SAS/STAT® 14.3 User's Guide
The HPLOGISTIC 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 56
The HPLOGISTIC Procedure

Contents

Overview: HPLOGISTIC Procedure

The HPLOGISTIC procedure is a high-performance statistical procedure that fits logistic regression models for binary, binomial, and multinomial data on the SAS appliance.

The HPLOGISTIC procedure fits logistic regression models in the broader sense; the procedure permits several link functions and can handle ordinal and nominal data with more than two response categories (multinomial data).

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

NOTE: Distributed mode requires SAS High-Performance Statistics.

PROC HPLOGISTIC Features

The HPLOGISTIC procedure estimates the parameters of a logistic regression model by using maximum likelihood techniques. It also does the following:

- provides model-building syntax with the CLASS and effect-based MODEL statements, which are familiar from SAS/STAT analytic procedures (in particular, the GLM, LOGISTIC, GLIMMIX, and MIXED procedures)
- provides response-variable options as in the LOGISTIC procedure
- performs maximum likelihood estimation
- provides multiple link functions
- provides cumulative link models 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 contains predicted probabilities and other observationwise statistics

Because the HPLOGISTIC procedure is a high-performance statistical 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
- performs parallel reads of input data and parallel writes of output data when the data source is the appliance database

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

PROC HPLOGISTIC Contrasted with Other SAS Procedures

For general contrasts, see the section "Common Features of SAS High-Performance Statistical Procedures" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*). The following remarks contrast the HPLOGISTIC procedure with the LOGISTIC procedure in SAS/STAT software.

The CLASS statement in the HPLOGISTIC procedure permits two parameterizations: the GLM parameterization and a reference parameterization. In contrast to the LOGISTIC, GENMOD, and other procedures that permit multiple parameterizations, the HPLOGISTIC 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 the GLM or reference parameterization.

The default parameterization of CLASS variables in the HPLOGISTIC procedure is the GLM parameterization. The LOGISTIC procedure uses the EFFECT parameterization for the CLASS variables by default. In either procedure, you can change the parameterization with the PARAM= option in the CLASS statement.

The default optimization technique used by the LOGISTIC procedure is Fisher scoring; the HPLOGISTIC procedure uses by default 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, with the TECHNIQUE= option in the PROC HPLOGISTIC statement.

The LOGISTIC procedure offers a wide variety of postfitting analyses, such as contrasts, estimates, tests of model effects, least squares means, and odds ratios. This release of the HPLOGISTIC procedure is limited in postfitting functionality, since with large data sets the focus is primarily on model fitting and scoring.

The HPLOGISTIC procedure is specifically designed to operate in the high-performance distributed environment. By default, PROC HPLOGISTIC performs computations in multiple threads. The LOGISTIC procedure executes in a single thread.

Getting Started: HPLOGISTIC Procedure

Binary Logistic Regression

The following DATA step contains 100 observations on a dichotomous response variable (y), a character variable (C), and 10 continuous variables (x1–x10):

```
data getStarted;
   input C$ y x1-x10;
   datalines;
   D  0  10.2  6  1.6  38  15  2.4  20  0.8  8.5  3.9
   F  1  12.2  6  2.6  42  61  1.5  10  0.6  8.5  0.7
   D  1   7.7  1  2.1  38  61    1  90  0.6  7.5  5.2
   J  1  10.9  7  3.5  46  42  0.3   0  0.2    6  3.6
   E  0  17.3  6  3.8  26  47  0.9  10  0.4  1.5  4.7
   A  0  18.7  4  1.8   2  34  1.7  80    1  9.5  2.2
   B  0   7.2  1  0.3  48  61  1.1  10  0.8  3.5    4
   D  0   0.1  3  2.4   0  65  1.6  70  0.8  3.5  0.7
   H  1   2.4  4  0.7  38  22  0.2  20    0    3  4.2
   J  0  15.6  7  1.4   0  98  0.3   0    1    5  5.2
   J  0  11.1  3  2.4  42  55  2.2  60  0.6  4.5  0.7
   F  0     4  6  0.9   4  36  2.1  30  0.8    9  4.6
   A  0   6.2  2  1.8  14  79  1.1  70  0.2    0  5.1
   H  0   3.7  3  0.8  12  66  1.3  40  0.4  0.5  3.3
   A  1   9.2  3  2.3  48  51  2.3  50    0    6  5.4
   G  0    14  3    2  18  12  2.2   0    0    3  3.4
   E  1  19.5  6  3.7  26  81  0.1  30  0.6    5  4.8
   C  0    11  3  2.8  38   9  1.7  50  0.8  6.5  0.9
   I  0  15.3  7  2.2  20  98  2.7 100  0.4    7  0.8
   H  1   7.4  4  0.5  28  65  1.3  60  0.2  9.5  5.4
   F  0  11.4  2  1.4  42  12  2.4  10  0.4    1  4.5
   C  1  19.4  1  0.4  42   4  2.4  10    0  6.5  0.1
   G  0   5.9  4  2.6  12  57  0.8  50  0.4    2  5.8
   G  1  15.8  6  3.7  34   8  1.3  90  0.6  2.5  5.7
   I  0    10  3  1.9  16  80    3  90  0.4  9.5  1.9
   E  0  15.7  1  2.7  32  25  1.7  20  0.2  8.5    6
   G  0    11  5  2.9  48  53  0.1  50    1  3.5  1.2
   J  1  16.8  0  0.9  14  86  1.4  40  0.8    9    5
   D  1    11  4  3.2  48  63  2.8  90  0.6    0  2.2
   J  1   4.8  7  3.6  24   1  2.2  20    1  8.5  0.5
   J  1  10.4  5    2  42  56    1  20    0  3.5  4.2
   G  0  12.7  7  3.6   8  56  2.1  70    1  4.5  1.5
   G  0   6.8  1  3.2  30  27  0.6   0  0.8    2  5.6
   E  0   8.8  0  3.2   2  67  0.7  10  0.4    1    5
   I  1   0.2  0  2.9  10  41  2.3  60  0.2    9  0.3
   J  1   4.6  7  3.9  50  61  2.1  50  0.4    3  4.9
   J  1   2.3  2  3.2  36  98  0.1  40  0.6  4.5  4.3
   I  0  10.8  3  2.7  28  58  0.8  80  0.8    3    6
   B  0   9.3  2  3.3  44  44  0.3  50  0.8  5.5  0.4
   F  0   9.2  6  0.6   4  64  0.1   0  0.6  4.5  3.9
```

```
D  0   7.4  0  2.9  14   0  0.2  30  0.8  7.5  4.5
G  0  18.3  3  3.1   8  60  0.3  60  0.2    7  1.9
F  0   5.3  4  0.2  48  63  2.3  80  0.2    8  5.2
C  0   2.6  5  2.2  24   4  1.3  20    0    2  1.4
F  0  13.8  4  3.6   4   7  1.1  10  0.4  3.5  1.9
B  1  12.4  6  1.7  30  44  1.1  60  0.2    6  1.5
I  0   1.3  1  1.3   8  53  1.1  70  0.6    7  0.8
F  0  18.2  7  1.7  26  92  2.2  30    1  8.5  4.8
J  0   5.2  2  2.2  18  12  1.4  90  0.8    4  4.9
G  1   9.4  2  0.8  22  86  0.4  30  0.4    1  5.9
J  1  10.4  2  1.7  26  31  2.4  10  0.2    7  1.6
J  0    13  1  1.8  14  11  2.3  50  0.6  5.5  2.6
A  0  17.9  4  3.1  46  58  2.6  90  0.6  1.5  3.2
D  1  19.4  6    3  20  50  2.8 100  0.2    9  1.2
I  0  19.6  3  3.6  22  19  1.2   0  0.6    5  4.1
I  1     6  2  1.5  30  30  2.2  20  0.4  8.5  5.3
G  0  13.8  1  2.7   0  52  2.4  20  0.8    6    2
B  0  14.3  4  2.9  30  11  0.6  90  0.6  0.5  4.9
E  0  15.6  0  0.4  38  79  0.4  80  0.4    1  3.3
D  0    14  2    1  22  61    3  90  0.6    2  0.1
C  1   9.4  5  0.4  12  53  1.7  40    0    3  1.1
H  0  13.2  1  1.6  40  15  0.7  40  0.2    9  5.5
A  0  13.5  5  2.4  18  89  1.6  20  0.4  9.5  4.7
E  0   2.6  4  2.3  38   6  0.8  20  0.4    5  5.3
E  0  12.4  3  1.3  26   8  2.8  10  0.8    6  5.8
D  0   7.6  2  0.9  44  89  1.3  50  0.8    6  0.4
I  0  12.7  1  2.3  42   6  2.4  10  0.4    1    3
C  1  10.7  4  3.2  28  23  2.2  90  0.8  5.5  2.8
H  0  10.1  2  2.3  10  62  0.9  50  0.4  2.5  3.7
C  1  16.6  1  0.5  12  88  0.1  20  0.6  5.5  1.8
I  1   0.2  3  2.2   8  71  1.7  80  0.4  0.5  5.5
C  0  10.8  4  3.5  30  70  2.3  60  0.4  4.5  5.9
F  0   7.1  4    3  14  63  2.4  70    0    7  3.1
D  0  16.5  1  3.3  30  80  1.6  40    0  3.5  2.7
H  0  17.1  7  2.1  30  45  1.5  60  0.6  0.5  2.8
D  0   4.3  1  1.5  24  44    0  70    0    5  0.5
H  0    15  2  0.2  14  87  1.8  50    0  4.5  4.7
G  0  19.7  3  1.9  36  99  1.5  10  0.6    3  1.7
H  1   2.8  6  0.6  34  21    2  60    1    9  4.7
G  0  16.6  3  3.3  46   1  1.4  70  0.6  1.5  5.3
E  0  11.7  5  2.7  48   4  0.9  60  0.8  4.5  1.6
F  0  15.6  3  0.2   4  79  0.5   0  0.8  1.5  2.9
C  1   5.3  6  1.4   8  64    2  80  0.4    9  4.2
B  1   8.1  7  1.7  40  36  1.4  60  0.6    6  3.9
I  0  14.8  2  3.2   8  37  0.4  10    0  4.5    3
D  0   7.4  4    3  12   3  0.6  60  0.6    7  0.7
D  0   4.8  3  2.3  44  41  1.9  60  0.2    3  3.1
A  0   4.5  0  0.2   4  48  1.7  80  0.8    9  4.2
D  0   6.9  6  3.3  14  92  0.5  40  0.4  7.5    5
B  0   4.7  4  0.9  14  99  2.4  80    1  0.5  0.7
I  1   7.5  4  2.1  20  79  0.4  40  0.4  2.5  0.7
C  0   6.1  0  1.4  38  18  2.3  60  0.8  4.5  0.7
C  0  18.3  1    1  26  98  2.7  20    1  8.5  0.5
F  0  16.4  7  1.2  32  94  2.9  40  0.4  5.5  2.1
```

```
   I  0   9.4  2  2.3  32  42  0.2  70  0.4  8.5  0.3
   F  1  17.9  4  1.3  32  42    2  40  0.2    1  5.4
   H  0  14.9  3  1.6  36  74  2.6  60  0.2    1  2.3
   C  0  12.7  0  2.6   0  88  1.1  80  0.8  0.5  2.1
   F  0   5.4  4  1.5   2   1  1.8  70  0.4  5.5  3.6
   J  1  12.1  4  1.8  20  59  1.3  60  0.4    3  3.8
;
```

The following statements fit a logistic model to these data by using a classification effect for variable C and 10 regressor effects for x1–x10:

```
proc hplogistic data=getStarted;
   class C;
   model y = C x1-x10;
run;
```

The default output from this analysis is presented in Figure 56.1 through Figure 56.11.

The "Performance Information" table in Figure 56.1 shows that the procedure executes in single-machine mode—that is, the model is fit on the machine where the SAS session executes. This run of the HPLOGISTIC procedure was performed on a multicore machine with the same number of CPUs as there are threads; that is, one computational thread was spawned per CPU.

Figure 56.1 Performance Information

The HPLOGISTIC Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	4

Figure 56.2 displays the "Model Information" table. The HPLOGISTIC procedure uses a Newton-Raphson algorithm to model a binary distribution for the variable y with a logit link function. The CLASS variable C is parameterized using the GLM parameterization, which is the default.

Figure 56.2 Model Information

Model Information	
Data Source	WORK.GETSTARTED
Response Variable	y
Class Parameterization	GLM
Distribution	Binary
Link Function	Logit
Optimization Technique	Newton-Raphson with Ridging

The CLASS variable C has 10 unique formatted levels, and these are displayed in the "Class Level Information" table in Figure 56.3.

Figure 56.3 Class Level Information

Class Level Information		
Class	**Levels**	**Values**
C	10	A B C D E F G H I J

Figure 56.4 displays the "Number of Observations" table. All 100 observations in the data set are used in the analysis.

Figure 56.4 Number of Observations

Number of Observations Read	100
Number of Observations Used	100

The "Response Profile" table in Figure 56.5 is produced by default for binary and multinomial response variables. It shows the breakdown of the response variable levels by frequency. By default for binary data, the HPLOGISTIC procedure models the probability of the event with the lower-ordered value in the "Response Profile" table—this is indicated by the note that follows the table. In this example, the values represented by y = '0' are modeled as the "successes" in the Bernoulli experiments.

Figure 56.5 Response Profile

Response Profile

Ordered Value	y	Total Frequency
1	0	69
2	1	31

You are modeling the probability that y='0'.

You can use the response-variable options in the MODEL statement to affect which value of the response variable is modeled.

Figure 56.6 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 $\mathbf{X}$, which correspond to one column for the intercept, 10 columns for the effect associated with the classification variable C, and one column each for the continuous variables x1–x10. However, the rank of the crossproducts matrix is only 20. Because the classification variable C uses GLM parameterization and because the model contains an intercept, there is one singularity in the crossproducts matrix of the model. Consequently, only 20 parameters enter the optimization.

Figure 56.6 Dimensions in Binomial Logistic Regression

Dimensions	
Columns in X	21
Number of Effects	12
Max Effect Columns	10
Rank of Cross-product Matrix	20
Parameters in Optimization	20

The "Iteration History" table is shown in Figure 56.7. The Newton-Raphson algorithm with ridging converged after four iterations, not counting the initial setup iteration.

Figure 56.7 Iteration History

Iteration History				
Iteration	Evaluations	Objective Function	Change	Max Gradient
0	4	0.4493546916	.	0.410972
1	2	0.4436453992	0.00570929	0.081339
2	2	0.4435038109	0.00014159	0.003302
3	2	0.4435035933	0.00000022	5.623E-6
4	2	0.4435035933	0.00000000	1.59E-11

Figure 56.8 displays the final convergence status of the Newton-Raphson algorithm. The GCONV= relative convergence criterion is satisfied.

Figure 56.8 Convergence Status

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

The "Fit Statistics" table is shown in Figure 56.9. The –2 log likelihood at the converged estimates is 88.7007. 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. Because of the large number of parameters relative to the number of observations, the discrepancy between the –2 log likelihood and, say, AIC, is substantial in this case.

Figure 56.9 Fit Statistics

Fit Statistics	
-2 Log Likelihood	88.7007
AIC (smaller is better)	128.70
AICC (smaller is better)	139.33
BIC (smaller is better)	180.80

Figure 56.10 shows the global test for the null hypothesis that all model effects jointly do not affect the probability of success of the binary response. The test is significant (p-value = 0.0135). One or more of the model effects thus significantly affects the probability of observing an event.

Figure 56.10 Null Test

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	35.1194	19	0.0135

However, a look at the "Parameter Estimates" table in Figure 56.11 shows that many parameters have fairly large p-values, indicating that one or more of the model effects might not be necessary.

Figure 56.11 Parameter Estimates

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	1.2101	1.7507	Infty	0.69	0.4894
C A	3.4341	1.6131	Infty	2.13	0.0333
C B	2.1638	1.4271	Infty	1.52	0.1295
C C	0.6552	1.0810	Infty	0.61	0.5445
C D	2.4945	1.1094	Infty	2.25	0.0245
C E	3.2449	1.4321	Infty	2.27	0.0235
C F	3.6054	1.3070	Infty	2.76	0.0058
C G	2.0841	1.1898	Infty	1.75	0.0798
C H	2.9368	1.2939	Infty	2.27	0.0232
C I	1.3785	1.0319	Infty	1.34	0.1816
C J	0	.	.	.	.
x1	0.03218	0.05710	Infty	0.56	0.5730
x2	-0.3677	0.1538	Infty	-2.39	0.0168
x3	0.3146	0.3574	Infty	0.88	0.3787
x4	-0.05196	0.02443	Infty	-2.13	0.0334
x5	-0.00683	0.01056	Infty	-0.65	0.5177
x6	0.2539	0.3785	Infty	0.67	0.5024
x7	-0.00723	0.01073	Infty	-0.67	0.5004
x8	2.5370	0.9942	Infty	2.55	0.0107
x9	-0.1675	0.1068	Infty	-1.57	0.1168
x10	-0.2222	0.1577	Infty	-1.41	0.1590

Syntax: HPLOGISTIC Procedure

The following statements are available in the HPLOGISTIC procedure:

PROC HPLOGISTIC < *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* **;**
SELECTION *selection-options* **;**
WEIGHT *variable* **;**

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

PROC HPLOGISTIC Statement

PROC HPLOGISTIC < *options* > **;**

The PROC HPLOGISTIC statement invokes the procedure. Table 56.1 summarizes the available options in the PROC HPLOGISTIC statement by function. The options are then described fully in alphabetical order.

Table 56.1 PROC HPLOGISTIC 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
Options Related to Output	
ITDETAILS	Adds detail information to "Iteration History" table
ITSELECT	Displays the "Iteration History" table with model selection
NOPRINT	Suppresses ODS output
NOCLPRINT	Limits or suppresses the display of class levels
NOITPRINT	Suppresses generation of the iteration history table
NOSTDERR	Suppresses computation of covariance matrix and standard errors
Options Related to Optimization	
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 the starting values
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 name to the "Parameter Estimates" table
TECHNIQUE=	Selects the optimization technique
Tolerances	
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 Formats	
FMTLIBXML=	Specifies the file reference for a format stream

You can specify the following options in the PROC HPLOGISTIC 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 overflows.

ABSFCONV=*r*

ABSFTOL=*r*

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 with the lowest function value and $\boldsymbol{\psi}^{(k-1)}$ is defined as the vertex with the highest function value in the simplex. The default value is $r = 0$.

ABSGCONV=*r*

ABSGTOL=*r*

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 jth parameter. This criterion is not used by the NMSIMP technique. The default value is r=1E–5.

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 the global specification with the ALPHA= option in the MODEL statement.

DATA=*SAS-data-set*

names the input SAS data set for PROC HPLOGISTIC 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*

FTOL=*r*

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 with the lowest function value, and $\boldsymbol{\psi}^{(k-1)}$ is defined as the vertex with the highest function value in the simplex.

The default value is r=$2 \times \epsilon$ where ϵ 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 more information about how to generate a XML stream for your formats, see the section "Working with Formats" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

GCONV=*r*

GTOL=*r*

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 default value is r=1E–8.

INEST=*SAS-data-set*

names the TYPE=EST 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 HPLOGISTIC uses only the first observation for which _TYPE_=PARMS, and it ignores BY variables. For more information about TYPE=EST data sets, see Chapter A, "Special SAS Data Sets."

If you specify TECH=NONE or MAXITER=0, then the values in the INEST= data set are used as the parameter estimates, but the null model is still computed at the optimum value for the intercepts. If you specify TECH=NONE or MAXITER=0 and you specify a null model in the MODEL statement, then the null model is computed at the starting values for the intercept 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.

ITSELECT

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

MAXFUNC=*n*

MAXFU=*n*

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

- TRUREG, NRRIDG, NEWRAP: 125
- QUANEW, DBLDOG: 500
- CONGRA: 1,000
- NMSIMP: 3,000

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

MAXITER=*n*

MAXIT=*n*

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

- TRUREG, NRRIDG, NEWRAP: 50
- QUANEW, DBLDOG: 200
- CONGRA: 400
- NMSIMP: 1,000

These default values also apply when *n* is specified as a missing value. You can choose the optimization technique with 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 the MAXTIME= option is checked only once at the end of each iteration. Therefore, the actual running time can be longer than that specified by the MAXTIME= option.

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 can 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.

NOITPRINT

suppresses the generation of the "Iteration History" table.

NOPRINT

suppresses the generation of ODS output.

NORMALIZE=YES | NO

specifies whether the objective function should be normalized during the optimization by the reciprocal of the used frequency count. The default is to normalize the objective function. This option affects the values reported in the "Iteration History" table. The results reported in the "Fit Statistics" are always displayed for the nonnormalized log-likelihood function.

NOSTDERR
suppresses the computation of the covariance matrix and the standard errors of the logistic 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(type=EST) label=_TYPE_;
   label Estimate=PARMS;
   var Estimate;
   id ParmName;
run;
```

SINGCHOL=*number*
tunes the singularity criterion in Cholesky decompositions. The default is 1E7 times the machine epsilon; this product is approximately 1E–9 on most computers.

SINGSWEEP=*number*
tunes the singularity criterion for sweep operations. The default is 1E7 times the machine epsilon; this product is approximately 1E–9 on most computers.

SINGULAR=*number*
tunes the general singularity criterion applied by the HPLOGISTIC procedure in sweeps and inversions. The default is 1E7 times the machine epsilon; this product is approximately 1E–9 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.

For more information, see the section "Choosing an Optimization Algorithm" on page 4454.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC HPLOGISTIC 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 HPLOGISTIC 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 HPLOGISTIC 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 for high-performance statistical procedures is documented in the section "CLASS Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

The HPLOGISTIC procedure does not support the SPLIT option in the CLASS statement. The HPLOGISTIC 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 56.2 summarizes the *options* available in the CODE statement.

Table 56.2 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. High-performance statistical procedures that support the FREQ statement treat 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 output data sets created by high-performance statistical procedures, provided that the output data set produces one (or more) records per input observation.

For documentation about the common ID statement in high-performance statistical procedures, 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* < **(***response-options***)** > **=** < *effects* > < / *model-options* > **;**

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

You can specify a single *response* variable that contains your 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. Note that the values of both *events* and (*trials* – *events*) must be nonnegative and the value of *trials* must be positive.

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 HPLOGISTIC procedure models probabilities for binary data. The *model-options* control other aspects of model formation and inference. Table 56.3 summarizes these options.

Table 56.3 MODEL Statement Options

Option	Description
Response Variable Options	
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
ASSOCIATION	Requests association statistics
CL	Requests confidence limits
CTABLE	Requests classification statistics
CUTPOINT=	Specifies a cutpoint for binary classification
DDFM=	Specifies the degrees-of-freedom method
INCLUDE=	Includes effects in all models for model selection
LACKFIT	Requests the Hosmer and Lemeshow goodness-of-fit test
LINK=	Specifies the link function
NOCHECK	Suppresses checking for infinite parameters
NOINT	Suppresses the intercept
OFFSET=	Specifies the offset variable
PRIOR=	Specifies prior probabilities
RSQUARE	Requests a generalized coefficient of determination
START=	Includes effects in the initial model for model selection
STB	Displays standardized estimates

Response Variable Options

Response variable options determine how the HPLOGISTIC 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 HPLOGISTIC 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 HPLOGISTIC 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 value (formatted, if a format is applied) of the event category 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 with formatted value '1' represent events in the data. The probability modeled by the HPLOGISTIC procedure is thus the probability that the variable def takes on the (formatted) value '1'.

```
proc hplogistic 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 HPLOGISTIC run or in the DATA step that created the data set), the levels are ordered by their internal (numeric) value. The following table shows the interpretation of the ORDER= option:

ORDER=	Levels Sorted By
DATA	Order of appearance in the input data set
FORMATTED	External formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	Descending frequency count (levels with the most observations come first in the order)
FREQDATA	Order of descending frequency count; by order of appearance in the input data set when counts are tied
FREQFORMATTED	Order of descending frequency count; by formatted value (as above) when counts are tied
FREQINTERNAL	Order of descending frequency count; 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 on the SORT procedure in the *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 value (formatted if a format is applied) of the reference category 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 be constructed with the confidence level 1–*number*. The value of *number* must be between 0 and 1. By default, *number* is equal to the value of the ALPHA= option in the PROC HPLOGISTIC statement, or 0.05 if you do not specify that option.

ASSOCIATION
displays measures of association between predicted probabilities and observed responses for binary or binomial response models. These measures assess the predictive ability of the model. The displayed statistics are the concordance index *c* (the area under the ROC curve, AUC), Somers' *D* statistic (Gini's coefficient), the Goodman-Kruskal gamma statistic, and Kendall's tau-*a* statistic. For more information, see the section "Association Statistics" on page 4451.

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 with the ALPHA= option.

CTABLE< **=***SAS-data-set* **>**

OUTROC< **=***SAS-data-set* **>**

displays a table for binary or binomial response models that contains the frequencies of observations that are correctly and incorrectly classified as events and nonevents, the sensitivity, the 1–specificity, the positive and negative predictive values, and the correct classification rate. For more information, see the section "Classification Table and ROC Curves" on page 4449.

Classification is carried out by initially binning the predicted probabilities as discussed in the section "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 4452. The PRIOR= option does not change the reported predicted probabilities.

Because the number of cutpoints can be very large, you can store the table in an output data set. If you specify a PARTITION statement, then the statistics are computed by their roles, and a Role variable indicates to which partition the computations belong.

CUTPOINT=*value*

specifies a value between 0 and 1 used for classifying observations when you have a binary or binomial response variable. If the predicted probability of an observation equals or exceeds the cutpoint, the observation is classified as an event; otherwise it is classified as a nonevent. This option affects computation of the misclassification rate and the true positive and true negative fractions in the "Partition Fit Statistics" table. By default, CUTPOINT=0.5.

DDFM=RESIDUAL | NONE

specifies how degrees of freedom for statistical inference be determined in the "Parameter Estimates Table."

The HPLOGISTIC procedure always displays the statistical tests and confidence intervals in the "Parameter Estimates" tables in terms of a t test and a two-sided probability from a t distribution. With the DDFM= option, you can control the degrees of freedom of this t distribution and thereby switch between small-sample inference and large-sample inference based on the normal or chi-square distribution.

The default is DDFM=NONE, which leads to z-based statistical tests and confidence intervals. The HPLOGISTIC procedure then displays the degrees of freedom in the DF column as Infty, the p-values are identical to those from a Wald chi-square test, and the square of the t value equals the Wald chi-square statistic.

If you specify DDFM=RESIDUAL, the degrees of freedom are finite and determined by the number of usable frequencies (observations) minus the number of nonredundant model parameters. This leads to t-based statistical tests and confidence intervals. If the number of frequencies is large relative to the number of parameters, the inferences from the two degrees-of-freedom methods are almost identical.

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 the INCLUDE= option must be explanatory effects that are specified in the MODEL statement before the slash (/).

LACKFIT< (DFREDUCE=*r* NGROUPS=*G*) >
performs the Hosmer and Lemeshow goodness-of-fit test (Hosmer and Lemeshow 2000) for binary response models.

The subjects are divided into at most *G* groups of roughly the same size, based on the percentiles of the estimated probabilities. You can specify *G* as any integer greater than or equal to 5; by default, *G*=10. Let the actual number of groups created be g. The discrepancies between the observed and expected number of observations in these g groups are summarized by the Pearson chi-square statistic, which is then compared to a chi-square distribution with g–*r* degrees of freedom. You can specify a nonnegative integer *r* that satisfies $g–r \geq 1$; by default, *r*=2.

A small p-value suggests that the fitted model is not an adequate model. For more information, see the section "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 4452.

LINK=*keyword*
specifies the link function for the model. The *keywords* and the associated link functions are shown in Table 56.4.

Table 56.4 Built-in Link Functions of the HPLOGISTIC Procedure

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

For the probit and cumulative probit links, $\Phi^{-1}(\cdot)$ denotes the quantile function of the standard normal distribution.

If the response variable has more than two categories, the HPLOGISTIC procedure fits a model with a cumulative link function 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.

NOCHECK
disables the checking process that determines whether maximum likelihood estimates of the regression parameters exist. For more information, see the section "Existence of Maximum Likelihood Estimates" on page 4445.

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 with missing values for the offset variable are excluded from the analysis.

PRIOR=*SAS-data-set*

PRIOR=*number*

PEVENT=*number*

PRIOR=ALLDATA

specifies prior probabilities (prevalences) that are used for computing posterior predicted probabilities. When you know what percentage of the population has a rare event and you oversample that rare event, specifying the prior probabilities as the prevalence of events in your population enables you to produce posterior probabilities that reflect the population, not the data.

You can specify your priors in a SAS data set in which a _PRIOR_ column contains the prior probabilities. For events/trials MODEL statement syntax, this data set should also include an _OUTCOME_ variable that contains the values EVENT and NONEVENT; for single-trial syntax, this data set should include the response variable that contains the unformatted response categories. Each row of the data set contains a unique response variable level and its prior. For binary and binomial response models, you can instead specify the probability of an event as *number*. If you also specify a PARTITION statement, you can specify PRIOR=ALLDATA to compute the prevalences as the observed proportions of the response levels gathered across all the roles.

If your response Y takes values $i = 1, \ldots, k$ that have observed empirical probabilities $\mathrm{OldPrior}_i = \frac{n_i}{n}$, you specify priors Prior_i, and your model predicted probabilities are $\hat{p}_i$, then the posterior predicted probabilities Post_i are computed as

$$\mathrm{Post}_i = \frac{\hat{p}_i \frac{\mathrm{Prior}_i}{\mathrm{OldPrior}_i}}{\sum_{j=1}^{k} \hat{p}_j \frac{\mathrm{Prior}_j}{\mathrm{OldPrior}_j}}$$

The POST= option in the OUTPUT statement writes the posterior to the output data set. If your priors are identical to the empirical probabilities, then the posteriors are identical to the model-predicted probabilities.

The priors do not affect the model-fitting process, but instead modify the following statistics in the "Partition Fit Statistics" table: false positive fraction, false negative fraction, false response fraction (for multinomial response models), and misclassification rate. The "Classification" table statistics PPV, NPV, and Percent Correct are also adjusted as described in the section "Classification Table and ROC Curves" on page 4449.

RSQUARE

R2

requests a generalized coefficient of determination (R square, R^2) and a scaled version thereof for the fitted model. The results are added to the "Fit Statistics" table. For more information about the computation of these measures, see the section "Generalized Coefficient of Determination" on page 4448.

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.

STB

displays the standardized estimates for the parameters in the "Parameter Estimates" table. The standardized estimate of β_i is given by $\widehat{\beta}_i/(s/s_i)$, where s_i is the total sample standard deviation for the *i*th explanatory variable and

$$s = \begin{cases} \pi/\sqrt{3} & \text{LOGIT and GLOGIT links} \\ 1 & \text{PROBIT link} \\ \pi/\sqrt{6} & \text{CLOGLOG and LOGLOG links} \end{cases}$$

The sample standard deviations for parameters that are associated with CLASS variables are computed using their codings. The standardized estimates are not computed for the intercept parameters.

OUTPUT Statement

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

The OUTPUT statement creates a data set that contains observationwise statistics that PROC HPLOGISTIC computes after fitting the model. The variables in the input data set are *not* included in the output data set, in order to avoid data duplication for large data sets; however, variables that you specify in the ID statement or COPYVAR= option are included.

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

The output statistics are computed based on the final parameter estimates. If the optimization does not converge, then the output data set is not created.

When there are more than two response levels, values are computed only for variables that are named by the XBETA, POST, and PRED keywords; the other variables have missing values. These statistics are computed for every response category, and the automatic variable _LEVEL_ identifies the response category upon which the computed values are based. That is, every observation generates several rows in the output data set. 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, POST, and PRED statistics are computed even though these observations do not affect the model fit. This enables, for instance, 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.

COPYVAR=*variable*

COPYVARS=(*variables***)**

transfers one or more *variables* from the input data set to the output data set. Variables named in an ID statement are also copied from the input data set to the output data set.

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

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

The following are valid *keywords* for adding statistics to the OUTPUT data set:

LINP | XBETA

requests the linear predictor $\eta = \mathbf{x}'\boldsymbol{\beta}$. The default name is Xbeta.

PEARSON | PEARS | RESCHI

requests the Pearson residual, $\frac{\sqrt{wn}(y/n-\mu)}{\sqrt{\mu(1-\mu)}}$, where μ is the estimate of the predicted event probability, w is the weight of the observation, and n is the number of binomial trials (n=1 for binary observations). The default name is Pearson. This statistic is not computed for multinomial models.

POSTERIOR | POST

requests a numeric variable that contains the posterior predicted probability of each observation that is used in fitting the model. The default name is _POST_. If you do not specify the PRIOR option in the MODEL statement, then this value is the same as the predicted probability.

PREDICTED | PRED | P

requests predicted values (predicted probabilities of events) for the response variable. The default name is Pred.

RESIDUAL | RESID | R

requests the raw residual, $y - \mu$, where μ is the estimate of the predicted event probability. The default name is Residual. This statistic is not computed for multinomial models.

ROLE

requests a numeric variable that indicates the role played by each observation in fitting the model. The default name is _ROLE_. Table 56.5 shows how this variable is interpreted for each observation.

Table 56.5 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 variables.

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

ALLSTATS
adds all available statistics to the output data set.

OBSCAT
requests (for multinomial models) that observationwise statistics be produced for the response level only. If you do not specify the OBSCAT option 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-options* **;**

The PARTITION statement specifies how observations in the input data set are logically partitioned into disjoint subsets for model training, validation, and testing. For more information, see the section "Using Validation and Test Data" on page 4447. Either you can designate a variable in the input data set and a set of formatted values of that variable to determine the role of each observation, or you can specify proportions to use for random assignment of observations for each role.

You must specify one and only one of the following *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 formatted values of this variable that are used to assign observations roles are specified in the TEST=, TRAIN=, and VALIDATE= suboptions. If you do not specify the TRAIN= suboption, then all observations whose role is not determined by the TEST= or VALIDATE= 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 value less than or equal to 0, the seed is generated from reading the time of day from the computer's clock.

PERFORMANCE Statement

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

The PERFORMANCE statement defines performance parameters for multithreaded and distributed computing, passes variables about the distributed computing environment, and requests detailed results about the performance characteristics of the HPLOGISTIC procedure.

With the PERFORMANCE statement you can also control whether the HPLOGISTIC procedure executes in single-machine mode or distributed mode.

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

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 HPLOGISTIC 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=FORWARD	performs forward selection. This method starts with no effects in the model and adds effects.
METHOD=BACKWARD	performs backward elimination. This method starts with all effects in the model and deletes effects.
METHOD=BACKWARD(FAST)	performs fast backward elimination when SELECT=SL. This method starts with all effects in the model and deletes effects without refitting the model.
METHOD=STEPWISE	performs stepwise selection. This method is similar to the FORWARD method except that effects already in the model do not necessarily stay there.

The default criterion for the SELECT=, CHOOSE=, and STOP= options in the SELECTION statement is the significance level (SL), where effects enter and leave the model based on the significance level of an approximate chi-square test statistic. You can specify the following criteria in the SELECT=, CHOOSE=, and STOP= options:

AIC uses Akaike's information criterion (Akaike 1974).

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

BIC | SBC uses Schwarz' Bayesian criterion (Schwarz 1978).

SL uses the significance level of the score test as the criterion (not available for a CHOOSE= option).

VALIDATE uses the average square error (ASE) that is computed on the VALIDATE partition as the criterion (not available for a SELECT= option).

For more information, see the section "Information Criteria" on page 4448. 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.

NOTE: If you use the fast backward elimination method, then the –2 log likelihood, AIC, AICC, and BIC statistics are approximated at each step where the model is not refit, and hence they do not match the values that are computed when that model is fit outside the selection routine. Similarly, if you specify SELECT=AIC, AICC, or BIC, the selection criteria are estimated (Lawless and Singhal 1978), and hence they do not match the values that are computed when that model is fit outside the selection routine.

NOTE: The default model hierarchy method is HIERARCHY=NONE for the stepwise, forward, and fast backward selection methods. The backward elimination method always uses the HIERARCHY=SINGLE method.

When you specify the DETAILS= option in the SELECTION statement, the HPLOGISTIC procedure produces the following:

DETAILS=SUMMARY produces a summary table that shows the effect that is added or removed at each step along with the *p*-value and the SELECT=, CHOOSE=, and STOP= criteria. The summary table is produced by default if the DETAILS= option is not specified.

DETAILS=STEPS produces a detailed listing of all candidates at each step and their ranking in terms of the selection criterion for entry into or removal from the model.

DETAILS=ALL produces the preceding two tables and a table of selection details, which displays fit statistics for the model at each step of the selection process and an approximate chi-square score or likelihood ratio statistic.

WEIGHT Statement

WEIGHT *variable* **;**

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

Details: HPLOGISTIC Procedure

Missing Values

Any observation with 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 with the MISSING option in the CLASS statement. Observations with a nonpositive weight or with 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.

Response Distributions

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

- binary distribution
- binomial distribution
- multinomial distribution

The expressions for the log-likelihood functions of these distributions are given in the next section.

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

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

The value 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 ith 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, the HPLOGISTIC procedure fits the model as if the data had arisen from a binomial distribution. For example, the following statements fit a binomial regression model with regressors x1 and x2. The variables e and t represent the events and trials for the binomial distribution:

```
proc hplogistic;
   model e/t = x1 x2;
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.

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. With multinomial data, these outcomes are generically referred to as "categories" or levels.

Whenever the HPLOGISTIC procedure determines that the response variable 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.

Log-Likelihood Functions

The HPLOGISTIC 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 ith observation with 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.

Binary Distribution

The HPLOGISTIC procedure computes the log-likelihood function $l(\mu_i(\boldsymbol{\beta}); y_i)$ for the ith 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, μ_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 η_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 with the *response-options* in the MODEL statement, and you can choose the link function with the LINK= option in the MODEL statement.

If a WEIGHT statement is given 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 HPLOGISTIC procedure computes the log-likelihood function $l(\mu_i(\boldsymbol{\beta}); y_i)$ for the ith 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 ith observation, respectively. μ_i measures the probability of events (successes) in the underlying Bernoulli distribution whose aggregate follows the binomial distribution.

Multinomial Distribution

The multinomial distribution modeled by the HPLOGISTIC 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 ith observation is thus deceptively simple:

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

In this expression, J denotes the number of response categories (the number of possible outcomes) and μ_{ij} is the probability that the ith observation takes on the response value 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 zero. For more information, see the REF= *response-option* in the MODEL statement.

Existence of Maximum Likelihood Estimates

The likelihood equation for a logistic regression model does not always have a finite solution. Sometimes there is a nonunique maximum on the boundary of the parameter space, at infinity. The existence, finiteness,

and uniqueness of maximum likelihood estimates for the logistic regression model depend on the patterns of data points in the observation space (Albert and Anderson 1984; Santner and Duffy 1986).

Consider a binary response model. Let Y_j be the response of the jth subject, and let $\mathbf{x}_j$ be the vector of explanatory variables (including the constant 1 that is associated with the intercept). There are three mutually exclusive and exhaustive types of data configurations: complete separation, quasi-complete separation, and overlap.

Complete Separation There is a complete separation of data points if there exists a vector $\mathbf{b}$ that correctly allocates all observations to their response groups; that is,

$$\begin{cases} \mathbf{b}'\mathbf{x}_j > 0 & Y_j = 1 \\ \mathbf{b}'\mathbf{x}_j < 0 & Y_j = 2 \end{cases}$$

This configuration produces nonunique infinite estimates. If the iterative process of maximizing the likelihood function is allowed to continue, the log likelihood diminishes to 0, and the dispersion matrix becomes unbounded.

Quasi-complete Separation The data are not completely separable, but there is a vector $\mathbf{b}$ such that

$$\begin{cases} \mathbf{b}'\mathbf{x}_j \geq 0 & Y_j = 1 \\ \mathbf{b}'\mathbf{x}_j \leq 0 & Y_j = 2 \end{cases}$$

and equality holds for at least one subject in each response group. This configuration also yields nonunique infinite estimates. If the iterative process of maximizing the likelihood function is allowed to continue, the dispersion matrix becomes unbounded and the log likelihood diminishes to a nonzero constant.

Overlap If neither complete nor quasi-complete separation exists in the sample points, there is an overlap of sample points. In this configuration, the maximum likelihood estimates exist and are unique.

The HPLOGISTIC procedure uses a simple empirical approach to recognize the data configurations that lead to infinite parameter estimates. The basis of this approach is that any convergence method of maximizing the log likelihood must yield a solution that indicates complete separation, if such a solution exists. Upon convergence, if the predicted response equals the observed response for every observation, there is a complete separation of data points.

If the data are not completely separated, if an observation is identified to have an extremely large probability (≥ 0.95) of predicting the observed response, and if there have been at least eight iterations, then there are two possible situations. First, there is overlap in the data set, the observation is an atypical observation of its own group, and the iterative process stopped when a maximum was reached. Second, there is quasi-complete separation in the data set, and the asymptotic dispersion matrix is unbounded. If any of the diagonal elements of the dispersion matrix for the standardized observation vector (all explanatory variables standardized to zero mean and unit variance) exceeds 5,000, quasi-complete separation is declared. If either complete separation or quasi-complete separation is detected, a note is displayed in the procedure output.

Checking for quasi-complete separation is less foolproof than checking for complete separation. If neither type of separation is discovered and your parameter estimates have large standard errors, then this indicates that your data might be separable. The NOCHECK option in the MODEL statement turns off the process of checking for infinite parameter estimates.

Using Validation and Test Data

When you have sufficient data, you can subdivide your data into three parts called the training, validation, and test data. During the selection process, models are fit on the training data, and the prediction errors for the models so obtained are found by using the validation data. This prediction error on the validation data can be used to decide when to terminate the selection process and to decide which model. Finally, after a selected model has been obtained, the test set can be used to assess how the selected model generalizes on data that played no role in selecting the model.

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

You use a PARTITION statement to logically subdivide 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 subdivide the inData data set, reserving 50% for training and 25% each for validation and testing:

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

You can specify the SEED= option in the PARTITION statement to create the same partition data sets for a given number of compute nodes. However, changing the number of compute nodes changes the initial distribution of data, resulting in different partition data sets.

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 for each role. For example, the following statements assign roles to the observations in the inData data set that are 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 hplogistic 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 statistics are computed separately for each of these subsets. For more information, see the section "Partition Fit Statistics" on page 4448. For an illustration, see Example 56.4.

Using the Validation Statistic as the CHOOSE= Criterion

When you specify the CHOOSE=VALIDATE suboption of the METHOD= option in the SELECTION statement, the ASE is computed on the validation data for the models at each step of the selection process. The smallest model at any step that yields the smallest validation ASE is selected.

Using the Validation Statistic as the STOP= Criterion

When you specify the STOP=VALIDATE suboption of the METHOD= option in the SELECTION statement, the ASE is computed on the validation data for the models at each step of the selection process. At step k of the selection process, the best candidate effect to enter or leave the current model is determined and the validation ASE for this new model is computed. If this validation ASE is greater than the validation ASE for the model at step k, then the selection process terminates at step k.

Partition Fit Statistics

Specifying a PARTITION statement modifies the display of many tables by adding separate rows or columns for the training, validation, and test data sets. A "Partition Fit Statistics" table is also produced, and it displays the following statistics, which are useful for assessing the model and which should be very similar for the different roles when the training data are representative of the other data sets: average square error, misclassification rate, R^2, max-rescaled R^2, and McFadden's R^2. Binary and binomial response models also display the following statistics: area under the ROC curve, the Hosmer-Lemeshow test p-value, difference of means, Somers' D statistic, and the true positive and negative fractions. Polytomous response models also display the true fraction for each response level. For more information, see the sections "Model Fit and Assessment Statistics" on page 4448 and "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 4452.

Model Fit and Assessment Statistics

Information Criteria

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 sum 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 you do not specify a FREQ statement, F equals n, the number of observations used.

Generalized Coefficient of Determination

The goal of a coefficient of determination, also known as an R-square measure, is to express the agreement between a stipulated model and the data in terms of variation in the data that is explained by the model. In linear models, the R-square measure is based on residual sums of squares; because these are additive, a measure bounded between 0 and 1 is easily derived.

In more general models where parameters are estimated by the maximum likelihood principle, Cox and Snell (1989, pp. 208–209) and Magee (1990) proposed the following generalization of the coefficient of determination:

$$R^2 = 1 - \left\{ \frac{L(\mathbf{0})}{L(\widehat{\boldsymbol{\beta}})} \right\}^{\frac{2}{n}}$$

Here, $L(\mathbf{0})$ is the likelihood of the intercept-only model, $L(\widehat{\boldsymbol{\beta}})$ is the likelihood of the specified model, and n denotes the number of observations used in the analysis. This number is adjusted for frequencies if a FREQ statement is present and is based on the trials variable for binomial models.

As discussed in Nagelkerke (1991), this generalized R-square measure has properties similar to the coefficient of determination in linear models. If the model effects do not contribute to the analysis, $L(\widehat{\boldsymbol{\beta}})$ approaches $L(\mathbf{0})$ and R^2 approaches zero.

However, R^2 does not have an upper limit of 1. Nagelkerke suggested a rescaled generalized coefficient of determination, R^2_N, which achieves an upper limit of 1 by dividing R^2 by its maximum value:

$$R^2_{\max} = 1 - \{L(\mathbf{0})\}^{\frac{2}{n}}$$
$$R^2_N = \frac{R^2}{R^2_{\max}}$$

Another measure from McFadden (1974) is also bounded by 0 and 1:

$$R^2_M = 1 - \left(\frac{\log L(\widehat{\boldsymbol{\beta}})}{\log L(\mathbf{0})} \right)$$

If you specify the RSQUARE option in the MODEL statement, the HPLOGISTIC procedure computes R^2 and R^2_N. All three measures are computed for each data role when you specify a PARTITION statement.

These measures are most useful for comparing competing models that are not necessarily nested—that is, models that cannot be reduced to one another by simple constraints on the parameter space. Larger values of the measures indicate better models.

Classification Table and ROC Curves

For binary response data, the response Y is either an event or a nonevent; let the response Y take the value 1 for an event and 2 for a nonevent. From the fitted model, a predicted event probability $\hat{\pi}_i$ can be computed for each observation i. If the predicted event probability equals or exceeds some cutpoint value $z \in [0, 1]$, the observation is classified as an event; otherwise, it is classified as a nonevent. Suppose n_1 of n individuals experience an event, such as a disease, and the remaining $n_2 = n - n_1$ individuals are nonevents. The 2×2 decision matrix in Table 56.6 is obtained by cross-classifying the observed and predicted responses, where n_{ij} is the total number of observations that are observed to have $Y{=}i$ and are classified into j. In this table, let $Y = 1$ denote an observed event and $Y = 2$ denote a nonevent, and let $D = 1$ indicate that the observation is classified as an event and $D = 2$ denote that the observation is classified as a nonevent.

Table 56.6 Decision Matrix

	$D=1\ (\hat{\pi} \geq z)$	$D=2\ (\hat{\pi} < z)$	**Total**
$Y=1$ (event)	n_{11}	n_{12}	n_1
$Y=2$ (nonevent)	n_{21}	n_{22}	n_2

In the decision matrix, the *number of true positives*, n_{11}, is the number of event observations that are correctly classified as events; the *number of false positives*, n_{21}, is the number of nonevent observations that are incorrectly classified as events; the *number of false negatives*, n_{12}, is the number of event observations that are incorrectly classified as nonevents; and the *number of true negatives*, n_{22}, is the number of nonevent observations that are correctly classified as nonevents. The following statistics are computed from the preceding decision matrix:

Table 56.7 Statistics from the Decision Matrix with Cutpoint z

Statistic	**Equation**	**OUTROC Column**
Cutpoint	z	ProbLevel
Number of true positives	n_{11}	TruePos
Number of true negatives	n_{22}	TrueNeg
Number of false positives	n_{21}	FalsePos
Number of false negatives	n_{12}	FalseNeg
Sensitivity	n_{11}/n_1	TPF (true positive fraction)
1–specificity	n_{21}/n_2	FPF (false positive fraction)
Correct classification rate	$(n_{11}+n_{22})/n$	PercentCorrect (PC)
Misclassification rate	1 – PC	
Positive predictive value	$n_{11}/(n_{11}+n_{21})$	PPV
Negative predictive value	$n_{22}/(n_{12}+n_{22})$	NPV

The accuracy of the classification is measured by its ability to predict events and nonevents correctly. *Sensitivity* (TPF, true positive fraction) is the proportion of event responses that are predicted to be events. *Specificity* (1–FPF, true negative fraction) is the proportion of nonevent responses that are predicted to be nonevents.

You can also measure accuracy by how well the classification predicts the response. The *positive predictive value* (PPV) is the proportion of observations classified as events that are correctly classified. The *negative predictive value* (NPV) is the proportion of observations classified as nonevents that are correctly classified. The *correct classification rate* (PC) is the proportion of observations that are correctly classified.

If you also specify a PRIOR= option, then PROC HPLOGISTIC uses Bayes' theorem to modify the PPV, NPV, and PC as follows. Results of the classification are represented by two conditional probabilities: sensitivity, $\Pr(D=1|Y=1) = \frac{n_{11}}{n_1}$, and one minus the specificity, $\Pr(D=1|Y=2) = \frac{n_{21}}{n_2}$.

If the prevalence of the disease in the population $\Pr(Y=1)$ is provided by the value of the PRIOR= option, then the PPV, NPV, and PC are given by Fleiss (1981, pp. 4–5) as follows:

$$\begin{aligned}
\text{PPV} &= \Pr(Y=1|D=1) = \frac{\Pr(Y=1)\Pr(D=1|Y=1)}{\Pr(D=1|Y=2) + \Pr(Y=1)[\Pr(D=1|Y=1) - \Pr(D=1|Y=2)]} \\
\text{NPV} &= \Pr(Y=2|D=2) = \frac{[1-\Pr(D=1|Y=2)][1-\Pr(Y=1)]}{1-\Pr(D=1|Y=2) - \Pr(Y=1)[\Pr(D=1|Y=1) - \Pr(D=1|Y=2)]} \\
\text{PC} &= \Pr(Y=1|D=1) + \Pr(Y=2|D=2) \\
&= \Pr(D=1|Y=1)\Pr(Y=1) + \Pr(D=2|Y=2)[1-\Pr(Y=1)]
\end{aligned}$$

If you do not specify the PRIOR= option, then PROC HPLOGISTIC uses the sample proportion of diseased individuals; that is, $\Pr(Y=1) = n_1/n$. In such a case, the preceding values reduce to those in Table 56.7. Note that for a stratified sampling situation in which n_1 and n_2 are chosen a priori, n_1/n is not a desirable estimate of $\Pr(Y=1)$, so you should specify a PRIOR= option.

PROC HPLOGISTIC constructs the data for a receiver operating characteristic (ROC) curve by initially binning the predicted probabilities as discussed in the section "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 4452, then moving the cutpoint from 0 to 1 along the bin boundaries (so that the cutpoints correspond to the predicted probabilities), and then selecting those cutpoints where a change in the decision matrix occurs. The CTABLE option produces a table that includes these cutpoints and the statistics in Table 56.7 that correspond to each cutpoint. You can output this table to a SAS data set by specifying the CTABLE= option (see Table 56.7 for the column names), and you can display the ROC curve by using the SGPLOT procedure as shown in Example 56.2.

The area under the ROC curve (AUC), as determined by the trapezoidal rule, is given by the concordance index c, which is described in the section "Association Statistics" on page 4451.

For more information about the topics in this section, see Pepe (2003).

The "Partition Fit Statistics" table displays the misclassification rate, true positive fraction, true negative fraction, and AUC according to their roles. If you have a polytomous response, then instead of classifying according to a cutpoint, PROC HPLOGISTIC classifies the observation into the lowest response level (which has the largest predicted probability for that observation) and similarly computes a true response-level fraction.

Association Statistics

If you specify the ASSOCIATION option in the MODEL statement, PROC HPLOGISTIC displays measures of association between predicted probabilities and observed responses for binary or binomial response models. These measures assess the predictive ability of a model.

Of the n pairs of observations in the data set with different responses, let n_c be the number of pairs where the observation that has the lower-ordered response value has a lower predicted probability, let n_d be the number of pairs where the observation that has the lower-ordered response value has a higher predicted probability, and let $n_t = n - n_c - n_d$ be the rest. Let N be the sum of observation frequencies in the data. Then the following statistics are reported:

$$\begin{array}{lcl}
\text{concordance index } c \text{ (AUC)} & = & (n_c + 0.5n_t)/n \\
\text{Somers' } D \text{ (Gini coefficient)} & = & (n_c - n_d)/n \\
\text{Goodman-Kruskal gamma} & = & (n_c - n_d)/(n_c + n_d) \\
\text{Kendall's tau-}a & = & (n_c - n_d)/(0.5N(N-1))
\end{array}$$

Classification of the pairs is carried out by initially binning the predicted probabilities as discussed in the section "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 4452. The concordance index, c, is an estimate of the AUC, which is the area under the receiver operating characteristic (ROC) curve. If there are no ties, then Somers' D (Gini's coefficient) = $2c-1$.

If you specify a PARTITION statement, then PROC HPLOGISTIC displays the AUC and Somers' D in the "Association" and "Partition Fit Statistics" tables according to their roles.

Average Square Error

The average square error (ASE) is the average of the squared differences between the responses and the predictions. When you have a discrete number of response levels, the ASE is modified as shown in Table 56.8 (Brier 1950; Murphy 1973); it is also called the Brier score or Brier reliability:

Table 56.8 Average Square Error Computations

Response Type	ASE (Brier Score)
Polytomous	$\frac{1}{F}\sum_i f_i \sum_j (y_{ij} - \widehat{\pi}_{ij})^2$
Binary	$\frac{1}{F}\sum_i f_i (y_i(1-\widehat{\pi}_i)^2 + (1-y_i)\widehat{\pi}_i^2)$
Binomial	$\frac{1}{F}\sum_i f_i (r_i/t_i - \widehat{\pi}_i)^2$

In Table 56.8, $F = \sum_i f_i$, r_i is the number of events, t_i is the number of trials in binomial response models, and y_i=1 for events and 0 for nonevents in binary response models. For polytomous response models, y_{ij}=1 if the ith observation has response level j, and π_{ij} is the model-predicted probability of response level j for observation i.

Mean Difference

For a binary response model, write the mean of the model-predicted probabilities of event (Y=1) observations as $\overline{X}_1 = \frac{\sum_{i=1}^{n}(\pi_i | y_i=1)}{n_1}$ and of the nonevent (Y=2) observations as $\overline{X}_2 = \frac{\sum_{i=1}^{n}(\pi_i | y_i=2)}{n_1}$. The mean difference, or more precisely the difference of means, is $\overline{X}_1 - \overline{X}_2$, which Tjur (2009) relates to other R-square measures and calls the *coefficient of discrimination* because it is a measure of the model's ability to distinguish between the event and nonevent distributions. The difference of means is also the d' or Δm statistic (with unit standard error) that is discussed in the signal detection literature (McNicol 2005).

The Hosmer-Lemeshow Goodness-of-Fit Test

To evaluate the fit of the model, Hosmer and Lemeshow (2000) proposed a statistic that they show, through simulation, is distributed as chi-square when there is no replication in any of the subpopulations. This goodness-of-fit test is available only for binary response models.

The unit interval is partitioned into 2,000 equal-sized bins, and each observation i is placed into the bin that contains its estimated event probability. This effectively sorts the observations in increasing order of their estimated event probability.

The observations (and frequencies) are further combined into G groups. By default G = 10, but you can specify $G \geq 5$ with the NGROUPS= suboption of the LACKFIT option in the MODEL statement. Let F be

the total frequency. The target frequency for each group is $T = \lfloor F/G + 0.5 \rfloor$, which is the integer part of $F/G + 0.5$. Load the first group (g_j, $j = 1$) with the first of the 2,000 bins that has nonzero frequency f_1, and let the next nonzero bin have a frequency of f. PROC HPLOGISTIC performs the following steps for each nonzero bin to create the groups:

1. If $j = G$, then add this bin to group g_j.
2. Otherwise, if $f_j < T$ and $f_j + \lfloor f/2 \rfloor \leq T$, then add this bin to group g_j.
3. Otherwise, start loading the next group (g_{j+1}) with $f_{j+1} = f$, and set $j = j + 1$.

If the final group g_j has frequency $f_j < T/2$, then add these observations to the preceding group. The total number of groups actually created, g, can be less than G.

The Hosmer-Lemeshow goodness-of-fit statistic is obtained by calculating the Pearson chi-square statistic from the $2 \times g$ table of observed and expected frequencies. The statistic is written

$$\chi^2_{HL} = \sum_{j=1}^{g} \frac{(O_j - F_j \bar{\pi}_j)^2}{F_j \bar{\pi}_j (1 - \bar{\pi}_j)}$$

where, for the jth group g_j, $F_j = \sum_{i \in g_j} f_i$ is the total frequency of subjects, O_j is the total frequency of event outcomes, and $\bar{\pi}_j = \sum_{i \in g_j} f_i \hat{p}_i / F_j$ is the average estimated predicted probability of an event outcome. Let ϵ be the square root of the machine epsilon divided by 4,000, which is about 2.5E–12. Any $\bar{\pi}_j < \epsilon$ is set to ϵ; similarly, any $\bar{\pi}_j > 1 - \epsilon$ is set to $1 - \epsilon$.

The Hosmer-Lemeshow statistic is compared to a chi-square distribution with $g - r$ degrees of freedom. You can specify r with the DFREDUCE= suboption of the LACKFIT option in the MODEL statement. By default, $r = 2$, and to compute the Hosmer-Lemeshow statistic you must have $g - r \geq 1$. Large values of χ^2_{HL} (and small p-values) indicate a lack of fit of the model.

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 HPLOGISTIC procedure is determined by the number of CPUs on a machine and can be controlled by specifying the NTHREADS= option in the PERFORMANCE statement. This specification overrides the 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 HPLOGISTIC procedure allocates one thread per CPU by default.

The tasks that are multithreaded by the HPLOGISTIC procedure are primarily defined by dividing the data processed on a single machine among the threads—that is, the HPLOGISTIC procedure implements multithreading through a data-parallel model. For example, if the input data set has 1,000 observations and you are 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
- summarization of data for the Hosmer-Lemeshow test and association statistics

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 go into choosing a particular optimization technique for a particular problem are complex. Trial and error can be involved.

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, and, 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 56.9 shows which derivatives are required for each optimization technique.

Table 56.9 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 HPLOGISTIC 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 56.9.

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 Δ 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), and 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 that 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 required for most nonlinear mixed model applications.

The QUANEW technique implemented by the HPLOGISTIC 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 α 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 with 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 $\mathbf{s}^{(k)}$ as the linear combination of the steepest descent or ascent search direction $\mathbf{s}_1^{(k)}$ and a quasi-Newton search direction $\mathbf{s}_2^{(k)}$:

$$\mathbf{s}^{(k)} = \alpha_1 \mathbf{s}_1^{(k)} + \alpha_2 \mathbf{s}_2^{(k)}$$

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) and 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, many iterations are required to obtain a precise solution, but each of the CONGRA iterations is computationally cheap.

The CONGRA subroutine should be used for optimization problems with large p. For the unconstrained or boundary-constrained case, CONGRA 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 conjugate gradient 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 α 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 HPLOGISTIC produces. The output is organized into various tables, which are discussed in the order of 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 time (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 the HPLOGISTIC procedure determined based on your input and options. The "Model Information" table also displays the distribution of the data that is assumed by the HPLOGISTIC procedure. See the section "Response Distributions" on page 4443 for how the procedure determines the response distribution.

Class Level Information

The "Class Level Information" table lists the levels of every variable 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 with the ORDER= option in the CLASS statement. You can suppress the "Class Level Information" table completely or partially with the NOCLPRINT= option in the PROC HPLOGISTIC 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 read from the input data set and the number of observations 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 you specify a PARTITION statement, the table displays the values for each role.

Response Profile

The "Response Profile" table displays the ordered value from which the HPLOGISTIC 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 used in the analysis is reported. You can affect the ordering of the response values with the *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 you specify a PARTITION statement, the table displays the values for each role.

Selection Information

When you specify the SELECTION statement, the HPLOGISTIC procedure produces by default a series of tables with 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 HPLOGISTIC procedure produces the "Selection Summary" table with information about which effects were entered into or removed from the model at the steps of the model selection process. The statistic 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.

Stop Reason

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

Selection Reason

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

Selected Effects

When you specify the SELECTION statement, the HPLOGISTIC procedure produces a simple table that tells you which effects were selected into the final model.

Candidate Entry and Removal Details

When you specify the DETAILS=ALL or DETAILS=STEPS option in the SELECTION statement, the HPLOGISTIC procedure produces the "Candidate Entry and Removal Details" table, which displays the effect names and values of the criterion used to select entering or departing effects at each step of the selection process. For each step, the effects are displayed in sorted order from best to worst of the selection criterion.

Selection Details

When you specify the DETAILS=ALL option in the SELECTION statement, the HPLOGISTIC 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. If you specify SELECT=AIC, AICC, or BIC then the likelihood ratio chi-square statistic is displayed along with the estimated selection criteria; otherwise the score or Wald chi-square statistic is displayed. Fit statistics computed at each step are also displayed.

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 HPLOGISTIC procedure is normalized by default to enable comparisons across data sets with different sampling intensity. You can control normalization with the NORMALIZE= option in the PROC HPLOGISTIC statement.

If you specify the ITDETAILS option in the PROC HPLOGISTIC statement, information about the parameter estimates and gradients in the course of the optimization is added to the "Iteration History" table.

The "Iteration History" table is displayed by default unless you specify the NOITPRINT option or perform a model selection. To generate the history from a model selection process, specify the ITSELECT or ITDETAILS 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 assess convergence programmatically. 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 was exceeded, such as the maximum number of iterations or the maximum number of function evaluations. To modify these limits, see the MAXITER=, MAXFUNC=, and MAXTIME= options in the PROC HPLOGISTIC 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. For example, it displays the number of columns in the design matrix, the rank of the matrix, the largest number of design columns associated with an effect, the number of compute nodes in distributed mode, and the number of threads per node.

Fit Statistics

The "Fit Statistics" table displays a variety of likelihood-based measures of fit. All statistics are presented in "smaller is better" form. If you specify a PARTITION statement, the table displays the values for each role. The values displayed in the "Fit Statistics" table are not based on a normalized log-likelihood function. For more information, see the section "Information Criteria" on page 4448.

Partition Fit Statistics

If you specify the PARTITION statement, the "Partition Fit Statistics" table displays statistics for comparing the training, validation, and testing results. For more information about the statistics displayed in this table, see the sections "Partition Fit Statistics" on page 4448, "Model Fit and Assessment Statistics" on page 4448, and "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 4452.

Global Tests

The "Global Tests" table provides a statistical test for the hypothesis of whether the final model provides a better fit than a model without effects (an "intercept-only" model).

If you specify the NOINT option in the MODEL statement, the reference model is one where the linear predictor is 0 for all observations.

Partition for the Hosmer and Lemeshow Test

If you specify the LACKFIT option in the MODEL statement, the "Partition for the Hosmer and Lemeshow Test" table displays the grouping used in the Hosmer-Lemeshow test. If you specify a PARTITION statement, a table is displayed for each role. For more information, see the section "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 4452. For examples of using this partition, see Hosmer and Lemeshow (2000).

Hosmer and Lemeshow Goodness-of-Fit Test

If you specify the LACKFIT option in the MODEL statement, the "Hosmer and Lemeshow Goodness-of-Fit Test" table provides a test of the fit of the model; small p-values reject the null hypothesis that the fitted model is adequate. If you specify a PARTITION statement, a row is displayed for each role. For more information, see the section "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 4452.

Association Statistics

If you specify the ASSOCIATION option in the MODEL statement, the "Association Statistics" table displays the concordance index c (the area under the ROC curve, AUC), Somers' D statistic (Gini's coefficient), Goodman-Kruskal's gamma statistic, and Kendall's tau-a statistic. If you also specify a PARTITION statement, a row is displayed for each role. For more information, see the section "Association Statistics" on page 4451.

Classification Table

The "Classification" table is displayed if you specify the CTABLE option without specifying an output data set. If you also specify a PARTITION statement, a table is displayed for each role. For more information, see the section "Classification Table and ROC Curves" on page 4449.

Parameter Estimates

The parameter estimates, their estimated (asymptotic) standard errors, and p-values for the hypothesis that the parameter is 0 are presented in the "Parameter Estimates" table. If you request confidence intervals by using the CL or ALPHA= option in the MODEL statement, confidence limits are produced for the estimate on the linear scale.

By default, a normal z statistic is used to test the parameter estimates and is displayed in the "t Value" column with DF='Infty'. The square of the z statistic is a chi-square, so these p-values are identical to those from a Wald chi-square test. You can specify the DDFM=RESIDUAL option in the MODEL statement to obtain small-sample t tests.

ODS Table Names

Each table created by the HPLOGISTIC procedure has a name 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 56.10.

Table 56.10 ODS Tables Produced by PROC HPLOGISTIC

Table Name	Description	Statement	Option
Association	Association of predicted probabilities and observed responses	MODEL	ASSOCIATION
CandidateDetails	Details about candidates for entry into or removal from the model	SELECTION	DETAILS=STEP
Classification	Classification table	MODEL	CTABLE
ClassLevels	Level information from the CLASS statement	CLASS	Default
ConvergenceStatus	Status of optimization at conclusion of optimization	PROC	Default
Dimensions	Model dimensions	PROC	Default
FitStatistics	Fit statistics	PROC	Default
GlobalTests	Test of the model versus the null model	PROC	Default
IterHistory	Iteration history	PROC	Default or ITSELECT
LackFitChiSq	Hosmer-Lemeshow chi-square test results	MODEL	LACKFIT
LackFitPartition	Partition for the Hosmer-Lemeshow test	MODEL	LACKFIT
ModelInfo	Information about the modeling environment	PROC	Default
NObs	Number of observations read and used, and number of events and trials, if applicable	PROC	Default
ParameterEstimates	Solutions for the parameter estimates associated with effects in MODEL statements	PROC	Default
PartFitStats	Fit statistics for the data roles	PARTITION statement	Default
PerformanceInfo	Information about the high-performance computing environment	PROC	Default
ResponseProfile	Response categories and category modeled in models for binary and multinomial data	PROC	Default
SelectedEffects	List of effects selected into model	SELECTION	Default

Table 56.10 *continued*

Table Name	Description	Statement	Option
SelectionDetails	Details about model selection, including fit statistics by step	SELECTION	DETAILS=ALL
SelectionInfo	Information about the settings for model selection	SELECTION	Default
SelectionReason	Reason why the particular model was selected	SELECTION	Default
SelectionSummary	Summary information about model selection steps	SELECTION	Default
StopReason	Reason for termination of model selection	SELECTION	Default
Timing	Absolute and relative times for tasks performed by the procedure	PERFORMANCE	DETAILS

Examples: HPLOGISTIC Procedure

Example 56.1: Model Selection

The following HPLOGISTIC statements examine the same data as in the section "Getting Started: HPLOGISTIC Procedure" on page 4418, 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 related to model selection be produced.

```
proc hplogistic data=getStarted;
   class C;
   model y = C x1-x10;
   selection method=forward details=all;
run;
```

The model selection tables are shown in Output 56.1.1 through Output 56.1.4.

The "Selection Information" table in Output 56.1.1 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 56.1.1 Selection Information

The HPLOGISTIC 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 56.1.2 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 x8 made the most significant contribution to the model among the candidate effects ($p = 0.0381$). In step 2 the most significant contribution when adding an effect to a model that contains the intercept and x8 was made by x2. 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 56.1.2 Selection Summary Information

Selection Summary			
Step	**Effect Entered**	**Number Effects In**	**p Value**
0	**Intercept**	1	.
1	**x8**	2	0.0381
2	**x2**	3	0.0255

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

Selected Effects: Intercept x2 x8

The DETAILS=ALL option requests further detail information about the steps of the model selection. The "Candidate Details" table in Output 56.1.3 list all candidates for each step in the order of significance of their score tests. The effect with smallest p-value less than the SLE level of 0.05 is added in each step.

Output 56.1.3 Candidate Details

Candidate Entry and Removal Details

Step	Rank	Effect	Candidate For	p Value
1	1	x8	Entry	0.0381
	2	x2	Entry	0.0458
	3	x4	Entry	0.0557
	4	x9	Entry	0.1631
	5	C	Entry	0.1858
	6	x1	Entry	0.2715
	7	x10	Entry	0.4434
	8	x5	Entry	0.7666
	9	x3	Entry	0.8006
	10	x7	Entry	0.8663
	11	x6	Entry	0.9626
2	1	x2	Entry	0.0255
	2	x4	Entry	0.0721
	3	x9	Entry	0.1080
	4	C	Entry	0.1241
	5	x1	Entry	0.2778
	6	x10	Entry	0.5250
	7	x5	Entry	0.6993
	8	x7	Entry	0.7103
	9	x3	Entry	0.8743
	10	x6	Entry	0.9577

The DETAILS=ALL option also produces the "Selection Details" table, which provides fit statistics and the value of the score test chi-square statistic at each step.

Output 56.1.4 Selection Details

Selection Details

Step	Effect Entered	Number Effects In	Chi-Square	Pr > ChiSq	-2 LogL	AIC	AICC	BIC
0	Initial Model	1			123.82	125.82	125.86	128.43
1	x8	2	4.2986	0.0381	119.46	123.46	123.59	128.67
2	x2	3	4.9882	0.0255	114.40	120.40	120.65	128.21

Output 56.1.5 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 56.9. 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 discrepancy between the –2 log likelihood and the information criteria is less severe than previously noted.

Output 56.1.5 Fit Statistics and Null Test

Fit Statistics	
-2 Log Likelihood	114.40
AIC (smaller is better)	120.40
AICC (smaller is better)	120.65
BIC (smaller is better)	128.21

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	9.4237	2	0.0090

The parameter estimates of the selected model are given in Output 56.1.6. 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 56.1.6 Parameter Estimates

Parameter Estimates					
Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	0.8584	0.5503	Infty	1.56	0.1188
x2	-0.2502	0.1146	Infty	-2.18	0.0290
x8	1.7840	0.7908	Infty	2.26	0.0241

You can construct the prediction equation for this model from the parameter estimates as follows. The estimated linear predictor for an observation is

$$\widehat{\eta} = 0.8584 - 0.2503 \times x_2 + 1.7840 \times x_8$$

and the predicted probability that variable y takes on the value 0 is

$$\widehat{\Pr}(Y = 0) = \frac{1}{1 + \exp\{-\widehat{\eta}\}}$$

Example 56.2: Modeling Binomial Data

If $Y_1, \cdots, Y_n$ are independent binary (Bernoulli) random variables with common success probability π, then their sum is a binomial random variable. In other words, a binomial random variable with parameters n and π can be generated as the sum of n Bernoulli(π) random experiments. The HPLOGISTIC 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. 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, π) distribution, where the success probability π is a function of heating 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
;
```

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 ASSOCIATION option displays ordinal measures of association between the observed responses and predicted probabilities. The CTABLE=ROC option stores statistics that are used for evaluating the predictive power of the model in the Roc data set. The LACKFIT option produces the Hosmer and Lemeshow goodness-of-fit test for binary response models. The OUTPUT statement stores the linear predictors and the predicted probabilities in the Out data set along with the ID variable.

```
proc hplogistic data=Ingots;
   model r/n = Heat Soak Heat*Soak / association ctable=Roc lackfit;
   id Obsnum;
   output out=Out xbeta predicted=Pred;
run;
```

The "Performance Information" table in Output 56.2.1 shows that the procedure executes in single-machine mode. The example is executed on a single machine with the same number of cores as the number of threads used; that is, one computational thread was spawned per CPU.

Output 56.2.1 Performance Information

The HPLOGISTIC 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 56.2.2). This is the default link function in the HPLOGISTIC procedure for binary and binomial data. The procedure estimates the parameters of the model by a Newton-Raphson algorithm.

Output 56.2.2 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 56.2.2 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 56.2.3 displays the "Iteration History" and convergence status tables for this run. The HPLOGISTIC procedure converged after four iterations (not counting the initial setup iteration) and meets the GCONV= convergence criterion.

Output 56.2.3 Iteration History and Convergence Status

Iteration History				
Iteration	**Evaluations**	**Objective Function**	**Change**	**Max Gradient**
0	4	0.7676329445	.	6.378002
1	2	0.7365832479	0.03104970	0.754902
2	2	0.7357086248	0.00087462	0.023623
3	2	0.7357075299	0.00000109	0.00003
4	2	0.7357075299	0.00000000	5.42E-11

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

Output 56.2.4 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, since the rank of the crossproducts matrix equals the number of columns in **X**. All parameters are estimable and participate in the optimization.

Output 56.2.4 Dimensions in Binomial Logistic Regression

Dimensions	
Columns in X	4
Number of Effects	4
Max Effect Columns	1
Rank of Cross-product Matrix	4
Parameters in Optimization	4

Output 56.2.5 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 56.2.5 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

Output 56.2.6 shows the test of the global hypothesis that the effects jointly do not impact the probability of ingot readiness. The chi-square test statistic can be obtained by comparing the –2 log-likelihood value of the model with covariates to the value in the intercept-only model. The test is significant with a p-value of 0.0082. One or more of the effects in the model have a significant impact on the probability of ingot readiness.

Output 56.2.6 Null Test

Testing Global Null Hypothesis: BETA=0			
Test	**Chi-Square**	**DF**	**Pr > ChiSq**
Likelihood Ratio	11.7663	3	0.0082

Output 56.2.7 shows the tables that are produced when you specify the LACKFIT option in the MODEL statement. The first table displays the partition that PROC HPLOGISTIC uses to compute the chi-square test that is displayed in the second table; the large *p*-value does not reject the adequacy of the model fit. For more information, see the section "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 4452.

Output 56.2.7 Hosmer-Lemeshow Goodness-of-Fit Test

Partition for the Hosmer and Lemeshow Test

	Events		Nonevents		
Group	Observed	Expected	Observed	Expected	Total
1	0	0.24	34	33.76	34
2	0	0.47	43	42.53	43
3	0	0.66	52	51.34	52
4	2	0.46	31	32.54	33
5	0	0.48	31	30.52	31
6	0	0.36	19	18.64	19
7	1	1.94	55	54.06	56
8	4	1.59	40	42.41	44
9	1	1.63	42	41.37	43
10	4	4.17	28	27.83	32

Hosmer and Lemeshow Goodness-of-Fit Test

Chi-Square	DF	Pr > ChiSq
11.9771	8	0.1522

Output 56.2.8 displays the "Association Statistics" table, which is produced when you specify the ASSOCIATION option in the MODEL statement. The table contains four measures of association for assessing the predictive ability of a model. For more information, see the section "Association Statistics" on page 4451.

Output 56.2.8 Association of Observed Responses and Predicted Probabilities

Association Statistics

Concordance Index	Somers' D	Gamma	Tau-a
0.770556	0.541111	0.585759	0.032601

The "Parameter Estimates" table in Output 56.2.9 displays the estimates and standard errors of the model effects.

Output 56.2.9 Parameter Estimates

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	-5.9902	1.6666	Infty	-3.59	0.0003
Heat	0.09634	0.04707	Infty	2.05	0.0407
Soak	0.2996	0.7551	Infty	0.40	0.6916
Heat*Soak	-0.00884	0.02532	Infty	-0.35	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 56.2.10 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;
```

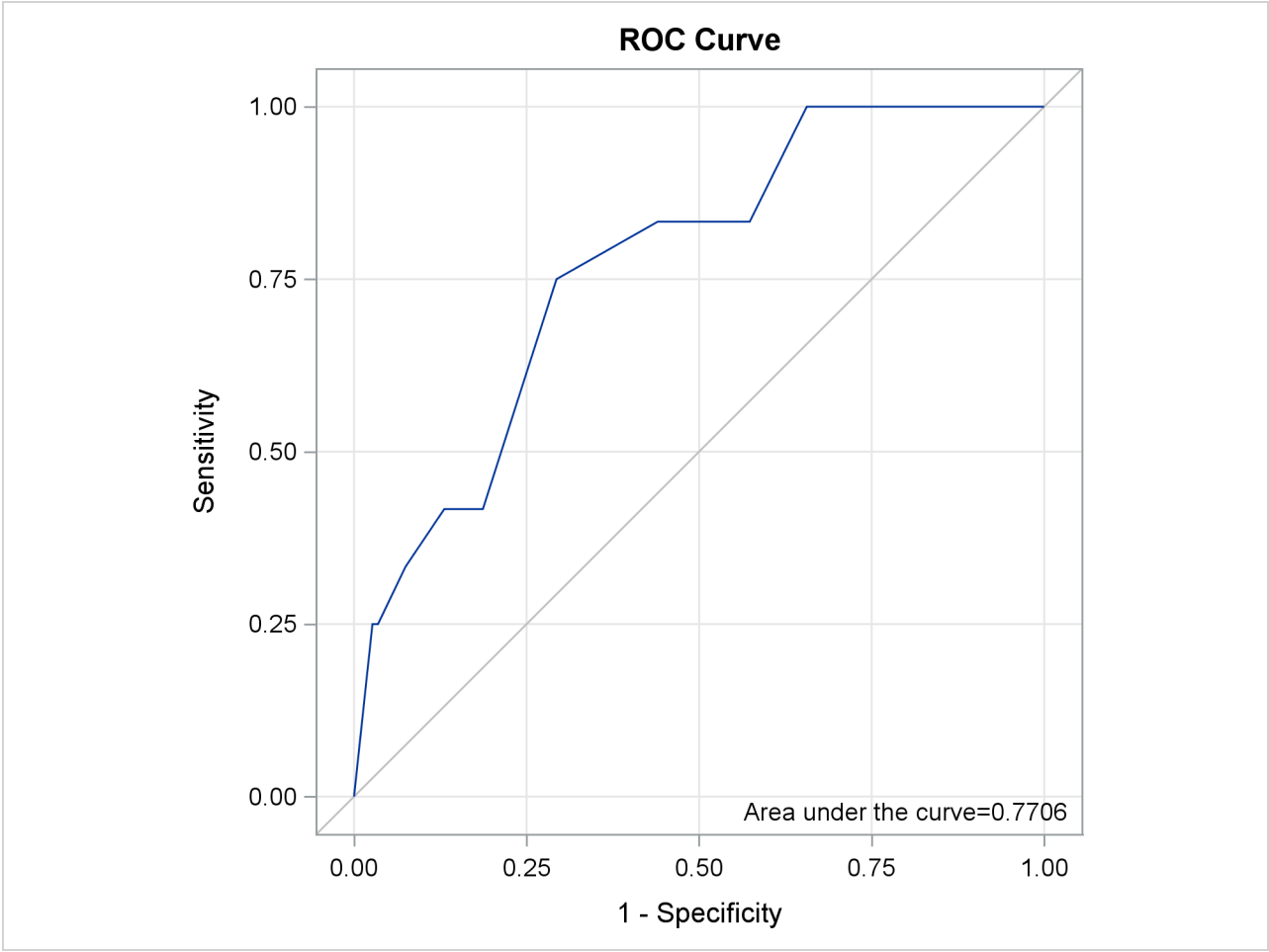
Output 56.2.10 Predicted Probability for Heat=14 and Soak=1.7

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

The CTABLE=ROC option computes statistics for binary response models (based on classifying observations according to whether their predicted probabilities exceed certain values) and stores the results in the Roc data set. For more information, see the section "Classification Table and ROC Curves" on page 4449. You can use this data set to display the ROC curve by using the SGPLOT procedure as follows:

```
proc sgplot data=Roc aspect=1 noautolegend;
   title 'ROC Curve';
   xaxis values=(0 to 1 by 0.25) grid offsetmin=.05 offsetmax=.05;
   yaxis values=(0 to 1 by 0.25) grid offsetmin=.05 offsetmax=.05;
   lineparm x=0 y=0 slope=1 / lineattrs=(color=ligr);
   series x=FPF y=TPF;
   inset 'Area under the curve=0.7706' / position=bottomright;
run;
```

Output 56.2.11 Receiver Operating Characteristics Curve



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

The following DATA step expands the Ingots data set with 12 events in 387 trials into a binary data set with 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 HPLOGISTIC statements fit the model with 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 HPLOGISTIC procedure models the probability that the variable y takes on the value ('1').

```
proc hplogistic data=Ingots_binary;
   model y(event='1') = Heat Soak Heat*Soak;
run;
```

Output 56.2.12 displays the "Performance Information", "Model Information," "Number of Observations," and the "Response Profile" tables. The data are now modeled as binary (Bernoulli distributed) with a logit link function. The "Response Profile" table shows that the binary response breaks down into 375 observations where y equals zero and 12 observations where y equals 1.

Output 56.2.12 Model Information in Binary Model

The HPLOGISTIC 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 56.2.13 displays the result for the test of the global null hypothesis and the parameter estimates. These results match those in Output 56.2.6 and Output 56.2.9.

Output 56.2.13 Null Test and Parameter Estimates

Testing Global Null Hypothesis: BETA=0			
Test	**Chi-Square**	**DF**	**Pr > ChiSq**
Likelihood Ratio	11.7663	3	0.0082

Output 56.2.13 *continued*

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	-5.9902	1.6666	Infty	-3.59	0.0003
Heat	0.09634	0.04707	Infty	2.05	0.0407
Soak	0.2996	0.7551	Infty	0.40	0.6916
Heat*Soak	-0.00884	0.02532	Infty	-0.35	0.7270

Example 56.3: Ordinal Logistic Regression

Consider a study of the effects of various cheese additives on taste. Researchers tested four cheese additives and obtained 52 response ratings for each additive. Each response was measured on a scale of nine categories ranging from strong dislike (1) to excellent taste (9). The data, given in McCullagh and Nelder (1989, p. 175) in the form of a two-way frequency table of additive by rating, are saved in the data set Cheese by using the following program. The variable y contains the response rating. The variable Additive specifies the cheese additive (1, 2, 3, or 4). The variable freq gives the frequency with which each additive received each rating.

```
data Cheese;
   do Additive = 1 to 4;
      do y = 1 to 9;
         input freq @@;
         output;
      end;
   end;
   label y='Taste Rating';
   datalines;
0  0  1  7  8  8 19  8  1
6  9 12 11  7  6  1  0  0
1  1  6  8 23  7  5  1  0
0  0  0  1  3  7 14 16 11
;
```

The response variable y is ordinally scaled. A cumulative logit model is used to investigate the effects of the cheese additives on taste. The following statements invoke PROC HPLOGISTIC to fit this model with y as the response variable and three indicator variables as explanatory variables, with the fourth additive as the reference level. With this parameterization, each Additive parameter compares an additive to the fourth additive.

```
proc hplogistic data=Cheese;
   freq freq;
   class Additive(ref='4') / param=ref ;
   model y=Additive;
   title 'Multiple Response Cheese Tasting Experiment';
run;
```

Results from the logistic analysis are shown in Output 56.3.1 through Output 56.3.3.

The "Response Profile" table in Output 56.3.1 shows that the strong dislike (y=1) end of the rating scale is associated with lower Ordered Values in the "Response Profile" table; hence the probability of disliking the additives is modeled.

Output 56.3.1 Proportional Odds Model Regression Analysis

Multiple Response Cheese Tasting Experiment

The HPLOGISTIC Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	4

Data Access Information			
Data	**Engine**	**Role**	**Path**
WORK.CHEESE	V9	Input	On Client

Model Information	
Data Source	WORK.CHEESE
Response Variable	y
Frequency Variable	freq
Class Parameterization	Reference
Distribution	Multinomial
Link Function	Cumulative Logit
Optimization Technique	Newton-Raphson with Ridging

Class Level Information			
Class	**Levels**	**Reference Value**	**Values**
Additive	4	4	1 2 3 4

Number of Observations Read	36
Number of Observations Used	28
Sum of Frequencies Read	208
Sum of Frequencies Used	208

Response Profile		
Ordered Value	**y**	**Total Frequency**
1	1	7
2	2	10
3	3	19
4	4	27
5	5	41
6	6	28
7	7	39
8	8	25
9	9	12

You are modeling the probabilities of levels of y having lower Ordered Values in the Response Profile Table.

Output 56.3.2 Proportional Odds Model Regression Analysis

Iteration History

Iteration	Evaluations	Objective Function	Change	Max Gradient
0	4	2.0668312595	.	0.137412
1	2	1.7319560317	0.33487523	0.062757
2	2	1.7105150048	0.02144103	0.008919
3	2	1.7099716191	0.00054339	0.00035
4	2	1.7099709251	0.00000069	6.981E-7
5	2	1.7099709251	0.00000000	2.98E-12

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

Dimensions	
Columns in X	11
Number of Effects	2
Max Effect Columns	8
Rank of Cross-product Matrix	11
Parameters in Optimization	11

Fit Statistics	
-2 Log Likelihood	711.35
AIC (smaller is better)	733.35
AICC (smaller is better)	734.69
BIC (smaller is better)	770.06

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	148.4539	3	<.0001

The positive value (1.6128) for the parameter estimate for Additive=1 in Output 56.3.3 indicates a tendency toward the lower-numbered categories of the first cheese additive relative to the fourth. In other words, the fourth additive tastes better than the first additive. Similarly, the second and third additives are both less favorable than the fourth additive. The relative magnitudes of these slope estimates imply the preference ordering: fourth, first, third, second.

Output 56.3.3 Proportional Odds Model Regression Analysis

Parameter Estimates

Parameter	Taste Rating	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	1	-7.0802	0.5640	Infty	-12.55	<.0001
Intercept	2	-6.0250	0.4764	Infty	-12.65	<.0001
Intercept	3	-4.9254	0.4257	Infty	-11.57	<.0001
Intercept	4	-3.8568	0.3880	Infty	-9.94	<.0001
Intercept	5	-2.5206	0.3453	Infty	-7.30	<.0001
Intercept	6	-1.5685	0.3122	Infty	-5.02	<.0001
Intercept	7	-0.06688	0.2738	Infty	-0.24	0.8071
Intercept	8	1.4930	0.3357	Infty	4.45	<.0001
Additive 1		1.6128	0.3805	Infty	4.24	<.0001
Additive 2		4.9646	0.4767	Infty	10.41	<.0001
Additive 3		3.3227	0.4218	Infty	7.88	<.0001

Example 56.4: Partitioning Data

The Sashelp.JunkMail data set comes from a study that classifies whether an email is junk email (coded as 1) or not (coded as 0). The data were collected by Hewlett-Packard Labs and donated by George Forman. The data set, which is specified in the following program, contains 4,601 observations, with 2 binary variables and 57 continuous explanatory variables. The response variable, Class, is a binary indicator of whether an email is considered spam or not. The partitioning variable, Test, is a binary indicator that is used to divide the data into training and testing sets. The 57 explanatory variables are continuous variables that represent frequencies of some common words and characters and lengths of uninterrupted sequences of capital letters in emails.

In the following program, the PARTITION statement divides the data into two parts. The training data have a Test value of 0 and contain about two-thirds of the data; the rest of the data are used to evaluate the fit. A forward selection method selects the best model based on the training observations.

```
proc hplogistic data=Sashelp.JunkMail;
   model Class(event='1')=Make Address All _3d Our Over Remove Internet Order
         Mail Receive Will People Report Addresses Free Business Email You
         Credit Your Font _000 Money HP HPL George _650 Lab Labs Telnet _857
         Data _415 _85 Technology _1999 Parts PM Direct CS Meeting Original
         Project RE Edu Table Conference Semicolon Paren Bracket Exclamation
         Dollar Pound CapAvg CapLong CapTotal;
   partition rolevar=Test(train='0' test='1');
   selection method=forward;
run;
```

Selected results from the analysis are shown in Output 56.4.1 through Output 56.4.3.

The "Number of Observations" and "Response Profile" tables in Output 56.4.1 are divided into training and testing columns.

Output 56.4.1 Partitioned Counts

Multiple Response Cheese Tasting Experiment

The HPLOGISTIC Procedure

Number of Observations			
Description	Total	Training	Testing
Number of Observations Read	4601	3065	1536
Number of Observations Used	4601	3065	1536

Response Profile				
Ordered Value	0 - Not Junk, 1 - Junk	Total Frequency	Training	Testing
1	0	2788	1847	941
2	1	1813	1218	595

You are modeling the probability that Class='1'.

The standard likelihood-based fit statistics for the selected model are displayed in the "Fit Statistics" table, with a column for each of the training and testing subsets.

Output 56.4.2 Partitioned Fit Statistics

Fit Statistics		
Description	**Training**	**Testing**
-2 Log Likelihood	1202.18	813.03
AIC (smaller is better)	1262.18	873.03
AICC (smaller is better)	1262.80	874.27
BIC (smaller is better)	1443.02	1033.14

More fit statistics are displayed in the "Partition Fit Statistics" table shown in Output 56.4.3. These statistics are computed for both the training and testing data and should be very similar between the two groups when the training data are representative of the testing data. The statistics include the likelihood-based R-square statistics, as well as several prediction-based statistics that are described in the sections "Model Fit and Assessment Statistics" on page 4448 and "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 4452. For this model, the values of the statistics seem similar between the two disjoint subsets.

Output 56.4.3 More Partitioned Fit Statistics

Partition Fit Statistics		
Statistic	**Training**	**Testing**
Area under the ROCC	0.9769	0.9653
Average Square Error	0.05467	0.06351
Hosmer-Lemeshow Test	3.74E-49	0
Misclassification Error	0.07145	0.07878
R-Square	0.6139	0.5533
Max-rescaled R-Square	0.8305	0.7508
McFadden's R-Square	0.7081	0.6035
Mean Difference	0.7596	0.7393
Somers' D	0.9538	0.9307
True Negative Fraction	0.9556	0.9416
True Positive Fraction	0.8875	0.8891

If you want to display the "Partition Fit Statistics" table without partitioning your data set, you must identify all your data as training data. One way to do this is to define the fractions for the other roles to be zero:

```
proc hplogistic data=Sashelp.JunkMail;
   model Class(event='1')= Our Over Remove Internet Order Will
         Free Business You Your Font _000 Money HP George Parts
         Meeting RE Edu Semicolon Exclamation Dollar CapAvg
         CapLong;
   partition fraction(test=0 validation=0);
run;
```

Another way is to specify a constant variable as the training role:

```
data JunkMail;
   set Sashelp.JunkMail;
   Role=0;
run;
proc hplogistic data=JunkMail;
   model Class(event='1')= Our Over Remove Internet Order Will
```

```
          Free Business You Your Font _000 Money HP George Parts
          Meeting RE Edu Semicolon Exclamation Dollar CapAvg
          CapLong;
   partition role=Role(train='0');
run;
```

The resulting "Partition Fit Statistics" table is shown in Output 56.4.4.

Output 56.4.4 All Data Are Training Data

Multiple Response Cheese Tasting Experiment

The HPLOGISTIC Procedure

Partition Fit Statistics	
Statistic	**Training**
Area under the ROCC	0.9724
Average Square Error	0.05910
Hosmer-Lemeshow Test	854E-220
Misclassification Error	0.07324
R-Square	0.5932
Max-rescaled R-Square	0.8034
McFadden's R-Square	0.6708
Mean Difference	0.7325
Somers' D	0.9448
True Negative Fraction	0.9570
True Positive Fraction	0.8803

References

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

Albert, A., and Anderson, J. A. (1984). "On the Existence of Maximum Likelihood Estimates in Logistic Regression Models." *Biometrika* 71:1–10.

Brier, G. W. (1950). "Verification of Forecasts Expressed in Terms of Probability." *Monthly Weather Review* 78:1–3.

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.

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

Fleiss, J. L. (1981). *Statistical Methods for Rates and Proportions*. 2nd ed. New York: John Wiley & Sons.

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

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. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag.

Hosmer, D. W., Jr., and Lemeshow, S. (2000). *Applied Logistic Regression*. 2nd ed. New York: John Wiley & Sons.

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

Lawless, J. F., and Singhal, K. (1978). "Efficient Screening of Nonnormal Regression Models." *Biometrics* 34:318–327.

Magee, L. (1990). "R^2 Measures Based on Wald and Likelihood Ratio Joint Significant Tests." *American Statistician* 44:250–253.

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

McFadden, D. (1974). "Conditional Logit Analysis of Qualitative Choice Behavior." In *Frontiers in Econometrics*, edited by P. Zarembka, 105–142. New York: Academic Press.

McNicol, D. (2005). *A Primer of Signal Detection Theory*. Mahwah, NJ: Lawrence Erlbaum Associates.

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

Murphy, A. H. (1973). "A New Vector Partition of the Probability Score." *Journal of Applied Meteorology* 12:595–600.

Nagelkerke, N. J. D. (1991). "A Note on a General Definition of the Coefficient of Determination." *Biometrika* 78:691–692.

Pepe, M. S. (2003). *The Statistical Evaluation of Medical Tests for Classification and Prediction*. New York: Oxford University Press.

Santner, T. J., and Duffy, D. E. (1986). "A Note on A. Albert and J. A. Anderson's Conditions for the Existence of Maximum Likelihood Estimates in Logistic Regression Models." *Biometrika* 73:755–758.

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

Tjur, T. (2009). "Coefficients of Determination in Logistic Regression Models—a New Proposal: The Coefficient of Discrimination." *American Statistician* 63:366–372.

Subject Index

Syntax Index