-definiteness of the quasi-Newton update. In that case, either the update is skipped or the iterations are restarted by using an identity matrix, resulting in the steepest descent or ascent search direction.

### Double-Dogleg Optimization (DBLDOG)

The double-dogleg optimization method combines the ideas of the quasi-Newton and trust region methods. In each iteration, the double-dogleg algorithm computes the step $s^{(k)}$ as the linear combination of the steepest descent or ascent search direction $s_1^{(k)}$ and a quasi-Newton search direction $s_2^{(k)}$:

$$s^{(k)} = \alpha_1 s_1^{(k)} + \alpha_2 s_2^{(k)}$$

The step is requested to remain within a prespecified trust region radius (Fletcher 1987, p. 107). Thus, the DBLDOG subroutine uses the dual quasi-Newton update but does not perform a line search.

The double-dogleg optimization technique works well for medium-sized to moderately large optimization problems, where the objective function and the gradient can be computed much faster than the Hessian. The implementation is based on Dennis and Mei (1979); Gay (1983), but it is extended for dealing with boundary and linear constraints. The DBLDOG technique generally requires more iterations than the TRUREG, NEWRAP, and NRRIDG techniques, which require second-order derivatives; however, each of the DBLDOG iterations is computationally cheap. Furthermore, the DBLDOG technique requires only gradient calls for the update of the Cholesky factor of an approximate Hessian.

### Conjugate Gradient Optimization (CONGRA)

Second-order derivatives are not required by the CONGRA algorithm and are not even approximated. The CONGRA algorithm can be expensive in function and gradient calls, but it requires only $O(p)$ memory for unconstrained optimization. In general, the algorithm must perform many iterations to obtain a precise solution, but each of the CONGRA iterations is computationally cheap.

The CONGRA algorithm should be used for optimization problems that have large $p$. For the unconstrained or boundary-constrained case, the CONGRA algorithm requires only $O(p)$ bytes of working memory, whereas all other optimization methods require order $O(p^2)$ bytes of working memory. During $p$ successive iterations, uninterrupted by restarts or changes in the working set, the CONGRA algorithm computes a cycle of $p$ conjugate search directions. In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size $\alpha$ that satisfies the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit for the step size.

### Nelder-Mead Simplex Optimization (NMSIMP)

The Nelder-Mead simplex method does not use any derivatives and does not assume that the objective function has continuous derivatives. The objective function itself needs to be continuous. This technique is quite expensive in the number of function calls, and it might be unable to generate precise results for $p \gg 40$.

The original Nelder-Mead simplex algorithm is implemented and extended to boundary constraints. This algorithm does not compute the objective for infeasible points, but it changes the shape of the simplex adapting to the nonlinearities of the objective function. This change contributes to an increased speed of convergence and uses a special termination criterion.

## Displayed Output

The following sections describe the output that PROC HPGENSELECT produces by default. The output is organized into various tables, which are discussed in the order of their appearance.

### Performance Information

The "Performance Information" table is produced by default. It displays information about the execution mode. For single-machine mode, the table displays the number of threads used. For distributed mode, the table displays the grid mode (symmetric or asymmetric), the number of compute nodes, and the number of threads per node.

If you specify the DETAILS option in the PERFORMANCE statement, the procedure also produces a "Timing" table in which elapsed times (absolute and relative) for the main tasks of the procedure are displayed.

### Model Information

The "Model Information" table displays basic information about the model, such as the response variable, frequency variable, link function, and the model category that the HPGENSELECT procedure determined based on your input and options. The "Model Information" table also displays the distribution of the data that is assumed by the HPGENSELECT procedure. For information about how the procedure determines the response distribution, see the section "Response Distributions" on page 4331.

### Class Level Information

The "Class Level Information" table lists the levels of every variable that is specified in the CLASS statement. You should check this information to make sure that the data are correct. You can adjust the order of the CLASS variable levels by specifying the ORDER= option in the CLASS statement. You can suppress the "Class Level Information" table completely or partially by specifying the NOCLPRINT= option in the PROC HPGENSELECT statement.

If the classification variables use reference parameterization, the "Class Level Information" table also displays the reference value for each variable.

### Number of Observations

The "Number of Observations" table displays the number of observations that are read from the input data set and the number of observations that are used in the analysis. If a FREQ statement is present, the sum of the frequencies read and used is displayed. If the events/trials syntax is used, the number of events and trials is also displayed. If a PARTITION statement is specified, the table displays the values for each role.

### Response Profile

The "Response Profile" table displays the ordered value from which the HPGENSELECT procedure determines the probability being modeled as an event in binary models and the ordering of categories in multinomial models. For each response category level, the frequency that is used in the analysis is reported. You can affect the ordering of the response values by specifying *response-options* in the MODEL statement. For binary and generalized logit models, the note that follows the "Response Profile" table indicates which outcome is modeled as the event in binary models and which value serves as the reference category.

The "Response Profile" table is not produced for binomial data. You can find information about the number of events and trials in the "Number of Observations" table. If a PARTITION statement is specified, the table displays the values for each role.

### Entry and Removal Candidates

When you specify the DETAILS=ALL or DETAILS=STEPS option in the SELECTION statement, the HPGENSELECT procedure produces "Entry Candidates" and "Removal Candidates" tables that display the effect names and the values of the criterion that is 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.

### Selection Information

When you specify the SELECTION statement, the HPGENSELECT procedure produces by default a series of tables that have information about the model selection. The "Selection Information" table informs you about the model selection method, selection and stop criteria, and other parameters that govern the selection. You can suppress this table by specifying DETAILS=NONE in the SELECTION statement.

### Selection Summary

When you specify the SELECTION statement, the HPGENSELECT procedure produces the "Selection Summary" table, which contains information about which effects were entered into or removed from the model at the steps of the model selection process. The *p*-value for the score chi-square test that led to the removal or entry decision is also displayed. You can request further details about the model selection steps by specifying DETAILS=STEPS or DETAILS=ALL in the SELECTION statement. You can suppress the display of the "Selection Summary" table by specifying DETAILS=NONE in the SELECTION statement.

### Selection Details

When you specify the DETAILS=ALL option in the SELECTION statement, the HPGENSELECT procedure produces the "Selection Details" table, which contains information about which effects were entered into or removed from the model at the steps of the model selection process. When you specify METHOD=FORWARD, BACKWARD, or STEPWISE, the *p*-value and the chi-square test statistic that led to the removal or entry decision are also displayed. Fit statistics for the model at the steps are also displayed. When you specify METHOD=LASSO, fit statistics for the model at the steps are displayed.

### Stop Reason

When you specify the SELECTION statement, the HPGENSELECT procedure produces a simple table that tells you why model selection stopped.

### Selection Reason

When you specify the SELECTION statement, the HPGENSELECT procedure produces a simple table that tells you why the final model was selected.

### Selected Effects

When you specify the SELECTION statement, the HPGENSELECT procedure produces a simple table that tells you which effects were selected to be included in the final model.

### Iteration History

For each iteration of the optimization, the "Iteration History" table displays the number of function evaluations (including gradient and Hessian evaluations), the value of the objective function, the change in the objective function from the previous iteration, and the absolute value of the largest (projected) gradient element. The objective function used in the optimization in the HPGENSELECT procedure is normalized by default to enable comparisons across data sets that have different sampling intensity. You can control normalization by specifying the NORMALIZE= option in the PROC HPGENSELECT statement.

If you specify the ITDETAILS option in the PROC HPGENSELECT statement, information about the parameter estimates and gradients in the course of the optimization is added to the "Iteration History" table. To generate the history from a model selection process, specify the ITSELECT option.

### Convergence Status

The convergence status table is a small ODS table that follows the "Iteration History" table in the default output. In the listing it appears as a message that indicates whether the optimization succeeded and which convergence criterion was met. If the optimization fails, the message indicates the reason for the failure. If you save the convergence status table to an output data set, a numeric Status variable is added that enables you to programmatically assess convergence. The values of the Status variable encode the following:

0 Convergence was achieved, or an optimization was not performed because TECHNIQUE=NONE is specified.

1 The objective function could not be improved.

2 Convergence was not achieved because of a user interrupt or because a limit (such as the maximum number of iterations or the maximum number of function evaluations) was reached. To modify these limits, see the MAXITER=, MAXFUNC=, and MAXTIME= options in the PROC HPGENSELECT statement.

3 Optimization failed to converge because function or derivative evaluations failed at the starting values or during the iterations or because a feasible point that satisfies the parameter constraints could not be found in the parameter space.

### Dimensions

The "Dimensions" table displays size measures that are derived from the model and the environment. It displays the number of effects in the model, the number of columns in the design matrix, and the number of parameters for which maximum likelihood estimates are computed.

### Optimization Stage Details

The "Optimization Stage Details" table displays the optimization stages that are used to fit Tweedie models. The type of optimization, the percentage of observations used, and the number of observations used are displayed for each stage.

### Fit Statistics

The "Fit Statistics" table displays a variety of likelihood-based measures of fit. All statistics are presented in "smaller is better" form.

The calculation of the information criteria uses the following formulas, where $p$ denotes the number of effective parameters, $f$ denotes the number of frequencies used, and $l$ is the log likelihood evaluated at the converged estimates:

$$\begin{aligned}
\text{AIC} &= -2l + 2p \\
\text{AICC} &= \begin{cases} -2l + 2pf/(f-p-1) & \text{when } f > p+2 \\ -2l + 2p(p+2) & \text{otherwise} \end{cases} \\
\text{BIC} &= -2l + p\log(f)
\end{aligned}$$

If no FREQ statement is given, $f$ equals $n$, the number of observations used.

If a PARTITION statement is specified, the table displays the values for each role. In addition, the average squared error (ASE) is computed separately for each role. The ASE for each data role is the sum of the squared differences between the responses and the predictions for observations in that role divided by the number of observations in that role.

The values displayed in the "Fit Statistics" table are not based on a normalized log-likelihood function.

### Parameter Estimates

The "Parameter Estimates" table displays the parameter estimates, their estimated (asymptotic) standard errors, chi-square statistics, and $p$-values for the hypothesis that the parameter is 0.

If you request confidence intervals by specifying the CL option in the MODEL statement, confidence limits for regression parameters are produced for the estimate on the linear scale. Confidence limits for the dispersion parameter of those distributions that possess a dispersion parameter are produced on the log scale, because the dispersion must be greater than 0. Similarly, confidence limits for the power parameter of the Tweedie distribution are produced on the log scale.

### Parameter Estimates Correlation Matrix

When you specify the CORR option in the PROC HPGENSELECT statement, the correlation matrix of the parameter estimates is displayed.

### Parameter Estimates Covariance Matrix

When you specify the COV option in the PROC HPGENSELECT statement, the covariance matrix of the parameter estimates is displayed. The covariance matrix is computed as the inverse of the negative of the matrix of second derivatives of the log-likelihood function with respect to the model parameters (the Hessian matrix), evaluated at the parameter estimates.

### Zero-Inflation Parameter Estimates

The parameter estimates for zero-inflation probability in zero-inflated models, their estimated (asymptotic) standard errors, chi-square statistics, and $p$-values for the hypothesis that the parameter is 0 are presented in the "Parameter Estimates" table. If you request confidence intervals by specifying the CL option in the MODEL statement, confidence limits for regression parameters are produced for the estimate on the linear scale.

## ODS Table Names

Each table created by the HPGENSELECT procedure has a name that is associated with it, and you must use this name to refer to the table when you use ODS statements. These names are listed in Table 54.11.

**Table 54.11** ODS Tables Produced by PROC HPGENSELECT

| Table Name | Description | Required Statement and Option |
|---|---|---|
| ClassLevels | Level information from the CLASS statement | CLASS |
| ConvergenceStatus | Status of optimization at conclusion of optimization | Default output |
| CorrelationMatrix | Correlation matrix of parameter estimates | PROC HPGENSELECT CORR |
| CovarianceMatrix | Covariance matrix of parameter estimates | PROC HPGENSELECT COV |
| Dimensions | Model dimensions | Default output |
| EntryCandidates | Candidates for entry at step | SELECTION DETAILS=ALL \| STEPS |
| FitStatistics | Fit statistics | Default output |
| IterHistory | Iteration history | PROC HPGENSELECT ITDETAILS or PROC HPGENSELECT ITSUMMARY or PROC HPGENSELECT ITSELECT |
| LassoSelectionDetails | Details about model selection by LASSO, including fit statistics by step | SELECTION DETAILS=ALL |
| ModelInfo | Information about the modeling environment | Default output |
| NObs | Number of observations read and used, and number of events and trials, if applicable | Default output |
| OptimizationStages | Optimization stages that are used to fit Tweedie models | MODEL DISTRIBUTION=TWEEDIE |
| ParameterEstimates | Solutions for the parameter estimates that are associated with effects in MODEL statements | Default output |
| PerformanceInfo | Information about the high-performance computing environment | Default output |

**Table 54.11** *continued*

| Table Name | Description | Required Statement / Option |
|---|---|---|
| Regularization | Maximum regularization parameter used in penalized log likelihood for LASSO model selection and regularization parameter of the chosen model | SELECTION METHOD=LASSO |
| RemovalCandidates | Candidates for removal at step | SELECTION DETAILS=ALL \| STEPS |
| ResponseProfile | Response categories and the category that is modeled in models for binary and multinomial data | Default output |
| SelectedEffects | List of effects that are selected to be included in model | SELECTION |
| SelectionDetails | Details about model selection, including fit statistics by step | SELECTION DETAILS=ALL |
| SelectionInfo | Information about the settings for model selection | SELECTION |
| SelectionReason | Reason why the particular model was selected | SELECTION |
| SelectionSummary | Summary information about model selection steps | SELECTION |
| StopReason | Reason for termination of model selection | SELECTION |
| Timing | Absolute and relative times for tasks performed by the procedure | PERFORMANCE DETAILS |
| ZeroParameterEstimates | Solutions for the parameter estimates that are associated with effects in ZEROMODEL statements | ZEROMODEL |

# Examples: HPGENSELECT Procedure

## Example 54.1: Model Selection

The following HPGENSELECT statements examine the same data that is used in the section "Getting Started: HPGENSELECT Procedure" on page 4299, but they request model selection via the forward selection technique. Model effects are added in the order of their significance until no more effects make a significant improvement of the current model. The DETAILS=ALL option in the SELECTION statement requests that all tables that are related to model selection be produced.

The data set getStarted is shown in the section "Getting Started: HPGENSELECT Procedure" on page 4299. It contains 100 observations on a count response variable (Y), a continuous variable (Total) to be used in Example 54.3, and five categorical variables (C1–C5), each of which has four numerical levels.

A log-linked Poisson regression model is specified by using classification effects for variables C1–C5. The following statements request model selection by using the forward selection method:

```
proc hpgenselect data=getStarted;
   class C1-C5;
   model Y = C1-C5 / Distribution=Poisson;
   selection method=forward details=all;
run;
```

The model selection tables are shown in Output 54.1.1 through Output 54.1.3.

The "Selection Information" table in Output 54.1.1 summarizes the settings for the model selection. Effects are added to the model only if they produce a significant improvement as judged by comparing the $p$-value of a score test to the entry significance level (SLE), which is 0.05 by default. The forward selection stops when no effect outside the model meets this criterion.

**Output 54.1.1** Selection Information

**The HPGENSELECT Procedure**

| Selection Information | |
|---|---|
| **Selection Method** | Forward |
| **Select Criterion** | Significance Level |
| **Stop Criterion** | Significance Level |
| **Effect Hierarchy Enforced** | None |
| **Entry Significance Level (SLE)** | 0.05 |
| **Stop Horizon** | 1 |

The "Selection Summary" table in Output 54.1.2 shows the effects that were added to the model and their significance level. Step 0 refers to the null model that contains only an intercept. In the next step, effect C2 made the most significant contribution to the model among the candidate effects ($p < 0.0001$). In step 2, the most significant contribution when adding an effect to a model that contains the intercept and C2 was made by C5. In step 3, the variable C1 ($p = 0.0496$) was added. In the subsequent step, no effect could be added to the model that would produce a $p$-value less than 0.05, so variable selection stops.

**Output 54.1.2** Selection Summary Information

**The HPGENSELECT Procedure**

| Selection Summary | | | |
|---|---|---|---|
| **Step** | **Effect Entered** | **Number Effects In** | **p Value** |
| **0** | **Intercept** | 1 | . |
| **1** | **C2** | 2 | <.0001 |
| **2** | **C5** | 3 | <.0001 |
| **3** | **C1** | 4 | 0.0496 |

Selection stopped because no candidate for entry is significant at the 0.05 level.

**Selected Effects:** Intercept C1 C2 C5

The DETAILS=ALL option produces the "Selection Details" table, which provides fit statistics and the value

of the score test chi-square statistic at each step.

**Output 54.1.3** Selection Details

**Selection Details**

| Step | Description | Effects In Model | Chi-Square | Pr > ChiSq | -2 LogL | AIC | AICC | BIC |
|---|---|---|---|---|---|---|---|---|
| **0** | Initial Model | 1 | | | 350.193 | 352.193 | 352.234 | 354.798 |
| **1** | C2 entered | 2 | 25.7340 | <.0001 | 324.611 | 332.611 | 333.032 | 343.032 |
| **2** | C5 entered | 3 | 23.0291 | <.0001 | 303.580 | 317.580 | 318.798 | 335.817 |
| **3** | C1 entered | 4 | 7.8328 | 0.0496 | 295.263 | 315.263 | 317.735 | 341.315 |

Output 54.1.4 displays information about the selected model. Notice that the –2 log likelihood value in the "Fit Statistics" table is larger than the value for the full model in Figure 54.7. This is expected because the selected model contains only a subset of the parameters. Because the selected model is more parsimonious than the full model, the information criteria AIC, AICC and BIC are smaller than in the full model, indicating a better fit.

**Output 54.1.4** Fit Statistics

| Fit Statistics | |
|---|---|
| **-2 Log Likelihood** | 295.26 |
| **AIC (smaller is better)** | 315.26 |
| **AICC (smaller is better)** | 317.74 |
| **BIC (smaller is better)** | 341.31 |
| **Pearson Chi-Square** | 85.0656 |
| **Pearson Chi-Square/DF** | 0.9452 |

The parameter estimates of the selected model are given in Output 54.1.5. Notice that the effects are listed in the "Parameter Estimates" table in the order in which they were specified in the MODEL statement and not in the order in which they were added to the model.

**Output 54.1.5** Parameter Estimates

**Parameter Estimates**

| Parameter | DF | Estimate | Standard Error | Chi-Square | Pr > ChiSq |
|---|---|---|---|---|---|
| **Intercept** | 1 | 0.775498 | 0.242561 | 10.2216 | 0.0014 |
| **C1 0** | 1 | -0.211240 | 0.207209 | 1.0393 | 0.3080 |
| **C1 1** | 1 | -0.685575 | 0.255713 | 7.1879 | 0.0073 |
| **C1 2** | 1 | -0.127612 | 0.203663 | 0.3926 | 0.5309 |
| **C1 3** | 0 | 0 | . | . | . |
| **C2 0** | 1 | 0.958378 | 0.239731 | 15.9817 | <.0001 |
| **C2 1** | 1 | 0.738529 | 0.237098 | 9.7024 | 0.0018 |
| **C2 2** | 1 | 0.211075 | 0.255791 | 0.6809 | 0.4093 |
| **C2 3** | 0 | 0 | . | . | . |
| **C5 0** | 1 | -0.825545 | 0.214054 | 14.8743 | 0.0001 |
| **C5 1** | 1 | -0.697611 | 0.202607 | 11.8555 | 0.0006 |
| **C5 2** | 1 | -0.566706 | 0.213961 | 7.0153 | 0.0081 |
| **C5 3** | 0 | 0 | . | . | . |

## Example 54.2: Modeling Binomial Data

If $Y_1, \cdots, Y_n$ are independent binary (Bernoulli) random variables that have common success probability $\pi$, then their sum is a binomial random variable. In other words, a binomial random variable that has parameters $n$ and $\pi$ can be generated as the sum of $n$ Bernoulli($\pi$) random experiments. The HPGENSELECT procedure uses a special syntax to express data in binomial form: the *events/trials* syntax.

Consider the following data, taken from Cox and Snell (1989, pp. 10–11), of the number, r, of ingots not ready for rolling, out of n tested, for a number of combinations of heating time and soaking time.

```
data Ingots;
   input Heat Soak r n @@;
   Obsnum= _n_;
   datalines;
7 1.0 0 10  14 1.0 0 31  27 1.0 1 56  51 1.0 3 13
7 1.7 0 17  14 1.7 0 43  27 1.7 4 44  51 1.7 0  1
7 2.2 0  7  14 2.2 2 33  27 2.2 0 21  51 2.2 0  1
7 2.8 0 12  14 2.8 0 31  27 2.8 1 22  51 4.0 0  1
7 4.0 0  9  14 4.0 0 19  27 4.0 1 16
;
```

If each test is carried out independently and if for a particular combination of heating and soaking time there is a constant probability that the tested ingot is not ready for rolling, then the random variable $r$ follows a Binomial($n, \pi$) distribution, where the success probability $\pi$ is a function of heating and soaking time.

The following statements show the use of the events/trials syntax to model the binomial response. The *events* variable in this situation is r (the number of ingots not ready for rolling), and the *trials* variable is n (the number of ingots tested). The dependency of the probability of not being ready for rolling is modeled as a function of heating time, soaking time, and their interaction. The OUTPUT statement stores the linear predictors and the predicted probabilities in the Out data set along with the ID variable.

```
proc hpgenselect data=Ingots;
   model r/n = Heat Soak Heat*Soak / dist=Binomial;
   id Obsnum;
   output out=Out xbeta predicted=Pred;
run;
```

The "Performance Information" table in Output 54.2.1 shows that the procedure executes in single-machine mode.

**Output 54.2.1** Performance Information

**The HPGENSELECT Procedure**

| Performance Information | |
|---|---|
| **Execution Mode** | Single-Machine |
| **Number of Threads** | 4 |

The "Model Information" table shows that the data are modeled as binomially distributed with a logit link function (Output 54.2.2). This is the default link function in the HPGENSELECT procedure for binary and binomial data. The procedure uses a ridged Newton-Raphson algorithm to estimate the parameters of the model.

**Output 54.2.2** Model Information and Number of Observations

| Model Information | |
|---|---|
| **Data Source** | WORK.INGOTS |
| **Response Variable (Events)** | r |
| **Response Variable (Trials)** | n |
| **Distribution** | Binomial |
| **Link Function** | Logit |
| **Optimization Technique** | Newton-Raphson with Ridging |

| | |
|---|---|
| **Number of Observations Read** | 19 |
| **Number of Observations Used** | 19 |
| **Number of Events** | 12 |
| **Number of Trials** | 387 |

The second table in Output 54.2.2 shows that all 19 observations in the data set were used in the analysis and that the total number of events and trials equal 12 and 387, respectively. These are the sums of the variables r and n across all observations.

Output 54.2.3 displays the "Dimensions" table for the model. There are four columns in the design matrix of the model (the **X** matrix); they correspond to the intercept, the Heat effect, the Soak effect, and the interaction of the Heat and Soak effects. The model is nonsingular, because the rank of the crossproducts matrix equals the number of columns in **X**. All parameters are estimable and participate in the optimization.

**Output 54.2.3** Dimensions in Binomial Logistic Regression

| Dimensions | |
|---|---|
| **Number of Effects** | 4 |
| **Number of Parameters** | 4 |
| **Columns in X** | 4 |

Output 54.2.4 displays the "Fit Statistics" table for this run. Evaluated at the converged estimates, –2 times the value of the log-likelihood function equals 27.9569. Further fit statistics are also given, all of them in "smaller is better" form. The AIC, AICC, and BIC criteria are used to compare non-nested models and to penalize the model fit for the number of observations and parameters. The –2 log-likelihood value can be used to compare nested models by way of a likelihood ratio test.

**Output 54.2.4** Fit Statistics

| Fit Statistics | |
|---|---|
| **-2 Log Likelihood** | 27.9569 |
| **AIC (smaller is better)** | 35.9569 |
| **AICC (smaller is better)** | 38.8140 |
| **BIC (smaller is better)** | 39.7346 |
| **Pearson Chi-Square** | 13.4350 |
| **Pearson Chi-Square/DF** | 0.8957 |

The "Parameter Estimates" table in Output 54.2.5 displays the estimates and standard errors of the model effects.

**Output 54.2.5** Parameter Estimates

**Parameter Estimates**

| Parameter | DF | Estimate | Standard Error | Chi-Square | Pr > ChiSq |
|---|---|---|---|---|---|
| Intercept | 1 | -5.990191 | 1.666622 | 12.9183 | 0.0003 |
| Heat | 1 | 0.096339 | 0.047067 | 4.1896 | 0.0407 |
| Soak | 1 | 0.299574 | 0.755068 | 0.1574 | 0.6916 |
| Heat*Soak | 1 | -0.008840 | 0.025319 | 0.1219 | 0.7270 |

You can construct the prediction equation of the model from the "Parameter Estimates" table. For example, an observation with Heat equal to 14 and Soak equal to 1.7 has linear predictor

$$\widehat{\eta} = -5.9902 + 0.09634 \times 14 + 0.2996 \times 1.7 - 0.00884 \times 14 \times 7 = -4.34256$$

The probability that an ingot with these characteristics is not ready for rolling is

$$\widehat{\pi} = \frac{1}{1 + \exp\{-(-4.34256)\}} = 0.01284$$

The OUTPUT statement computes these linear predictors and probabilities and stores them in the Out data set. This data set also contains the ID variable, which is used by the following statements to attach the covariates to these statistics. Output 54.2.6 shows the probability that an ingot with Heat equal to 14 and Soak equal to 1.7 is not ready for rolling.

```
data Out;
   merge Out Ingots;
   by Obsnum;
proc print data=Out;
   where Heat=14 & Soak=1.7;
run;
```

**Output 54.2.6** Predicted Probability for Heat=14 and Soak=1.7

| Obs | Obsnum | Pred | Xbeta | Heat | Soak | r | n |
|---|---|---|---|---|---|---|---|
| 6 | 6 | 0.012836 | -4.34256 | 14 | 1.7 | 0 | 43 |

Binomial data are a form of grouped binary data where "successes" in the underlying Bernoulli trials are totaled. You can thus expand data for which you use the events/trials syntax and fit them with techniques for binary data.

The following DATA step expands the Ingots data set (which has 12 events in 387 trials) into a binary data set that has 387 observations.

```
data Ingots_binary;
   set Ingots;
   do i=1 to n;
     if i <= r then Y=1; else Y = 0;
     output;
   end;
run;
```

The following HPGENSELECT statements fit the model by using Heat effect, Soak effect, and their interaction to the binary data set. The **event='1'** response-variable option in the MODEL statement ensures that the HPGENSELECT procedure models the probability that the variable Y takes on the value '1'.

```
proc hpgenselect data=Ingots_binary;
   model Y(event='1') = Heat Soak Heat*Soak / dist=Binary;
run;
```

Output 54.2.7 displays the "Performance Information," "Model Information," "Number of Observations," and the "Response Profile" tables. The data are now modeled as binary (Bernoulli distributed) by using a logit link function. The "Response Profile" table shows that the binary response breaks down into 375 observations where Y equals 0 and 12 observations where Y equals 1.

**Output 54.2.7** Model Information in Binary Model

**The HPGENSELECT Procedure**

| Performance Information | |
|---|---|
| **Execution Mode** | Single-Machine |
| **Number of Threads** | 4 |

| Data Access Information | | | |
|---|---|---|---|
| **Data** | **Engine** | **Role** | **Path** |
| **WORK.INGOTS_BINARY** | V9 | Input | On Client |

| Model Information | |
|---|---|
| **Data Source** | WORK.INGOTS_BINARY |
| **Response Variable** | Y |
| **Distribution** | Binary |
| **Link Function** | Logit |
| **Optimization Technique** | Newton-Raphson with Ridging |

| | |
|---|---|
| **Number of Observations Read** | 387 |
| **Number of Observations Used** | 387 |

| Response Profile | | |
|---|---|---|
| **Ordered Value** | **Y** | **Total Frequency** |
| **1** | 0 | 375 |
| **2** | 1 | 12 |

**You are modeling the probability that Y='1'.**

Output 54.2.8 displays the parameter estimates. These results match those in Output 54.2.5.

**Output 54.2.8** Parameter Estimates

**Parameter Estimates**

| Parameter | DF | Estimate | Standard Error | Chi-Square | Pr > ChiSq |
|---|---|---|---|---|---|
| **Intercept** | 1 | -5.990191 | 1.666622 | 12.9183 | 0.0003 |
| **Heat** | 1 | 0.096339 | 0.047067 | 4.1896 | 0.0407 |
| **Soak** | 1 | 0.299574 | 0.755068 | 0.1574 | 0.6916 |
| **Heat*Soak** | 1 | -0.008840 | 0.025319 | 0.1219 | 0.7270 |

## Example 54.3: Tweedie Model

The following HPGENSELECT statements examine the data set getStarted used in the section "Getting Started: HPGENSELECT Procedure" on page 4299, but they request that a Tweedie model be fit by using the continuous variable Total as the response instead of the count variable Y. The following statements fit a log-linked Tweedie model to these data by using classification effects for variables C1–C5. In an insurance underwriting context, Y represents the total number of claims in each category that is defined by C1–C5, and Total represents the total cost of the claims (that is, the sum of costs for individual claims). The CODE statement requests that a text file named "Scoring Parameters.txt" be created. This file contains a SAS program that contains information from the model that allows scoring of a new data set based on the parameter estimates from the current model.

```
proc hpgenselect data=getStarted;
   class C1-C5;
   model Total = C1-C5 / Distribution=Tweedie Link=Log;
   code File='ScoringParameters.txt';
run;
```

The "Optimizations Stage Details" table in Output 54.3.1 shows the stages used in computing the maximum likelihood estimates of the parameters of the Tweedie model. Stage 1 uses quasi-likelihood and all of the data to compute starting values for stage 2, which uses all of the data and the Tweedie log likelihood to compute the final estimates.

**Output 54.3.1** Optimization Stage Details

**The HPGENSELECT Procedure**

**Optimization Stage Details**

| Optimization Stage | Optimization Type | Sampling Percentage | Observations Used |
|---|---|---|---|
| **1** | Quasilikelihood | 100.00 | 100 |
| **2** | Full Likelihood | 100.00 | 100 |

The "Parameter Estimates" table in Output 54.3.2 shows the resulting regression model parameter estimates, the estimated Tweedie dispersion parameter, and the estimated Tweedie power.

**Output 54.3.2** Parameter Estimates

**Parameter Estimates**

| Parameter | DF | Estimate | Standard Error | Chi-Square | Pr > ChiSq |
|---|---|---|---|---|---|
| **Intercept** | 1 | 3.888904 | 0.435325 | 79.8044 | <.0001 |
| **C1 0** | 1 | -0.072400 | 0.240613 | 0.0905 | 0.7635 |
| **C1 1** | 1 | -1.358456 | 0.324363 | 17.5400 | <.0001 |
| **C1 2** | 1 | 0.154711 | 0.237394 | 0.4247 | 0.5146 |
| **C1 3** | 0 | 0 | . | . | . |
| **C2 0** | 1 | 1.350591 | 0.289897 | 21.7050 | <.0001 |
| **C2 1** | 1 | 1.159242 | 0.275459 | 17.7106 | <.0001 |
| **C2 2** | 1 | 0.033921 | 0.303204 | 0.0125 | 0.9109 |
| **C2 3** | 0 | 0 | . | . | . |
| **C3 0** | 1 | -0.217763 | 0.272474 | 0.6387 | 0.4242 |
| **C3 1** | 1 | -0.289425 | 0.259751 | 1.2415 | 0.2652 |
| **C3 2** | 1 | -0.131961 | 0.276723 | 0.2274 | 0.6335 |
| **C3 3** | 0 | 0 | . | . | . |
| **C4 0** | 1 | -0.258069 | 0.288840 | 0.7983 | 0.3716 |
| **C4 1** | 1 | -0.057042 | 0.287566 | 0.0393 | 0.8428 |
| **C4 2** | 1 | 0.219697 | 0.272064 | 0.6521 | 0.4194 |
| **C4 3** | 0 | 0 | . | . | . |
| **C5 0** | 1 | -1.314657 | 0.257806 | 26.0038 | <.0001 |
| **C5 1** | 1 | -0.996980 | 0.236881 | 17.7138 | <.0001 |
| **C5 2** | 1 | -0.481185 | 0.235614 | 4.1708 | 0.0411 |
| **C5 3** | 0 | 0 | . | . | . |
| **Dispersion** | 1 | 5.296966 | 0.773401 | . | . |
| **Power** | 1 | 1.425625 | 0.048981 | . | . |

Now suppose you want to compute predicted values for some different data. If $\mathbf{x}$ is a vector of explanatory variables that might not be in the original data and $\hat{\boldsymbol{\beta}}$ is the vector of estimated regression parameters from the model, then $\mu = g^{-1}(\mathbf{x}'\hat{\boldsymbol{\beta}})$ is the predicted value of the mean, where $g$ is the log link function in this case. The following data contain new values of the regression variables C1–C5, from which you can compute predicted values based on information in the SAS program that is created by the CODE statement. This is called *scoring* the new data set.

```
data ScoringData;
   input C1-C5;
   datalines;
3 3 1 0 2
1 1 2 2 0
3 2 2 2 0
1 1 2 3 2
1 1 2 3 3
3 1 1 0 1
0 2 1 0 0
2 1 3 1 3
3 2 3 2 0
3 0 2 0 1
;
```

The following SAS DATA step creates the new data set Scores, which contains a variable P_Total that represents the predicted values of Total, along with the variables C1–C5. The resulting data are shown in Output 54.3.3.

```
data Scores;
   set ScoringData;
   %inc 'ScoringParameters.txt';
run;
proc print data=Scores;
run;
```

**Output 54.3.3** Predicted Values for Scoring Data

| Obs | C1 | C2 | C3 | C4 | C5 | P_Total |
|---|---|---|---|---|---|---|
| **1** | 3 | 3 | 1 | 0 | 2 | 17.465 |
| **2** | 1 | 1 | 2 | 2 | 0 | 11.737 |
| **3** | 3 | 2 | 2 | 2 | 0 | 14.819 |
| **4** | 1 | 1 | 2 | 3 | 2 | 21.683 |
| **5** | 1 | 1 | 2 | 3 | 3 | 35.083 |
| **6** | 3 | 1 | 1 | 0 | 1 | 33.237 |
| **7** | 0 | 2 | 1 | 0 | 0 | 7.303 |
| **8** | 2 | 1 | 3 | 1 | 3 | 171.711 |
| **9** | 3 | 2 | 3 | 2 | 0 | 16.909 |
| **10** | 3 | 0 | 2 | 0 | 1 | 47.110 |

## Example 54.4: Model Selection by the LASSO Method

This example shows how you can use PROC HPGENSELECT to perform model selection among Poisson regression models by using the LASSO method in single-machine and distributed modes. For more information about the execution modes of SAS High-Performance Statistics procedures, see the section "Processing Modes" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*). The focus of this example is to show how you use the LASSO method and how you can switch the modes of execution of PROC HPGENSELECT. The following DATA step generates the data for this example. There are 1,000,000 observations in the data set, and the response yPoisson is a Poisson variable with a mean that depends on 20 of the 100 regressors.

```
%let nObs      = 1000000;
%let nContIn   = 20;
%let nContOut  = 80;
%let Seed      = 12345;

data ex4Data;
array xIn{&nContIn};
array xOut{&nContOut};

drop i j sign xBeta expXbeta;

seed = &Seed;
do i=1 to &nObs;
   sign  = -1;
       xBeta = 0;
```

```
        do j=1 to dim(xIn);
            call ranuni(seed,xIn[j]);
            xBeta  = xBeta + j*sign*xIn{j};
            sign   = -sign;
        end;

        do j=1 to dim(xOut);
           call ranuni(seed,xOut[j]);
        end;

        call ranuni(seed,xSubtle);
        call ranuni(seed,xTiny);

        xBeta = xBeta + 0.1*xSubtle + 0.05*xTiny;
        expXbeta = exp(xBeta/20);
        call ranpoi(seed,expXbeta,yPoisson);
        output;
end;
run;
```

The following statements use PROC HPGENSELECT to select a model by using the LASSO method and only the first 10,000 observations:

```
proc hpgenselect data=ex4Data(Obs=10000);
   model yPoisson = x: / dist=Poisson;
   selection method=Lasso(choose=SBC) details=all;
   performance details;
run;
```

Output 54.4.1 shows the "Performance Information" table. This shows that the HPGENSELECT procedure executed in single-machine mode on four threads because the client machine has four CPUs. You can select a certain number of threads on any machine involved in the computations by using the NTHREADS= option in the PERFORMANCE statement.

**Output 54.4.1** Performance Information

**The HPGENSELECT Procedure**

| Performance Information | |
|---|---|
| **Execution Mode** | Single-Machine |
| **Number of Threads** | 4 |

Output 54.4.2 shows the models fit by maximizing the penalized log likelihoods for a sequence of regularization parameters. For each step in the sequence, Output 54.4.2 shows you the effects that are added or removed and the fit statistics AIC, AICC, and BIC (SBC) for each model. Unlike other methods, such as forward selection, LASSO selection includes zero or more effects in each step.

**Output 54.4.2** Selection Details

**Selection Details**

| Step | Description | Effects In Model | Lambda | AIC | AICC | BIC |
|---|---|---|---|---|---|---|
| **0** | Initial Model | 1 | 1 | 39592.399 | 39592.399 | 39599.609 |
| **1** | xIn17 entered | 5 | 0.8 | 38533.060 | 38533.066 | 38569.111 |
| | xIn18 entered | 5 | 0.8 | 38533.060 | 38533.066 | 38569.111 |
| | xIn19 entered | 5 | 0.8 | 38533.060 | 38533.066 | 38569.111 |
| | xIn20 entered | 5 | 0.8 | 38533.060 | 38533.066 | 38569.111 |
| **2** | xIn14 entered | 8 | 0.64 | 36731.498 | 36731.513 | 36789.181 |
| | xIn15 entered | 8 | 0.64 | 36731.498 | 36731.513 | 36789.181 |
| | xIn16 entered | 8 | 0.64 | 36731.498 | 36731.513 | 36789.181 |
| **3** | xIn11 entered | 11 | 0.512 | 34887.979 | 34888.005 | 34967.292 |
| | xIn12 entered | 11 | 0.512 | 34887.979 | 34888.005 | 34967.292 |
| | xIn13 entered | 11 | 0.512 | 34887.979 | 34888.005 | 34967.292 |
| **4** | xIn8 entered | 12 | 0.4096 | 33321.737 | 33321.769 | 33408.262 |
| **5** | xIn7 entered | 15 | 0.3277 | 32043.428 | 32043.476 | 32151.583 |
| | xIn9 entered | 15 | 0.3277 | 32043.428 | 32043.476 | 32151.583 |
| | xIn10 entered | 15 | 0.3277 | 32043.428 | 32043.476 | 32151.583 |
| **6** | xIn6 entered | 16 | 0.2621 | 31135.411 | 31135.465 | 31250.776 |
| **7** | xIn5 entered | 17 | 0.2097 | 30494.512 | 30494.573 | 30617.088 |
| **8** | xIn4 entered | 18 | 0.1678 | 30062.397 | 30062.465 | 30192.183 |
| **9** | xIn3 entered | 19 | 0.1342 | 29761.384 | 29761.460 | 29898.380 |
| **10** | xIn2 entered | 20 | 0.1074 | 29563.603 | 29563.687 | 29707.810 |
| **11** | | 20 | 0.0859 | 29432.716 | 29432.800 | 29576.922 |
| **12** | | 20 | 0.0687 | 29348.856 | 29348.940 | 29493.063 |
| **13** | xOut8 entered | 22 | 0.055 | 29297.961 | 29298.062 | 29456.588 |
| | xOut66 entered | 22 | 0.055 | 29297.961 | 29298.062 | 29456.588 |
| **14** | xIn1 entered | 26 | 0.044 | 29265.882 | 29266.022 | 29453.350* |
| | xOut34 entered | 26 | 0.044 | 29265.882 | 29266.022 | 29453.350* |
| | xOut37 entered | 26 | 0.044 | 29265.882 | 29266.022 | 29453.350* |
| | xOut65 entered | 26 | 0.044 | 29265.882 | 29266.022 | 29453.350* |
| **15** | xOut7 entered | 32 | 0.0352 | 29247.136 | 29247.348 | 29477.867 |
| | xOut22 entered | 32 | 0.0352 | 29247.136 | 29247.348 | 29477.867 |
| | xOut29 entered | 32 | 0.0352 | 29247.136 | 29247.348 | 29477.867 |
| | xOut51 entered | 32 | 0.0352 | 29247.136 | 29247.348 | 29477.867 |
| | xOut57 entered | 32 | 0.0352 | 29247.136 | 29247.348 | 29477.867 |
| | xOut60 entered | 32 | 0.0352 | 29247.136 | 29247.348 | 29477.867 |
| **16** | xOut1 entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| | xOut6 entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| | xOut14 entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| | xOut18 entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| | xOut27 entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| | xOut30 entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| | xOut33 entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| | xOut53 entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| | xOut56 entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| | xOut59 entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| | xOut62 entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| | xTiny entered | 44 | 0.0281 | 29244.461 | 29244.859 | 29561.716 |
| **17** | xOut11 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |

**Output 54.4.2** *continued*

**Selection Details**

| Step | Description | Effects In Model | Lambda | AIC | AICC | BIC |
|---|---|---|---|---|---|---|
| | xOut24 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |
| | xOut35 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |
| | xOut36 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |
| | xOut43 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |
| | xOut45 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |
| | xOut49 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |
| | xOut54 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |
| | xOut69 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |
| | xOut75 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |
| | xOut76 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |
| | xOut79 entered | 56 | 0.0225 | 29245.665 | 29246.307 | 29649.444 |
| **18** | xOut12 entered | 63 | 0.018 | 29241.974 | 29242.786 | 29696.225 |
| | xOut19 entered | 63 | 0.018 | 29241.974 | 29242.786 | 29696.225 |
| | xOut28 entered | 63 | 0.018 | 29241.974 | 29242.786 | 29696.225 |
| | xOut48 entered | 63 | 0.018 | 29241.974 | 29242.786 | 29696.225 |
| | xOut61 entered | 63 | 0.018 | 29241.974 | 29242.786 | 29696.225 |
| | xOut78 entered | 63 | 0.018 | 29241.974 | 29242.786 | 29696.225 |
| | xSubtle entered | 63 | 0.018 | 29241.974 | 29242.786 | 29696.225 |
| **19** | xOut4 entered | 69 | 0.0144 | 29241.231 | 29242.203 | 29738.744 |
| | xOut39 entered | 69 | 0.0144 | 29241.231 | 29242.203 | 29738.744 |
| | xOut40 entered | 69 | 0.0144 | 29241.231 | 29242.203 | 29738.744 |
| | xOut44 entered | 69 | 0.0144 | 29241.231 | 29242.203 | 29738.744 |
| | xOut67 entered | 69 | 0.0144 | 29241.231 | 29242.203 | 29738.744 |
| | xOut71 entered | 69 | 0.0144 | 29241.231 | 29242.203 | 29738.744 |
| **20** | xOut2 entered | 79 | 0.0115 | 29251.884 | 29253.158 | 29821.500 |
| | xOut3 entered | 79 | 0.0115 | 29251.884 | 29253.158 | 29821.500 |
| | xOut16 entered | 79 | 0.0115 | 29251.884 | 29253.158 | 29821.500 |
| | xOut21 entered | 79 | 0.0115 | 29251.884 | 29253.158 | 29821.500 |
| | xOut31 entered | 79 | 0.0115 | 29251.884 | 29253.158 | 29821.500 |
| | xOut41 entered | 79 | 0.0115 | 29251.884 | 29253.158 | 29821.500 |
| | xOut46 entered | 79 | 0.0115 | 29251.884 | 29253.158 | 29821.500 |
| | xOut50 entered | 79 | 0.0115 | 29251.884 | 29253.158 | 29821.500 |
| | xOut72 entered | 79 | 0.0115 | 29251.884 | 29253.158 | 29821.500 |
| | xOut77 entered | 79 | 0.0115 | 29251.884 | 29253.158 | 29821.500 |

*** Optimal Value of Criterion**

The model in step 14 had the smallest Schwarz Bayesian criterion (BIC in Output 54.4.2), and it was chosen as the final model because the CHOOSE=SBC option was specified in the SELECTION statement. Output 54.4.3 shows the parameter estimates for the selected model. You can see that the LASSO selection in which the final model was chosen based on the SBC criterion retains all 20 of the true effects but also keeps several extraneous effects.

**Output 54.4.3** Parameter Estimates for the Selected Model

| Parameter Estimates | | |
|---|---|---|
| **Parameter** | **DF** | **Estimate** |
| **Intercept** | 1 | -0.008605 |
| **xIn1** | 1 | -0.004242 |
| **xIn2** | 1 | 0.076690 |
| **xIn3** | 1 | -0.141401 |
| **xIn4** | 1 | 0.160843 |
| **xIn5** | 1 | -0.219377 |
| **xIn6** | 1 | 0.268402 |
| **xIn7** | 1 | -0.336859 |
| **xIn8** | 1 | 0.384260 |
| **xIn9** | 1 | -0.407838 |
| **xIn10** | 1 | 0.397287 |
| **xIn11** | 1 | -0.516073 |
| **xIn12** | 1 | 0.559396 |
| **xIn13** | 1 | -0.547200 |
| **xIn14** | 1 | 0.631820 |
| **xIn15** | 1 | -0.692216 |
| **xIn16** | 1 | 0.783101 |
| **xIn17** | 1 | -0.790773 |
| **xIn18** | 1 | 0.875372 |
| **xIn19** | 1 | -0.837447 |
| **xIn20** | 1 | 0.954752 |
| **xOut8** | 1 | 0.013643 |
| **xOut34** | 1 | 0.002167 |
| **xOut37** | 1 | 0.001344 |
| **xOut65** | 1 | -0.007790 |
| **xOut66** | 1 | 0.015540 |

Output 54.4.4 shows timing information for the PROC HPGENSELECT run. This table is produced when you specify the DETAILS option in the PERFORMANCE statement. You can see that, in this case, the majority of time is spent in performing model selection.

**Output 54.4.4** Timing

| Procedure Task Timing | | |
|---|---|---|
| **Task** | **Seconds** | **Percent** |
| **Reading and Levelizing Data** | 0.08 | 0.07% |
| **Candidate model fit** | 0.06 | 0.05% |
| **Performing Model Selection** | 112.89 | 99.88% |

You can switch to running PROC HPGENSELECT in distributed mode by specifying valid values for the NODES=, INSTALL=, and HOST= options in the PERFORMANCE statement. An alternative to specifying the INSTALL= and HOST= options in the PERFORMANCE statement is to set appropriate values for the GRIDHOST and GRIDINSTALLLOC environment variables by using OPTIONS SET commands. For more information about setting these options or environment variables, see the section "Processing Modes" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

The following statements provide an example. All 1,000,000 observations are used in this example of running PROC HPGENSELECT in distributed mode. To run these statements successfully, you need to set the macro variables GRIDHOST and GRIDINSTALLLOC to resolve to appropriate values, or you can replace the references to macro variables with appropriate values.

```
proc hpgenselect data=ex4Data;
   model yPoisson = x: / dist=Poisson;
   selection method=Lasso(choose=SBC) details=all;
   performance details nodes = 10
               host="&GRIDHOST" install="&GRIDINSTALLLOC";
run;
```

The Execution Mode row in the "Performance Information" table shown in Output 54.4.5 indicates that the calculations were performed in a distributed environment that used 10 nodes, each of which used 32 threads.

**Output 54.4.5** Performance Information in Distributed Mode

| Performance Information | |
|---|---|
| **Host Node** | << your grid host >> |
| **Install Location** | << your grid install location >> |
| **Execution Mode** | Distributed |
| **Number of Compute Nodes** | 10 |
| **Number of Threads per Node** | 32 |

Another indication of distributed execution is the following message in the SAS log, which is issued by all high-performance statistical procedures:

```
NOTE: The HPGENSELECT procedure is executing in the distributed
      computing environment with 10 worker nodes.
```

Output 54.4.6 shows timing information for this distributed run of PROC HPGENSELECT. As in the case of the single-machine mode, the majority of time in distributed mode is spent in performing the model selection.

**Output 54.4.6** Timing

| Procedure Task Timing | | |
|---|---|---|
| **Task** | **Seconds** | **Percent** |
| **Distributing Data** | 15.54 | 3.81% |
| **Reading and Levelizing Data** | 0.38 | 0.09% |
| **Candidate model fit** | 0.13 | 0.03% |
| **Performing Model Selection** | 392.10 | 96.07% |

Output 54.4.7 shows the models that were fit by maximizing the penalized log likelihoods for a sequence of regularization parameters. In this case, the model in the last step had the smallest SBC statistic and was the selected model.

**Output 54.4.7** Selection Details

**Selection Details**

| Step | Description | Effects In Model | Lambda | AIC | AICC | BIC |
|---|---|---|---|---|---|---|
| **0** | Initial Model | 1 | 1 | 3931380.08 | 3931380.08 | 3931391.90 |
| **1** | xIn16 entered | 6 | 0.8 | 3811013.61 | 3811013.61 | 3811084.50 |
| | xIn17 entered | 6 | 0.8 | 3811013.61 | 3811013.61 | 3811084.50 |
| | xIn18 entered | 6 | 0.8 | 3811013.61 | 3811013.61 | 3811084.50 |
| | xIn19 entered | 6 | 0.8 | 3811013.61 | 3811013.61 | 3811084.50 |
| | xIn20 entered | 6 | 0.8 | 3811013.61 | 3811013.61 | 3811084.50 |
| **2** | xIn13 entered | 9 | 0.64 | 3611604.92 | 3611604.92 | 3611711.26 |
| | xIn14 entered | 9 | 0.64 | 3611604.92 | 3611604.92 | 3611711.26 |
| | xIn15 entered | 9 | 0.64 | 3611604.92 | 3611604.92 | 3611711.26 |
| **3** | xIn11 entered | 11 | 0.512 | 3424886.61 | 3424886.61 | 3425016.58 |
| | xIn12 entered | 11 | 0.512 | 3424886.61 | 3424886.61 | 3425016.58 |
| **4** | xIn9 entered | 13 | 0.4096 | 3273653.05 | 3273653.05 | 3273806.65 |
| | xIn10 entered | 13 | 0.4096 | 3273653.05 | 3273653.05 | 3273806.65 |
| **5** | xIn7 entered | 15 | 0.3277 | 3160535.72 | 3160535.72 | 3160712.95 |
| | xIn8 entered | 15 | 0.3277 | 3160535.72 | 3160535.72 | 3160712.95 |
| **6** | xIn6 entered | 16 | 0.2621 | 3080686.19 | 3080686.19 | 3080875.24 |
| **7** | xIn5 entered | 17 | 0.2097 | 3025122.51 | 3025122.51 | 3025323.38 |
| **8** | xIn4 entered | 18 | 0.1678 | 2987411.53 | 2987411.53 | 2987624.21 |
| **9** | xIn3 entered | 19 | 0.1342 | 2962303.01 | 2962303.01 | 2962527.50 |
| **10** | | 19 | 0.1074 | 2945736.43 | 2945736.43 | 2945960.92 |
| **11** | xIn2 entered | 20 | 0.0859 | 2934645.35 | 2934645.36 | 2934881.66 |
| **12** | | 20 | 0.0687 | 2927476.58 | 2927476.58 | 2927712.89 |
| **13** | | 20 | 0.055 | 2922886.50 | 2922886.50 | 2923122.81 |
| **14** | xIn1 entered | 21 | 0.044 | 2919875.95 | 2919875.95 | 2920124.08 |
| **15** | | 21 | 0.0352 | 2917896.15 | 2917896.15 | 2918144.28 |
| **16** | | 21 | 0.0281 | 2916627.57 | 2916627.57 | 2916875.69 |
| **17** | | 21 | 0.0225 | 2915816.54 | 2915816.54 | 2916064.66 |
| **18** | | 21 | 0.018 | 2915297.37 | 2915297.37 | 2915545.49 |
| **19** | | 21 | 0.0144 | 2914965.03 | 2914965.03 | 2915213.16 |
| **20** | | 21 | 0.0115 | 2914752.31 | 2914752.31 | 2915000.44* |

*** Optimal Value of Criterion**

Output 54.4.8 shows the parameter estimates for the selected model. Selecting the final model based on the SBC criterion retains all 20 of the true effects and none of the extraneous effects.

**Output 54.4.8** Parameter Estimates

| Parameter Estimates | | |
|---|---|---|
| **Parameter** | **DF** | **Estimate** |
| **Intercept** | 1 | 0.011850 |
| **xIn1** | 1 | -0.037295 |
| **xIn2** | 1 | 0.090866 |
| **xIn3** | 1 | -0.137335 |
| **xIn4** | 1 | 0.188475 |
| **xIn5** | 1 | -0.238960 |
| **xIn6** | 1 | 0.287461 |
| **xIn7** | 1 | -0.336226 |
| **xIn8** | 1 | 0.389728 |
| **xIn9** | 1 | -0.438611 |
| **xIn10** | 1 | 0.486188 |
| **xIn11** | 1 | -0.539309 |
| **xIn12** | 1 | 0.587680 |
| **xIn13** | 1 | -0.637463 |
| **xIn14** | 1 | 0.684980 |
| **xIn15** | 1 | -0.735036 |
| **xIn16** | 1 | 0.790630 |
| **xIn17** | 1 | -0.836793 |
| **xIn18** | 1 | 0.885972 |
| **xIn19** | 1 | -0.934294 |
| **xIn20** | 1 | 0.985428 |

# References

Akaike, H. (1974). "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control* AC-19:716–723.

Burnham, K. P., and Anderson, D. R. (1998). *Model Selection and Inference: A Practical Information-Theoretic Approach*. New York: Springer-Verlag.

Cox, D. R., and Snell, E. J. (1989). *The Analysis of Binary Data*. 2nd ed. London: Chapman & Hall.

Dennis, J. E., Gay, D. M., and Welsch, R. E. (1981). "An Adaptive Nonlinear Least-Squares Algorithm." *ACM Transactions on Mathematical Software* 7:348–368.

Dennis, J. E., and Mei, H. H. W. (1979). "Two New Unconstrained Optimization Algorithms Which Use Function and Gradient Values." *Journal of Optimization Theory and Applications* 28:453–482.

Dunn, P. K., and Smyth, G. K. (2005). "Series Evaluation of Tweedie Exponential Dispersion Model Densities." *Statistics and Computing* 15:267–280.

Dunn, P. K., and Smyth, G. K. (2008). "Series Evaluation of Tweedie Exponential Dispersion Model Densities by Fourier Inversion." *Statistics and Computing* 18:73–86.

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

Eskow, E., and Schnabel, R. B. (1991). "Algorithm 695: Software for a New Modified Cholesky Factorization." *ACM Transactions on Mathematical Software* 17:306–312.

Fletcher, R. (1987). *Practical Methods of Optimization*. 2nd ed. Chichester, UK: John Wiley & Sons.

Frees, E. W. (2010). *Regression Modeling with Actuarial and Financial Applications*. Cambridge: Cambridge University Press.

Gay, D. M. (1983). "Subroutines for Unconstrained Minimization." *ACM Transactions on Mathematical Software* 9:503–524.

Hastie, T. J., Tibshirani, R. J., and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer-Verlag.

Hurvich, C. M., and Tsai, C.-L. (1989). "Regression and Time Series Model Selection in Small Samples." *Biometrika* 76:297–307.

McCullagh, P., and Nelder, J. A. (1989). *Generalized Linear Models*. 2nd ed. London: Chapman & Hall.

Moré, J. J., and Sorensen, D. C. (1983). "Computing a Trust-Region Step." *SIAM Journal on Scientific and Statistical Computing* 4:553–572.

Nesterov, Y. (2013). "Gradient Methods for Minimizing Composite Objective Function." *Mathematical Programming* 140:125–161.

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

Yuan, M., and Lin, L. (2006). "Model Selection and Estimation in Regression with Grouped Variables." *Journal of the Royal Statistical Society, Series B* 68:49–67.

# Subject Index

# Syntax Index