



THE  
POWER  
TO KNOW.

# **SAS/STAT<sup>®</sup> 9.22 User's Guide**

## **Shared Concepts and Topics**

### **(Book Excerpt)**



This document is an individual chapter from *SAS/STAT® 9.22 User's Guide*.

The correct bibliographic citation for the complete manual is as follows: SAS Institute Inc. 2010. *SAS/STAT® 9.22 User's Guide*. Cary, NC: SAS Institute Inc.

Copyright © 2010, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

**For a Web download or e-book:** Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

**U.S. Government Restricted Rights Notice:** Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, May 2010

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at [support.sas.com/publishing](http://support.sas.com/publishing) or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

## Chapter 19

# Shared Concepts and Topics

### Contents

---

Levelization of Classification Variables . . . . .	<b>407</b>
Parameterization of Model Effects . . . . .	<b>409</b>
GLM Parameterization of Classification Variables and Effects . . . . .	410
Intercept . . . . .	410
Regression Effects . . . . .	410
Main Effects . . . . .	410
Interaction Effects . . . . .	411
Nested Effects . . . . .	412
Continuous-Nesting-Class Effects . . . . .	412
Continuous-by-Class Effects . . . . .	413
General Effects . . . . .	413
Other Parameterizations . . . . .	414
EFFECT Statement ( <b>Experimental</b> ) . . . . .	<b>418</b>
Collection Effects . . . . .	420
Lag Effects . . . . .	420
Multimember Effects . . . . .	423
Polynomial Effects . . . . .	425
Spline Effects . . . . .	428
Splines and Spline Bases . . . . .	432
Truncated Power Function Basis . . . . .	433
B-Spline Basis . . . . .	434
EFFECTPLOT Statement . . . . .	<b>436</b>
Syntax: EFFECTPLOT Statement . . . . .	436
Dictionary of Options . . . . .	438
ODS Graphics: EFFECTPLOT Statement . . . . .	446
Examples: EFFECTPLOT Statement . . . . .	447
Example 19.1: A Saddle Surface . . . . .	447
Example 19.2: Unbalanced Two-Way ANOVA . . . . .	451
Example 19.3: Logistic Regression . . . . .	456
ESTIMATE Statement . . . . .	<b>462</b>
Syntax: ESTIMATE Statement . . . . .	462
Positional and Nonpositional Syntax for Coefficients in Linear Functions . . . . .	473
Joint Hypothesis Tests with Complex Alternatives, the Chi-Bar-Square Statistic . . . . .	476
ODS Table Names: ESTIMATE Statement . . . . .	477

ODS Graphics: ESTIMATE Statement . . . . .	478
LSMEANS Statement . . . . .	<b>479</b>
Syntax: LSMEANS Statement . . . . .	480
ODS Table Names: LSMEANS Statement . . . . .	494
ODS Graphics: LSMEANS Statement . . . . .	494
LSMESTIMATE Statement . . . . .	<b>496</b>
Syntax: LSMESTIMATE Statement . . . . .	497
ODS Table Names: LSMESTIMATE Statement . . . . .	507
ODS Graphics: LSMESTIMATE Statement . . . . .	507
NLOPTIONS Statement . . . . .	<b>508</b>
Syntax: NLOPTIONS Statement . . . . .	509
Choosing an Optimization Algorithm . . . . .	522
First- or Second-Order Algorithms . . . . .	522
Algorithm Descriptions . . . . .	523
SLICE Statement . . . . .	<b>526</b>
Syntax: SLICE Statement . . . . .	528
ODS Table Names: SLICE Statement . . . . .	529
STORE Statement . . . . .	<b>529</b>
Syntax: STORE Statement . . . . .	530
TEST Statement . . . . .	<b>530</b>
Syntax: TEST Statement . . . . .	531
ODS Table Names: TEST Statement . . . . .	532
Programming Statements . . . . .	<b>533</b>
References . . . . .	<b>535</b>

---



This chapter introduces a number of concepts that are common to two or more SAS/STAT procedures. Most sections display a listing of the procedures for which the shared topic is relevant.

## Levelization of Classification Variables

A classification variable is a variable that enters the statistical analysis or model not through its values, but through its levels. The process of associating values of a variable with levels is termed *levelization*.

This section covers in particular procedures that support a CLASS statement for specifying classification variables. Some of the concepts discussed also apply to procedures that use different syntax to request levelization of variables (for example, the CLASS() transformation in the TRANSREG procedure).

During the process of levelization, observations that share the same value are assigned to the same level. The manner in which values are grouped can be affected by the inclusion of formats. The sort order of the levels can be determined with the ORDER= option in the procedure statement. With the GENMOD, GLMSELECT, and LOGISTIC procedures, you can also control the sorting order separately for each variable in the CLASS statement.

Consider the data on nine observations in [Table 19.1](#). The variable A is integer valued, and the variable X is a continuous variable with a missing value for the fourth observations. The fourth and fifth columns of [Table 19.1](#) apply two different formats to the variable X.

**Table 19.1** Example Data for Levelization

Obs	A	x	FORMAT x 3.0	FORMAT x 3.1
1	2	1.09	1	1.1
2	2	1.13	1	1.1
3	2	1.27	1	1.3
4	3	.	.	.
5	3	2.26	2	2.3
6	3	2.48	2	2.5
7	4	3.34	3	3.3
8	4	3.34	3	3.3
9	4	3.14	3	3.1

By default, levelization of the variables groups observations by the formatted value of the variable, except for numerical variables for which no explicit format is provided. Numerical variables for which no explicit format is provided are sorted by their internal value. The levelization of the four columns in [Table 19.1](#) leads to the level assignment in [Table 19.2](#).

**Table 19.2** Values and Levels

Obs	A		X		FORMAT x 3.0		FORMAT x 3.1	
	Value	Level	Value	Level	Value	Level	Value	Level
1	2	1	1.09	1	1	1	1.1	1
2	2	1	1.13	2	1	1	1.1	1
3	2	1	1.27	3	1	1	1.3	2
4	3	2	.	.	.	.	.	.
5	3	2	2.26	4	2	2	2.3	3
6	3	2	2.48	5	2	2	2.5	4
7	4	3	3.34	7	3	3	3.3	6
8	4	3	3.34	7	3	3	3.3	6
9	4	3	3.14	6	3	3	3.1	5

The ORDER= option in the PROC statement specifies the sorting order for the levels of CLASS variables. When ORDER=FORMATTED (which is the default) is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. To order numeric class levels with no explicit format by their BEST12. formatted values, you can specify the BEST12. format explicitly for the CLASS variables.

The following table shows how values of the ORDER= option are interpreted.

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

For FORMATTED and INTERNAL values, 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*.

The GLMSELECT, LOGISTIC, and GENMOD procedures support a MISSING option in the CLASS statement. When this option is in effect, missing values (‘.’ for a numeric variable and blanks for a character variable) are included in the levelization and are assigned a level. Table 19.3 displays the results of levelizing the values in Table 19.1 when the MISSING option is in effect.

**Table 19.3** Values and Levels with MISSING Option

Obs	A		X		FORMAT x 3.0		FORMAT x 3.1	
	Value	Level	Value	Level	Value	Level	Value	Level
1	2	1	1.09	2	1	2	1.1	2
2	2	1	1.13	3	1	2	1.1	2
3	2	1	1.27	4	1	2	1.3	3
4	3	2	.	1	.	1	.	1
5	3	2	2.26	5	2	3	2.3	4
6	3	2	2.48	6	2	3	2.5	5
7	4	3	3.34	8	3	4	3.3	7
8	4	3	3.34	8	3	4	3.3	7
9	4	3	3.14	7	3	4	3.1	6

When the MISSING option is not specified, or for procedures whose CLASS statement does not support this option, it is important to understand the implications of missing values for your statistical analysis. When a SAS/STAT procedure levelizes the CLASS variables, an observation for which a CLASS variable has a missing value is excluded from the analysis. This is true regardless of whether the variable is used to form the statistical model. Consider, for example, the case where some observations contain missing values for variable A but the records for these observations are otherwise complete with respect to all other variables in the statistical models. The analysis results from the following statements do not include any observations for which variable A contains missing values, even though A is not specified in the MODEL statement:

```
class A B;
model y = B x B*x;
```

Many statistical procedures print a “Number of Observations” table that shows the number of observations read from the data set and the number of observations used in the analysis. Pay careful attention to this table—especially when your data set contains missing values—to ensure that no observations are unintentionally excluded from the analysis.

## Parameterization of Model Effects

The general form of a linear regression model is defined in Chapter 3, “Regression Models and Models with Classification Effects” as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

This section describes how matrices of regressor effects such as  $\mathbf{X}$  are constructed in SAS/STAT software. These constructions (*parameterization* rules) apply to regression models, models with classification effects, generalized linear models, and mixed models. The simplest and most general parameterization rules are the ones used in the GLM procedure, and they are discussed first. Several procedures also support alternate parameterizations of classification variables, including the CATMOD, GENMOD, GLMSELECT, LOGISTIC, PHREG, SURVEYLOGISTIC, and SURVEYPHREG procedures. These are discussed after the GLM parameterization of classification variables and model effects.

All modeling procedures that have a CLASS statement support classification variables and effects, and those procedures that additionally support the supplemental parameterizations have a PARAM= option in the CLASS statement.

## GLM Parameterization of Classification Variables and Effects

This section applies to the following procedures:

GAM, GENMOD, GLIMMIX, GLM, GLMPOWER, GLMSELECT, LIFEREG, LOGISTIC, MI, MIXED, MULLTEST, ORTHOREG, PHREG, PLS, QUANTREG, ROBUSTREG, SURVEYLOGISTIC, and SURVEYPHREG.

### Intercept

By default, SAS/STAT linear models automatically include a column of 1s in **X** which corresponds to an intercept parameter. In many procedures you can use the NOINT option in the MODEL statement to suppress this intercept. For example, the NOINT option is useful when the MODEL statement contains a classification effect and you want the parameter estimates to be in terms of the mean response for each level of that effect.

### Regression Effects

Numeric variables or polynomial terms that involve them can be included in the model as regression effects (covariates). The actual values of such terms are included as columns of the relevant model matrices. You can use the bar operator with a regression effect to generate polynomial effects. For example,  $X|X|X$  expands to  $X \ X^*X \ X^*X^*X$ , which is a cubic model.

### Main Effects

If a classification variable has  $m$  levels, the GLM parameterization generates  $m$  columns for its main effect in the model matrix. Each column is an indicator variable for a given level. The order of the columns is the sort order of the values of their levels and frequently can be controlled with the ORDER= option in the procedure or CLASS statement.

Table 19.4 is an example where  $\beta_0$  denotes the intercept and A and B are classification variables with two and three levels, respectively.

**Table 19.4** Example of Main Effects

Data		I	A		B		
A	B	$\beta_0$	A1	A2	B1	B2	B3
1	1	1	1	0	1	0	0
1	2	1	1	0	0	1	0
1	3	1	1	0	0	0	1
2	1	1	0	1	1	0	0
2	2	1	0	1	0	1	0
2	3	1	0	1	0	0	1

Typically, there are more columns for these effects than there are degrees of freedom to estimate them. In other words, the GLM parameterization of main effects is *singular*.

## Interaction Effects

Often a model includes interaction (crossed) effects to account for how the effect of a variable changes with the values of other variables. With an interaction, the terms are first reordered to correspond to the order of the variables in the CLASS statement. Thus, B\*A becomes A\*B if A precedes B in the CLASS statement. Then, the GLM parameterization generates columns for all combinations of levels that occur in the data. The order of the columns is such that the rightmost variables in the interaction change faster than the leftmost variables (Table 19.5). In the MIXED and GLIMMIX procedures, which support both fixed- and random-effects models, empty columns (that is, columns that would contain all 0s) are not generated for fixed effects, but they are generated for random effects.

**Table 19.5** Example of Interaction Effects

Data		I	A		B			A*B					
A	B	$\beta_0$	A1	A2	B1	B2	B3	A1B1	A1B2	A1B3	A2B1	A2B2	A2B3
1	1	1	1	0	1	0	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	1	0	0	0
2	1	1	0	1	1	0	0	0	0	0	1	0	0
2	2	1	0	1	0	1	0	0	0	0	0	1	0
2	3	1	0	1	0	0	1	0	0	0	0	0	1

In the preceding matrix, main-effects columns are not linearly independent of crossed-effects columns; in fact, the column space for the crossed effects contains the space of the main effect.

When your model contains many interaction effects, you might be able to code them more parsimoniously by using the bar operator (|). The bar operator generates all possible interaction effects. For example, A|B|C expands to A B A\*B C A\*C B\*C A\*B\*C. To eliminate higher-order interaction effects, use the at sign (@) in conjunction with the bar operator. For instance, A|B|C|D@2 expands to A B A\*B C A\*C B\*C D A\*D B\*D C\*D.

## Nested Effects

Nested effects are generated in the same manner as crossed effects. Hence, the design columns generated by the following two statements are the same (but the ordering of the columns is different):

```
model Y=A B(A);
```

```
model Y=A A*B;
```

The nesting operator in SAS/STAT software is more of a notational convenience than an operation distinct from crossing. Nested effects are typically characterized by the property that the nested variables never appear as main effects. The order of the variables within nesting parentheses is made to correspond to the order of these variables in the CLASS statement. The order of the columns is such that variables outside the parentheses index faster than those inside the parentheses, and the rightmost nested variables index faster than the leftmost variables (Table 19.6).

**Table 19.6** Example of Nested Effects

Data		I	A		B(A)					
A	B	$\beta_0$	A1	A2	B1A1	B2A1	B3A1	B1A2	B2A2	B3A2
1	1	1	1	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	0
2	1	1	0	1	0	0	0	1	0	0
2	2	1	0	1	0	0	0	0	1	0
2	3	1	0	1	0	0	0	0	0	1

## Continuous-Nesting-Class Effects

When a continuous variable nests or crosses with a classification variable, the design columns are constructed by multiplying the continuous values into the design columns for the classification effect (Table 19.7).

**Table 19.7** Example of Continuous-Nesting-Class Effects

Data		I	A		X(A)	
X	A	$\beta_0$	A1	A2	X(A1)	X(A2)
21	1	1	1	0	21	0
24	1	1	1	0	24	0
22	1	1	1	0	22	0
28	2	1	0	1	0	28
19	2	1	0	1	0	19
23	2	1	0	1	0	23

This model estimates a separate intercept and a separate slope for X within each level of A.

## Continuous-by-Class Effects

Continuous-by-class effects generate the same design columns as continuous-nesting-class effects. Table 19.8 shows the construction of the  $X*A$  effect. The two columns for this effect are the same as the columns for the  $X(A)$  effect in Table 19.7.

**Table 19.8** Example of Continuous-by-Class Effects

Data		I	X	A		X*A	
X	A	$\beta_0$	X	A1	A2	X*A1	X*A2
21	1	1	21	1	0	21	0
24	1	1	24	1	0	24	0
22	1	1	22	1	0	22	0
28	2	1	28	0	1	0	28
19	2	1	19	0	1	0	19
23	2	1	23	0	1	0	23

You can use continuous-by-class effects together with pure continuous effects to test for homogeneity of slopes.

## General Effects

An example that combines all the effects is  $X1*X2*A*B*C(D E)$ . The continuous list comes first, followed by the crossed list, followed by the nested list in parentheses. You should be aware of the sequencing of parameters when you use statements that depend on the ordering of parameters. Such statements include CONTRAST and ESTIMATE statements, which are used in a number of procedures to estimate and test functions of the parameters.

Effects might be renamed by the procedure to correspond to ordering rules. For example,  $B*A(E D)$  might be renamed  $A*B(D E)$  to satisfy the following:

- Classification variables that occur outside parentheses (crossed effects) are sorted in the order in which they appear in the CLASS statement.
- Variables within parentheses (nested effects) are sorted in the order in which they appear in the CLASS statement.

The sequencing of the parameters generated by an effect can be described by which variables have their levels indexed faster:

- Variables in the crossed list index faster than variables in the nested list.
- Within a crossed or nested list, variables to the right index faster than variables to the left.

For example, suppose a model includes four effects—A, B, C, and D—each having two levels, 1 and 2. If the CLASS statement is

```
class A B C D;
```

then the order of the parameters for the effect B\*A(C D), which is renamed A\*B(C D), is

$$\begin{aligned} A_1 B_1 C_1 D_1 &\rightarrow A_1 B_2 C_1 D_1 \rightarrow A_2 B_1 C_1 D_1 \rightarrow A_2 B_2 C_1 D_1 \rightarrow \\ A_1 B_1 C_1 D_2 &\rightarrow A_1 B_2 C_1 D_2 \rightarrow A_2 B_1 C_1 D_2 \rightarrow A_2 B_2 C_1 D_2 \rightarrow \\ A_1 B_1 C_2 D_1 &\rightarrow A_1 B_2 C_2 D_1 \rightarrow A_2 B_1 C_2 D_1 \rightarrow A_2 B_2 C_2 D_1 \rightarrow \\ A_1 B_1 C_2 D_2 &\rightarrow A_1 B_2 C_2 D_2 \rightarrow A_2 B_1 C_2 D_2 \rightarrow A_2 B_2 C_2 D_2 \end{aligned}$$

Note that first the crossed effects B and A are sorted in the order in which they appear in the CLASS statement so that A precedes B in the parameter list. Then, for each combination of the nested effects in turn, combinations of A and B appear. The B effect changes fastest because it is rightmost in the cross list. Then A changes next fastest, and D changes next fastest. The C effect changes most slowly because it is leftmost in the nested list.

## Other Parameterizations

This section applies to the following procedures:

CATMOD, GENMOD, GLMSELECT, LOGISTIC, PHREG, and SURVEYPHREG.

Some SAS/STAT procedures, including GENMOD, GLMSELECT, and LOGISTIC, support nonsingular parameterizations for classification effects. A variety of these nonsingular parameterizations are available. In most of these procedures you use the PARAM= option in the CLASS statement to specify the parameterization.

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

### EFFECT

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

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



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

The EFFECT parameterization is the default parameterization in the CATMOD procedure. See the section “[Generation of the Design Matrix](#)” on page 1681, in Chapter 28, “[The CATMOD Procedure](#),” for further details about parameterization of model effects with the CATMOD procedure.

#### GLM

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

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

Parameter estimates of CLASS main effects that use the GLM coding scheme estimate the difference in the effects of each level compared to the last level. See the previous section for details about the GLM parameterization of model effects.

#### ORDINAL

##### THERMOMETER

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

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

The first level of the effect is a control or baseline level. Parameter estimates of CLASS main effects, using the ORDINAL coding scheme, estimate the differences between effects of successive levels. When the parameters have the same sign, the effect is monotonic across the levels.

#### POLYNOMIAL

##### POLY

Three columns are created. The first represents the linear term ( $x$ ), the second represents the quadratic term ( $x^2$ ), and the third represents the cubic term

$(x^3)$ , where  $x$  is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, ... according to their sorting order. The design matrix columns for A are as follows.

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

## REFERENCE

### REF

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

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

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

The REFERENCE parameterization is also available through the MODEL statement in the CATMOD procedure. See the section “[Generation of the Design Matrix](#)” on page 1681, in Chapter 28, “[The CATMOD Procedure](#),” for further details about parameterization of model effects with the CATMOD procedure.

### ORTHEFFECT

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

Orthogonal Effect Coding			
A	Design Matrix		
	AOEFF1	AOEFF2	AOEFF3
1	1.41421	−0.81650	−0.57735
2	0	1.63299	−0.57735
5	0	0	1.73205
7	−1.41421	−0.81649	−0.57735

## ORTHORDINAL

## ORTHOTHERM

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

Orthogonal Ordinal Coding			
A	Design Matrix		
	AOORD1	AOORD2	AOORD3
1	-1.73205	0	0
2	0.57735	-1.63299	0
5	0.57735	0.81650	-1.41421
7	0.57735	0.81650	1.41421

## ORTHPOLY

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

Orthogonal Polynomial Coding			
A	Design Matrix		
	AOPOLY1	AOPOLY2	AOPOLY5
1	-1.15311	0.90712	-0.92058
2	-0.73380	-0.54041	1.47292
5	0.52414	-1.37034	-0.92058
7	1.36277	1.00363	0.36823

## ORTHREF

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

Orthogonal Reference Coding			
A	Design Matrix		
	AOREF1	AOREF2	AOREF3
1	1.73205	0	0
2	-0.57735	1.63299	0
5	-0.57735	-0.81650	1.41421
7	-0.57735	-0.81650	-1.41421

## EFFECT Statement (Experimental)

This section applies to the following procedures:

GLIMMIX, GLMSELECT, HPMIXED, LOGISTIC, ORTHOREG, PHREG, PLS, QUANTREG, ROBUSTREG, SURVEYLOGISTIC, and SURVEYREG.

The EFFECT statement enables you to construct special collections of columns for design matrices. These collections are referred to as *constructed effects* to distinguish them from the usual model effects that are formed from continuous or classification variables, as discussed in the section “GLM Parameterization of Classification Variables and Effects” on page 410. For example, the terms A, B, x, A\*x, A\*B, and sub in the following statements define fixed, random, and subject effects of the usual type in a mixed model:

```
proc glimmix;
  class A B sub;
  model y = A B x A*x;
  random A*B / subject=sub;
run;
```

A constructed effect, on the other hand, is assigned through the EFFECT statement. For example, in the following program, the EFFECT statement defines a constructed effect named spl:

```
proc glimmix;
  class A B SUB;
  effect spl = spline(x);
  model y = A B A*spl;
  random A*B / subject=sub;
run;
```

The columns of spl are formed from the data set variable x as a cubic B-spline basis with three equally spaced interior knots.

Each constructed effect corresponds to a collection of columns that are referred to by using the name you supply. You can specify multiple EFFECT statements, and all EFFECT statements must precede the MODEL statement.

The general syntax for the EFFECT statement with *effect-specification* is

**EFFECT** *effect-name* = *effect-type* (*var-list* < / *effect-options* > );

The name of the effect is specified after the EFFECT keyword. This name can appear in only one EFFECT statement and cannot be the name of a variable in the input data set. The *effect-type* is specified after an equal sign, followed by a list of variables within parentheses which are used in constructing the effect. *Effect-options* that are specific to an *effect-type* can be specified after a slash (/) following the variable list.

The following *effect-types* are available and are discussed in the following sections:

COLLECTION	is a collection effect that defines one or more variables as a single effect with multiple degrees of freedom. The variables in a collection are considered as a unit for estimation and inference.
LAG	is a classification effect in which the level that is used for a given period corresponds to the level in the preceding period.
MULTIMEMBER   MM	is a multimember classification effect whose levels are determined by one or more variables that appear in a CLASS statement.
POLYNOMIAL   POLY	is a multivariate polynomial effect in the specified numeric variables.
SPLINE	is a regression spline effect whose columns are univariate spline expansions of one or more variables. A spline expansion replaces the original variable with an expanded or larger set of new variables.

Table 19.9 summarizes important options for each type of EFFECT statement.

**Table 19.9** Important EFFECT Statement Options

Option	Description
<b>Options for Collection Effects</b>	
DETAILS	Displays the constituents of the collection effect
<b>Options for Lag Effects</b>	
DESIGNROLE=	Names a variable that controls to which lag design an observation is assigned
DETAILS	Displays the lag design of the lag effect
NLAG=	Specifies the number of periods in the lag
PERIOD=	Names the variable that defines the period
WITHIN=	Names the variable or variables that define the group within which each period is defined
<b>Options for Multimember Effects</b>	
NOEFFECT	Specifies that observations with all missing levels for the multimember variables should have zero values in the corresponding design matrix columns
WEIGHT=	Specifies the weight variable for the contributions of each of the classification effects
<b>Options for Polynomial Effects</b>	
DEGREE=	Specifies the degree of the polynomial
MDEGREE=	Specifies the maximum degree of any variable in a term of the polynomial
STANDARDIZE=	Specifies centering and scaling suboptions for the variables that define the polynomial

**Table 19.9** *continued*

Option	Description
<b>Options for Spline Effects</b>	
<b>BASIS=</b>	Specifies the type of basis (B-spline basis or truncated power function basis) for the spline expansion
<b>DEGREE=</b>	Specifies the degree of the spline transformation
<b>KNOTMETHOD=</b>	Specifies how to construct the knots for spline effects

## Collection Effects

**EFFECT** *name*=**COLLECTION** (*var-list* < / **DETAILS** >) ;

You use a collection effect to define a set of variables that are treated as a single effect with multiple degrees of freedom. The variables in *var-list* can be continuous or classification variables. The columns in the design matrix that are contributed by a collection effect are the design columns of its constituent variables in the order in which they appear in the definition of the collection effect. If you specify the **DETAILS** option, then a table that shows the constituents of the collection effect is displayed.

## Lag Effects

**EFFECT** *name*=**LAG** (*variable* / *lag-options*) ;

A lag effect is a classification effect for the **CLASS** variable that is given after the keyword **LAG**. A lag effect is used to represent the effect of a previous value of the lagged variable when there is some inherent ordering of the observations of this variable. A typical example where lag effects are useful is a study in which different subjects are given sequences of treatments and you want to investigate whether the treatment in the previous period is important in understanding the outcome in the current period. You can do this by including a lagged treatment effect in your model.

The precise definition of a **LAG** effect depends on a subdivision of the data into disjoint subsets, often referred to as “subjects,” and an ordering into units called “periods” of the observations within a subject. For an observation that belongs to a given subject and at a given period, the design matrix columns of the lagged variable are the usual design matrix columns of that variable except for the observation at the preceding period for that subject. Observations at the initial period do not have a preceding value, and so the design matrix columns of the lag effect for these observations are set to zero. You can also define lag effects where the number of periods that are lagged is greater than one. If the number of periods that are lagged is  $n$ , then the design matrix columns of observations in periods less than or equal to  $n$  are set to zero. The design matrix columns that correspond to a subject at period  $p$ , where  $p > n$ , are the usual design matrix columns of the lagged variable for that subject at period  $p - n$ .

A convenient way to represent the organization of observations into subjects and periods is to form the lag design matrix. The rows and columns of this matrix correspond to the subjects and periods respectively. The lag design matrix entry is the treatment for the corresponding subject and period. In a valid lag design there is at most one observation for a given period and subject. For example, the following set of treatments by subject and period form a valid lag design:

Subject	Period	Treatment
Sheila	1	B
Joey	1	A
Athena	1	A
Gelindo	1	A
Sheila	2	C
Joey	2	A
Athena	2	.
Gelindo	2	B
Sheila	3	B
Joey	3	C
Athena	3	A
Gelindo	3	B

The associated lag design matrix is

Subject	--Period--		
	1	2	3
Athena	A		A
Gelindo	A	B	B
Joey	A	A	C
Sheila	B	C	B

Note that the subject Athena did not receive a treatment at period 2, and so the corresponding entry in the lag design matrix is missing. You can define a lag effect for this lag design with the following statements:

```
CLASS treatment;
EFFECT Lag = LAG( treatment / WITHIN=subject PERIOD=period);
```

When GLM coding is used for the CLASS variable treatment, the design matrix columns Lag\_A, Lag\_B, and Lag\_C for the constructed effect Lag are as follows:

Subject	period	treatment	Lag_A	Lag_B	Lag_C
Athena	1	A	0	0	0
Athena	2		1	0	0
Athena	3	A	.	.	.
Gelindo	1	A	0	0	0
Gelindo	2	B	1	0	0
Gelindo	3	B	0	1	0
Joey	1	A	0	0	0
Joey	2	A	1	0	0
Joey	3	C	1	0	0
Sheila	1	B	0	0	0
Sheila	2	C	0	1	0
Sheila	3	B	0	0	1

The design matrix columns for each subject at period 1 are all zero because there are no lagged observations for period 1. You can also see that the design matrix columns at period 3 for subject Athena are missing because Athena did not receive a treatment at period 2. Nevertheless, the design matrix columns for Athena at period 2 are nonmissing and correspond to the treatment “A” that she received in period 1.

The following *lag-options* are required:

**PERIOD=***variable*

specifies the period variable of the LAG design. The number of periods is the number of unique formatted values of the PERIOD= variable, and the ordering of the period is formed by sorting these formatted values in ascending order. You must specify a PERIOD= variable.

**WITHIN=***(variables)*

**WITHIN=***variable*

specifies a variable (or a list of variables within parentheses) that defines the subject grouping of the lag design. If there is only one WITHIN= *variable*, then the parentheses are not required. Each *subject* is defined by the unique set of formatted values of the *variables* in the WITHIN= list. The subjects are sorted in ascending lexicographic order. You must specify a WITHIN= variable.

You can also specify the following *lag-options*:

**DESIGNROW=***variable*

specifies a numeric variable that is used to subset observations into a fitting group in which the value of the DESIGNROW= variable is nonzero and a second group in which the value of the specified *variable* is zero. The observations in the fitting group are used to form the LAG design matrix that is used in fitting the model. The LAG design that corresponds to the non-fitting group is used when scoring observations in the input data set that do not belong to the fitting group. This option is useful when you want to obtain predicted values in an output data set for observations that are not used in fitting the model. If you do not specify a DESIGNROW= *variable*, then all observations are assigned to the fitting group.

**DETAILS**

requests a table that shows the lag design matrix of the lag effect.

**NLAG=** *n*

specifies the number of lags. By default NLAG=1.



## Multimember Effects

**EFFECT** *name*=**MULTIMEMBER** (*var-list* < / *mm-options*> ) ;

**EFFECT** *name*=**MM** (*var-list* < / *mm-options*> ) ;

A multimember effect is formed from one or more classification variables in such a way that each observation can be associated with one or more levels of the union of the levels of the classification variables. In other words, a multimember effect is a classification-type effect with possibly more than one nonzero column entry for each observation. Multimember effects are useful, for example, in modeling the following:

- nurses' effects on patient recovery in hospitals
- teachers' effects on student scores
- lineage effects in genetic studies (see [Example 38.16](#) in Chapter 38, "The GLIMMIX Procedure," for an application with random multimember effects in a genetic diallel experiment)

The levels of a multimember effect consist of the union of formatted values of the variables that define this effect. Each such level contributes one column to the design matrix. For each observation, the value that corresponds to each level of the multimember effect in the design matrix is the number of times that this level occurs for the observation.

For example, the following data provide teacher information and end-of-year test scores for students after two semesters:

Student	Score	Teacher1	Teacher2
Mary	87	Tobias	Cohen
Tom	89	Rodriguez	Tobias
Fred	82	Cohen	Cohen
Jane	88	Tobias	.
Jack	99	.	.

For example, Mary had different teachers in the two semesters, Fred had the same teacher in both semesters, and Jane received instruction only in the first semester.

You can model the effect of the teachers on student performance by using a multimember effect specified as follows:

```
CLASS teacher1 teacher2;
EFFECT teacher = MM(teacher1 teacher2);
```

The levels of the teacher effect are Cohen, Rodriguez, and Tobias, and the associated design matrix columns are as follows:

Student	Cohen	Rodriguez	Tobias
Mary	1	0	1
Tom	0	1	1
Fred	2	0	0
Jane	0	0	1
Jack	.	.	.

You can specify the following *mm-options* after a slash (/):

### DETAILS

requests a table that shows the levels of the multimember effect.

### NOEFFECT

specifies that, for observations with all missing levels of the multimember variables, the values in the corresponding design matrix columns be set to zero. If, in the preceding example, the teacher effect is defined by

```
EFFECT teacher = MM(teacher1 teacher2 / noeffect);
```

then the associated design matrix columns values for Jack are all zero. This enables you to include Jack in the analysis even though there is no effect of teachers on his performance.

A situation where it is important to designate observations as having no effect due to a classification variable is the analysis of crossover designs, where lagged treatment levels are used to model the carryover effects of treatments between periods. Since there is no carryover effect for the first period, the treatment lag effect in a crossover design can be modeled with a multimember effect that consists of a single classification variable and the NOEFFECT option, as in the following statements:

```
CLASS Treatment lagTreatment;  
EFFECT Carryover = MM(lagTreatment / noeffect);
```

The lagTreatment variable contains a missing value for the first period. Otherwise, it contains the value of the treatment variable for the preceding period.

### STDIZE

specifies that for each observation, the entries in the design matrix that corresponds to the multimember effect be scaled to have a sum of one.

### WEIGHT=*wght-list*

specifies numeric variables used to weigh the contributions of each of the classification effects that define the constructed multimember effect. The number of variables in *wght-list* must match the number of classification variables that define the effect.

## Polynomial Effects

**EFFECT** *name*=POLYNOMIAL (*var-list* </ *polynomial-options*>) ;

**EFFECT** *name*=POLY (*var-list* </ *polynomial-options*>) ;

The variables in *var-list* must be numeric. A design matrix column is generated for each term of the specified polynomial. By default, each of these terms is treated as a separate effect for the purpose of model building. For example, the statements

```
proc glmselect;
  effect MyPoly = polynomial(x1-x3/degree=2);
  model y = MyPoly;
run;
```

yield the identical analysis to the statements

```
proc glmselect;
  model y = x1 x2 x3 x1*x1 x1*x2 x1*x3 x2*x2 x2*x3 x3*x3;
run;
```

You can specify the following *polynomial-options* after a slash (/):

### DEGREE=*n*

specifies the degree of the polynomial. The degree must be a positive integer. The degree is typically a small integer, such as 1, 2, or 3. The default is DEGREE=1.

### DETAILS

requests a table that shows the details of the specified polynomial, including the number of terms generated. If you also specify the [STANDARDIZE](#) option, then a table that shows the standardization details is also produced.

### LABELSTYLE=(*style-opts*)

#### LABELSTYLE=*style-opt*

specifies how the terms in the polynomial are labeled. By default, powers are shown with ^ as the exponentiation operator and \* as the multiplication operator. For example, a polynomial term such as  $x_1^3 x_2 x_3^2$  is labeled `x1^3*x2*x3^2`. You can change the style of the label by using the following *style-opts* within parentheses. If you specify a single *style-opt*, then you can omit the enclosing parentheses.

### EXPAND

specifies that each variable with an exponent greater than 1 be written as products of that variable. For example, the term  $x_1^3 x_2 x_3^2$  receives the label `x1*x1*x1*x2*x3*x3`.

### EXPONENT <=*quoted string*>

specifies that each variable with an exponent greater than 1 be written using exponential notation. By default, the symbol ^ is used as the exponentiation operator. If you supply the optional quoted string after an equal sign, then that string is used as the exponentiation operator.

For example, if you specify

```
LABELSTYLE=(EXPONENT="**")
```

then the term  $x_1^3 x_2 x_3^2$  receives the label `x1**3*x2*x3**2`.

### **INCLUDENAME**

specifies that the name of the effect followed by an underscore be used as a prefix for term labels. For example, the following statement generates terms with labels `MyPoly_x1` and `MyPoly_x1^2`:

```
EFFECT MyPoly=POLYNOMIAL(x1/degree=2 labelstyle=INCLUDENAME)
```

The `INCLUDENAME` option is ignored if you also specify the `NOSEPARATE` option in the `EFFECT=POLYNOMIAL` statement.

### **PRODUCTSYMBOL=NONE** | *quoted string*

specifies that the supplied string be used as the product symbol. For example, the following statement generates terms with labels `x1`, `x2`, and `x1 x2`:

```
EFFECT MyPoly=POLYNOMIAL(x1 x2 / degree=2 mdegree=1  
labelstyle=(PRODUCTSYMBOL=" "))
```

If you specify `PRODUCTSYMBOL=NONE`, then the labels are formed by juxtaposing the constituent variable names.

### **MDEGREE=n**

specifies the maximum degree of any variable in a term of the polynomial. This degree must be a positive integer. The default is the degree of the specified polynomial. For example, the following statement generates the terms  $x_1$ ,  $x_2$ ,  $x_1^2$ ,  $x_1 x_2$ ,  $x_2^2$ ,  $x_1^2 x_2$ ,  $x_1 x_2^2$  and  $x_1^2 x_2^2$ :

```
EFFECT MyPoly=POLYNOMIAL(x1 x2/degree=4 MDEGREE=2);
```

### **NOSEPARATE**

specifies that the polynomial be treated as a single effect with multiple degrees of freedom. The effect name that you specify is used as the constructed effect name, and the labels of the terms are used as labels of the corresponding parameters.

### **STANDARDIZE** <(centerscale-opts)> <= standardize-opt >

specifies that the variables that define the polynomial be standardized. By default, the standardized variables receive prefix “s\_” in the variable names.

You can use the following *centerscale-opts* to specify how the center and scale are estimated:

**METHOD=MOMENTS**

specifies that the center be estimated by the variable mean and the scale be estimated by the standard deviation. If a weight variable is specified using a **WEIGHT** statement, the observations with invalid weights are ignored when forming the mean and standard deviation, but the weights are otherwise not used. Only observations that are used in performing the analysis are used for the standardization.

**METHOD=RANGE**

specifies that the center be estimated by the midpoint of the variable range and the scale be estimated as half the variable range. Any observation that has a missing value for any regressor used in the model is ignored when computing the range of variables in a polynomial effect. Observations with valid regressor values but missing or invalid values of frequency variables, weight variables, or dependent variables are used in computing variable ranges. The default (if you do not specify the **METHOD=** suboption) is **METHOD=RANGE**.

**METHOD=WMOMENTS**

is the same as **METHOD=MOMENTS** except that weighted means and weighted standard deviations are used.

Let

- $n$  = number of observations used in the analysis
- $w$  = weight variable
- $f$  = frequency variable
- $x$  = variable to be standardized
- $x_{(n)}$  =  $\text{Max}_{i=1}^n (x_i)$
- $x_{(1)}$  =  $\text{Min}_{i=1}^n (x_i)$
- $F$  = sum of frequencies  
=  $\sum_{i=1}^n f_i$
- $WF$  = sum of weighted frequencies  
=  $\sum_{i=1}^n w_i f_i$

Table 19.10 shows how the center and scale are computed for each of the supported methods.

**Table 19.10** Center and Scale Estimates by Method

Method	Center	Scale
Range	$(x_{(n)} + x_{(1)})/2$	$(x_{(n)} - x_{(1)})/2$
Moments	$\bar{x} = \sum_{i=1}^n f_i x_i / F$	$\sqrt{\sum_{i=1}^n f_i (x_i - \bar{x})^2 / (F - 1)}$
WMoments	$\bar{x}_w = \sum_{i=1}^n w_i f_i x_i / WF$	$\sqrt{\sum_{i=1}^n w_i f_i (x_i - \bar{x}_w)^2 / (F - 1)}$

**PREFIX=NONE** | *quoted-string*

specifies the prefix that is appended to standardized variables when forming the term labels. If you omit this option, the default prefix is “s\_”. If you specify PREFIX=NONE, then standardized variables are not prefixed.

You can control whether the standardization is to center, scale, or both center and scale by specifying a *standardize-opt*:

**CENTER**

specifies that variables be centered but not scaled. For a variable  $x$ ,

$$s_x = x - \text{center}$$

**CENTERSCALE**

specifies that variables be centered and scaled. This is the default if you do not specify a *standardization-opt*. For a variable  $x$ ,

$$s_x = \frac{x - \text{center}}{\text{scale}}$$

**NONE**

specifies that no standardization be performed.

**SCALE**

specifies that variables be scaled but not centered. For a variable  $x$ ,

$$s_x = \frac{x}{\text{scale}}$$

---

## Spline Effects

This section discusses the construction of spline effects through the experimental EFFECT statement. You can also include spline effects in statistical models by other means. The TRANSREG procedure has dedicated facilities for including regression splines in your model and controlling the construction of the splines. For example, you can use the TRANSREG procedure to fit a spline function but restrict the function to be always increasing or decreasing (monotone). See the section “[Using Splines and Knots](#)” on page 7683 in Chapter 91, “[The TRANSREG Procedure](#),” for more information about using splines with the TRANSREG procedure. The GAM and TPSPLINE procedures also can model the effects of regressor variables in terms of smooth functions that are generated from spline bases. For more information see Chapter 36, “[The GAM Procedure](#),” and Chapter 90, “[The TPSPLINE Procedure](#).”

A spline effect expands variables into spline bases whose form depends on the options that you specify. You can find details about regression splines and spline bases in the section “[Splines and Spline Bases](#)” on page 432. You request a spline effect with the syntax

**EFFECT** *name*=**SPLINE** (*var-list* < / *spline-options*>) ;

The variables in *var-list* must be numeric. Design matrix columns are generated separately for each of these variables, and the set of columns is collectively referred to with the specified name. By default, the spline basis that is generated for each variable is a cubic B-spline basis with three equally spaced knots positioned between the minimum and maximum values of that variable. This yields by default seven design matrix columns for each of the variables in the SPLINE effect.

You can specify the following *spline-options* after a slash (/):

#### **BASIS=BSPLINE**

specifies a B-spline basis for the spline expansion. For splines of degree  $d$  defined with  $n$  knots, this basis consists of  $n + d + 1$  columns. In order to completely specify the B-spline basis,  $d$  left-side boundary knots and  $\max\{d, 1\}$  right-side boundary knots are also required. See the suboptions [KNOTMETHOD=](#), [DATABOUNDARY=](#), [KNOTMIN=](#), and [KNOTMAX=](#) for details about how to specify the positions of both the internal and boundary knots. This is the default if you do not specify the BASIS= suboption.

#### **BASIS=TPF(options)**

specifies a truncated power function basis for the spline expansion. For splines of degree  $d$  defined with  $n$  knots for a variable  $x$ , this basis consists of an intercept, polynomials  $x$ ,  $x^2, \dots, x^d$  and one truncated power function for each of the  $n$  knots. Unlike the B-spline basis, no boundary knots are required. See the suboption [KNOTMETHOD=](#) for details about how you can specify the position of the internal knots.

You can modify the number of columns when you request BASIS=TPF with the following *options*:

#### **NOINT**

excludes the intercept column.

#### **NOPOWERS**

excludes the intercept and polynomial columns.

#### **DATABOUNDARY**

specifies that the extremes of the data be used as boundary knots when building a B-spline basis.

#### **DEGREE= $n$**

specifies the degree of the spline transformation. The degree must be a nonnegative integer. The degree is typically a small integer, such as 0, 1, 2, or 3. The default is DEGREE=3.

#### **DETAILS**

requests tables that show the knot locations and the knots associated with each spline basis function.

#### **KNOTMAX=*value***

specifies that, for each variable in the EFFECT statement, the right-side boundary knots be equally spaced starting at the maximum of the variable and ending at the specified value. This option is ignored for variables whose maximum value is greater than the specified value or if the [DATABOUNDARY](#) option is also specified.

**KNOTMETHOD=knot-method<(knot-options)>**

specifies how to construct the knots for spline effects. You can choose from the following *knot-methods* and affect the knot construction further with the method-specific *knot-options*:

**EQUAL<(n)>**

specifies that  $n$  equally spaced knots be positioned between the extremes of the data. The default is  $n = 3$ . For a B-spline basis, any needed boundary knots continue to be equally spaced unless the DATABOUNDARY option has also been specified. KNOTMETHOD=EQUAL is the default if no *knot-method* is specified.

**LIST(number-list)**

specifies the list of internal knots to be used in forming the spline basis columns. For a B-spline basis, the data extremes are used as boundary knots.

**LISTWITHBOUNDARY(number-list)**

specifies the list of all knots that are used in forming the spline basis columns. When you use a truncated power function basis, this list is interpreted as the list of internal knots. When you use a B-spline basis of degree  $d$ , then the first  $d$  entries are used as left-side boundary knots and the last  $\text{MAX}(d, 1)$  entries in the list are used as right-side boundary knots.

**MULTISCALE<(multiscale-options)>**

specifies that multiple B-spline bases be generated, corresponding to sets with an increasing number of internal knots. As you increase the number of internal knots, the spline basis you generate is able to approximate features of the data at finer scales. So, by generating bases at multiple scales, you facilitate the modeling of both coarse- and fine-grained features of the data. For scale  $i$ , the spline basis corresponds to  $2^i$  equally spaced internal knots. By default, the bases for scales 0–7 are generated. For each scale, a separate spline effect is generated. The name of the constructed spline effect at scale  $i$  is formed by appending  $\_Si$  to the effect name that you specify in the EFFECT statement. If you specify multiple variables in the EFFECT statement, then spline bases are generated separately for each variable at each scale and the name of the corresponding effect is obtained by appending the variable name followed by  $\_Si$  to the name in the EFFECT statement. For example, the following statement generates effects named spl\_x1\_S0, spl\_x1\_S1, spl\_x1\_S2, . . . , spl\_x1\_S7 and spl\_x2\_S1, spl\_x2\_S2, . . . , spl\_x2\_S7:

```
EFFECT spl = spline(x1 x2 / knotmethod=multiscale);
```

The MULTISCALE option is ignored if you specify the BASIS=TPF *spline-option*. The MULTISCALE option is not available for spline effects that are specified in the RANDOM statement of the GLIMMIX procedure.

You can control which scales are included with the following *multiscale-options*:

**STARTSCALE=n**

specifies the start scale, where  $n$  is a positive integer. The default is STARTSCALE=0.

**ENDSCALE=n**

specifies the end scale, where  $n$  is a positive integer. The default is ENDSCALE=7.



**PERCENTILES(*n*)**

requests that internal knots be placed at *n* equally spaced percentiles of the variable or variables named in the EFFECT statement. For example, the following statement positions internal knots at the deciles of the variable *x*. For a B-spline basis, the extremes of the data are used as boundary knots:

```
EFFECT spl = spline(x / knotmethod=percentiles(9));
```

**RANGEFRACTIONS(*fraction-list*)**

requests that internal knots be placed at each fraction of the ranges of the variables in the EFFECT statement. For example, if variable *x1* ranges between 1 and 3, and variable *x2* ranges between 0 and 20, then the following EFFECT statement uses internal knots 1.2, 2, and 2.5 for variable *x1* and internal knots 2, 10, and 15 for variable *x2*:

```
EFFECT spl = spline(x1 x2 / knotmethod=rangefractions(.1 .5 .75));
```

For a B-spline basis, the data extremes are used as boundary knots.

**KNOTMIN=*value***

specifies that for each variable in the EFFECT statement, the left-side boundary knots be equally spaced starting at the specified value and ending at the minimum of the variable. This option is ignored for variables whose minimum value is less than the specified value or if the **DATABOUNDARY** option is also specified.

**SEPARATE**

specifies that when multiple variables are specified in the EFFECT statement, the spline basis for each variable be treated as a separate effect. The names of these separated effects are formed by appending an underscore followed by the name of the variable to the name that you specify in the EFFECT statement. For example, the effect names generated with the following statement are *spl\_x1* and *spl\_x2*:

```
EFFECT spl = spline(x1 x2 / separate);
```

In procedures that support variable selection, such as the GLMSELECT procedure, these two effects can enter or leave the model independently during the selection process. Separated effects are not supported in the RANDOM statement of the GLIMMIX procedure.

**SPLIT**

specifies that each individual column in the design matrix that corresponds to the spline effect be treated as a separate effect that can enter or leave the model independently. Names for these split effects are generated by appending the variable name and an index for each column to the name that you specify in the EFFECT statement. For example, the effects generated for the spline effect in the following statement are *spl\_x1:1*, *spl\_x1:2*, ..., *spl\_x1:7* and *spl\_x2:1*, *spl\_x2:2*, ..., *spl\_x2:7*:

```
EFFECT spl = spline(x1 x2 / split);
```

The SPLIT option is not supported in the GLIMMIX procedure.

## Splines and Spline Bases

This section provides details about the construction of spline bases with the experimental EFFECT statement. A spline function is a piecewise polynomial function in which the individual polynomials have the same degree and connect smoothly at join points whose abscissa values, referred to as knots, are prespecified. You can use spline functions to fit curves to a wide variety of data.

A spline of degree 0 is a step function with steps located at the knots. A spline of degree 1 is a piecewise linear function where the lines connect at the knots. A spline of degree 2 is a piecewise quadratic curve whose values and slopes coincide at the knots. A spline of degree 3 is a piecewise cubic curve whose values, slopes, and curvature coincide at the knots. Visually, a cubic spline is a smooth curve, and it is the most commonly used spline when a smooth fit is desired. Note that when no knots are used, splines of degree  $d$  are simply polynomials of degree  $d$ .

More formally, suppose you specify knots  $k_1 < k_2 < k_3 < \dots < k_n$ . Then a spline of degree  $d \geq 0$  is a function  $S(x)$  with  $d - 1$  continuous derivatives such that

$$S(x) = \begin{cases} P_0(x) & x < k_1 \\ P_i(x) & k_i \leq x < k_{i+1}; i = 1, 2, \dots, n-1 \\ P_n(x) & x \geq k_n \end{cases}$$

where each  $P_i(x)$  is a polynomial of degree  $d$ . The requirement that  $S(x)$  has  $d - 1$  continuous derivatives is satisfied by requiring that the function values and all derivatives up to order  $d - 1$  of the adjacent polynomials at each knot match.

A counting argument yields the number of parameters that define a spline with  $n$  knots. There are  $n + 1$  polynomials of degree  $d$ , giving  $(n + 1)(d + 1)$  coefficients. However, there are  $d$  restrictions at each of the  $n$  knots, so the number of free parameters is  $(n + 1)(d + 1) - nd = n + d + 1$ . In mathematical terminology this says that the dimension of the vector space of splines of degree  $d$  on  $n$  distinct knots is  $n + d + 1$ . If you have  $n + d + 1$  basis vectors, then you can fit a curve to your data by regressing your dependent variable by using this basis for the corresponding design matrix columns. In this context, such a spline is known as a regression spline. The EFFECT statement provides a simple mechanism for obtaining such a basis.

If you remove the restriction that the knots of a spline must be distinct and allow repeated knots, then you can obtain functions with less smoothness and even discontinuities at the repeated knot location. For a spline of degree  $d$  and a repeated knot with multiplicity  $m \leq d$ , the piecewise polynomials that join such a knot are required to have only  $d - m$  matching derivatives. Note that this increases the number of free parameters by  $m - 1$  but also decreases the number of distinct knots by  $m - 1$ . Hence the dimension of the vector space of splines of degree  $d$  with  $n$  knots is still  $n + d + 1$ , provided that any repeated knot has a multiplicity less than or equal to  $d$ .

The EFFECT statement provides support for the commonly used *truncated power function* basis and *B-spline* basis. With exact arithmetic and by using the complete basis, you obtain the same fit with either of these bases. The following sections provide details about constructing spline bases for the space of splines of degree  $d$  with  $n$  knots that satisfies  $k_1 \leq k_2 \leq k_3 < \dots \leq k_n$ .

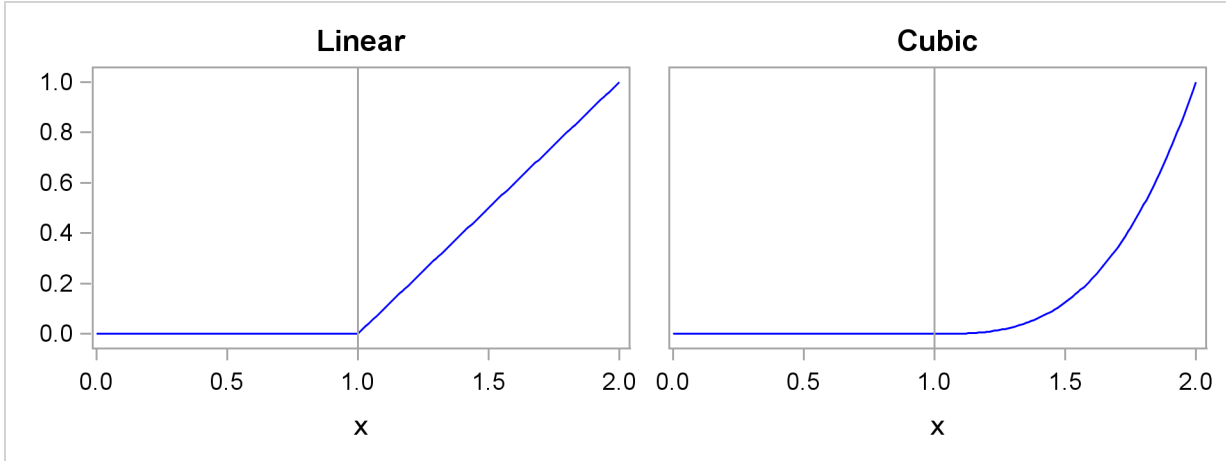
## Truncated Power Function Basis

A truncated power function for a knot  $k_i$  is a function defined by

$$t_i(x) = \begin{cases} 0 & x < k_i \\ (x - k_i)^d & x \geq k_i \end{cases}$$

Figure 19.1 shows such functions for  $d = 1$  and  $d = 3$  with a knot at  $x = 1$ .

**Figure 19.1** Truncated Power Functions with Knot at  $x = 1$



The name is derived from the fact that these functions are shifted power functions that get truncated to zero to the left of the knot. These functions are piecewise polynomial functions with two pieces whose function values and derivatives of all orders up to  $d - 1$  are zero at the defining knot. Hence these functions are splines of degree  $d$ . It is easy to see that these  $n$  functions are linearly independent. However, they do not form a basis, because such a basis requires  $n + d + 1$  functions. The usual way to add  $d + 1$  additional basis functions is to use the polynomials  $1, x, x^2, \dots, x^d$ . These  $d + 1$  functions together with the  $n$  truncated power functions  $t_i(x)$ ,  $i = 1, 2, \dots, n$  form the truncated power basis.

Note that each time a knot is repeated, the associated exponent used in the corresponding basis function is reduced by 1. For example, for splines of degree  $d$  with three repeated knots  $k_i = k_{i+1} = k_{i+2}$  the corresponding basis functions are  $t_i(x) = (x - k_i)_+^d$ ,  $t_{i+1}(x) = (x - k_i)_+^{d-1}$ , and  $t_{i+2}(x) = (x - k_i)_+^{d-2}$ . Provided that the multiplicity of each repeated knot is less than or equal to the degree, this construction continues to yield a basis for the associated space of splines.

The main advantage of the truncated power function basis is the simplicity of its construction and the ease of interpreting the parameters in a model that corresponds to these basis functions. However, there are two weaknesses when you use this basis for regression. These functions grow rapidly without bound as  $x$  increases, resulting in numerical precision problems when the  $x$  data span a wide range. Furthermore, many or even all of these basis functions can be nonzero when evaluated at some  $x$  value, resulting in a design matrix with few zeros that precludes the use of sparse matrix technology to speed up computation. This weakness can be addressed by using a B-spline basis.

## B-Spline Basis

A B-spline basis can be built by starting with a set of Haar basis functions, which are functions that are 1 between adjacent knots and 0 elsewhere, and then applying a simple linear recursion relationship  $d$  times, yielding the  $n + d + 1$  needed basis functions. For the purpose of building the B-spline basis, the  $n$  prespecified knots are referred to as internal knots. This construction requires  $d$  additional knots, known as boundary knots, to be positioned to the left of the internal knots, and  $\text{MAX}(d, 1)$  boundary knots to be positioned to the right of the internal knots. The actual values of these boundary knots can be arbitrary. The EFFECT statement provides several methods for placing the needed boundary knots, including the common method of using repeated values of the data extremes as the boundary knots. The boundary knot placement affects the precise form of the basis functions that are generated, but it does not affect the following two desirable properties:

1. The B-spline basis functions are nonzero over an interval that spans at most  $d + 2$  knots. This yields design matrix columns each of whose rows contain at most  $d + 2$  adjacent nonzero entries.
2. The computation of the basis functions at any  $x$  value is numerically stable and does not require evaluating powers of this value.

The following figures show the B-spline bases defined on  $[0, 1]$  with four equally spaced internal knots at 0.2, 0.4, 0.6, and 0.8.

Figure 19.2 shows a linear B-spline basis. Note that this basis consists of six functions each of which is nonzero over an interval that spans at most three knots.

**Figure 19.2** Linear B-Spline Basis with Four Equally Spaced Interior Knots

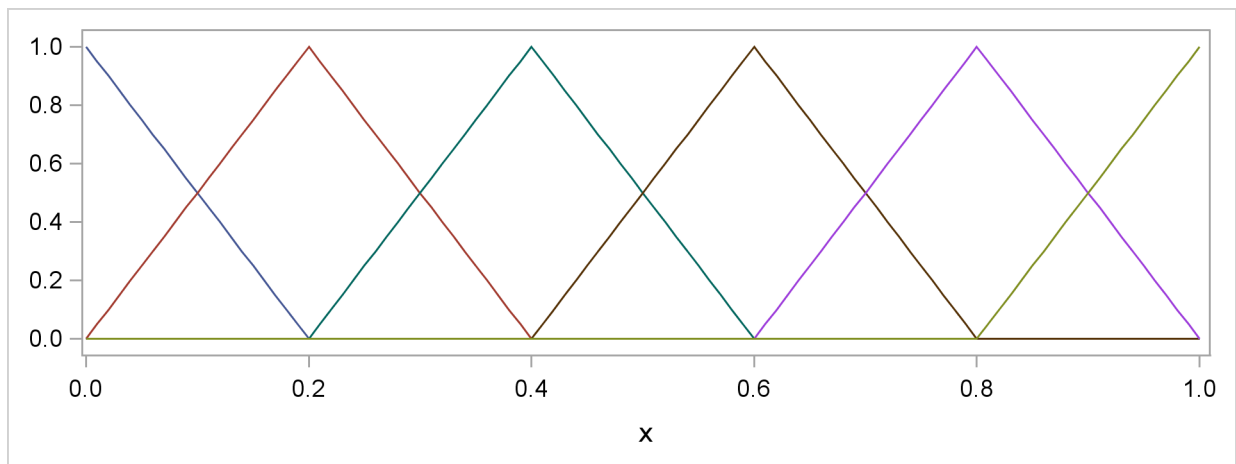


Figure 19.3 shows a cubic B-spline basis where the needed boundary knots are positioned at  $x = 0$  and  $x = 1$ . Note that this basis consists of eight functions, each of which is nonzero over an interval spanning at most five knots.

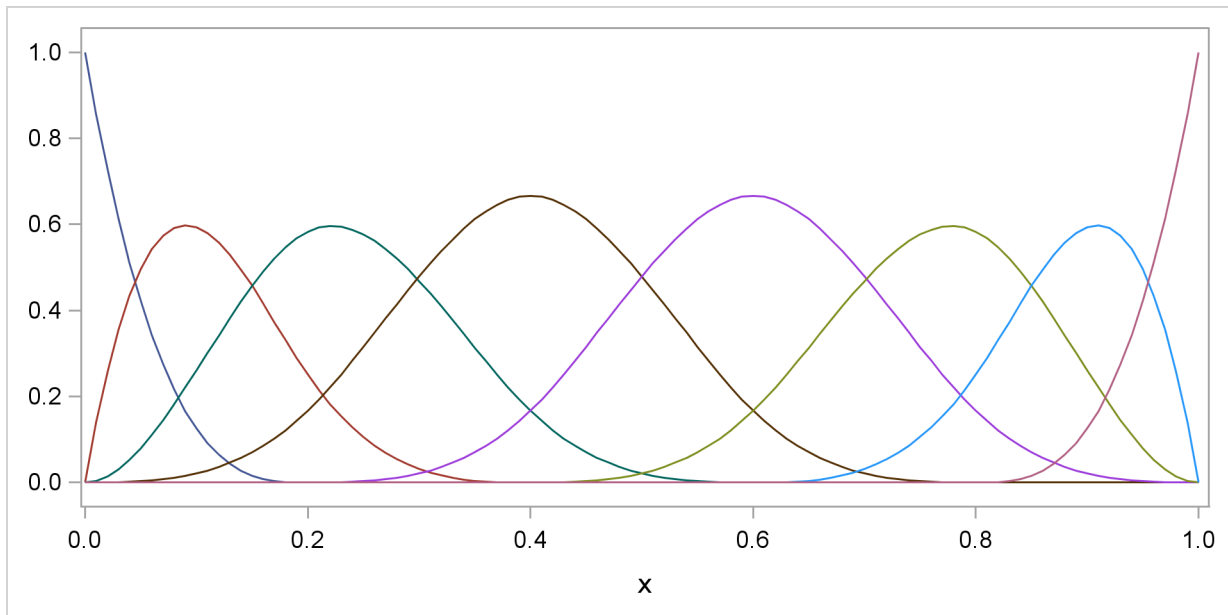
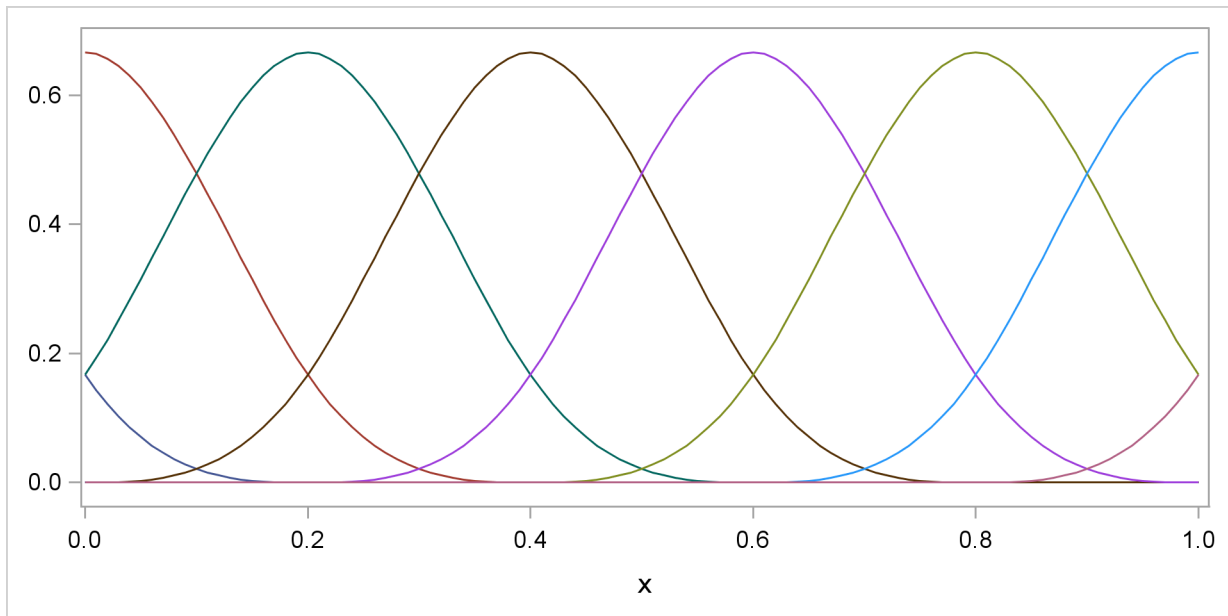
**Figure 19.3** Cubic B-Spline Basis with Four Equally Spaced Interior Knots

Figure 19.4 shows a different cubic B-spline basis where the needed left-side boundary knots are positioned at  $-0.6$ ,  $-0.4$ ,  $-0.2$ , and  $0$ . The right-side boundary knots are positioned at  $1$ ,  $1.2$ ,  $1.4$ , and  $1.6$ . Note that, as in the basis shown in Figure 19.3, this basis consists of eight functions, each of which is nonzero over an interval spanning at most five knots. The different positioning of the boundary knots has merely changed the shape of the individual basis functions.

**Figure 19.4** Cubic B-Spline Basis with Equally Spaced Boundary and Interior Knots

You can find details about this construction in Hastie, Tibshirani, and Friedman (2001).

## EFFECTPLOT Statement

This statement applies to the following procedures:  
GENMOD, LOGISTIC, ORTHOREG, and PLM.

The EFFECTPLOT statement produces a display (*effect plot*) of a complex fitted model and provides options for changing and enhancing the displays. One simple effect plot is the display for a linear regression of the response  $Y$  on a single predictor  $X$ : the regression line is drawn with the predicted response on the  $Y$  axis and the covariate on the  $X$  axis. The regression line can be enhanced by displaying the observations and adding confidence and prediction limits. When your model is more complicated—with more continuous and categorical covariates, nestings and interactions, and link functions—the effect plots display the behavior of some covariates over their ranges while fixing other covariates at some fixed values; this can enable easier interpretation and explanation of the resulting model.

By default, a single plot is produced based on the type of response variable and the number of continuous and classification covariates in the model. You can also specify options to do the following:

- select the variables to display on the plots
- produce multiple plots based on the following: the levels of classification covariates; the minimum, maximum, mean or middle (midrange) value of continuous covariates; and specified values of the covariates
- specify different fixed values for continuous and classification covariates that are not displayed on the plot
- panel and unpanel plots
- select variables to slice or group by
- display (or remove from display) observations and confidence limits

### Syntax: EFFECTPLOT Statement

```
EFFECTPLOT <plot-type> <(plot-definition-options)> >< / options > ;
```

The available *plot-types* and their *plot-definition-options* are described in Table 19.11. Table 19.13 lists the *options* that can be specified after a slash (/) for any *plot-type*, and Table 19.14 lists additional *options* that enhance specific *plot-types*. Full descriptions of the *plot-definition-options* and the other *options* are provided in the section “Dictionary of Options” on page 438.

**Table 19.11** *Plot-Types and Plot-Definition-Options*

Description	Plot-Definition-Options
<b>BOX <i>plot-type</i></b> Displays a box plot of continuous response data at each level of a CLASS effect, with predicted values superimposed and connected by a line. This is an alternative to the <b>INTERACTION</b> <i>plot-type</i> .	<b>PLOTBY=</b> variable or CLASS effect <b>X=</b> CLASS variable or effect
<b>CONTOUR <i>plot-type</i></b> Displays a contour plot of predicted values against two continuous covariates.	<b>PLOTBY=</b> variable or CLASS effect <b>X=</b> continuous variable <b>Y=</b> continuous variable
<b>FIT <i>plot-type</i></b> Displays a curve of predicted values versus a continuous variable.	<b>PLOTBY=</b> variable or CLASS effect <b>X=</b> continuous variable
<b>INTERACTION <i>plot-type</i></b> Displays a plot of predicted values (possibly with error bars) versus the levels of a CLASS effect. The predicted values are connected with lines and can be grouped by the levels of another CLASS effect.	<b>PLOTBY=</b> variable or CLASS effect <b>SLICEBY=</b> variable or CLASS effect <b>X=</b> CLASS variable or effect
<b>SLICEFIT <i>plot-type</i></b> Displays a curve of predicted values versus a continuous variable grouped by the levels of a CLASS effect.	<b>PLOTBY=</b> variable or CLASS effect <b>SLICEBY=</b> variable or CLASS effect <b>X=</b> continuous variable

By default, a single plot is produced based on the type of response variable and the number of continuous and classification covariates in the model as shown in Table 19.12. If you have a polytomous response model, then the response variable is treated as the grouping classification variable in this table. If your model does not fit into Table 19.12, then a default plot is not produced; however, specifying the *plot-type* argument displays a plot with the extra continuous covariates fixed at their mean values and the extra classification covariates fixed at their reference levels.

**Table 19.12** *Default Plot-Types*

Number of Covariates		Type of Response Variable	
Classification	Continuous	Continuous or Binary	Polytomous
1	0	INTERACTION	INTERACTION with groups
2	0	INTERACTION with groups	None
0	1	FIT	SLICEFIT
0	2	CONTOUR	None
1	1	SLICEFIT	None

Table 19.13 and Table 19.14 list the *options* that can be specified after a slash (/) to enhance the effect plots.

**Table 19.13** Available *Options* for All *Plot-Types*

AT<args>	ATLEN=	ATORDER=	ILINK	INDIVIDUAL*
LINK	MOFF	NCOLS=*	NOOBS*	NROWS=*
OBS<(options)>	PLOTBYLEN=	PREDLABEL=	UNPACK	

\* Not available for the BOX *plot-type*

**NOTE:** If your model contains an offset variable and the **MOFF** option is not specified or not valid, then the predicted values are computed only at the observations. In this case, the **FIT** and **SLICEFIT** *plot-types* display scatter plots of the predicted values, the **CONTOUR** *plot-type* displays the residuals against two continuous covariates but with no fitted surface, the **INTERACTION** *plot-type* does not connect the predicted values with lines, and the **BOX** *plot-type* is unchanged.

**Table 19.14** Additional *Options* for Each *Plot-Type*

Plot-Type	Options				
BOX	YRANGE=				
CONTOUR	EXTEND=	GRIDSIZE=			
FIT	ALPHA=	EXTEND=	GRIDSIZE=	NOCLI	
	NOCLM	NOLIMITS	SMOOTH	YRANGE=	
INTERACTION	ALPHA=	CLI	CLM	LIMITS	
	POLYBAR	YRANGE=			
SLICEFIT	ALPHA=	CLI	CLM	EXTEND=	
	GRIDSIZE=	LIMITS	YRANGE=		

## Dictionary of Options

This section describes the **EFFECTPLOT** *options* in alphabetical order.

### **ALPHA=***value*

specifies the significance level,  $0 \leq \text{value} \leq 1$ , for producing  $100(1 - \text{value}/2)\%$  prediction and confidence limits. By default, *value*=0.05.

### **AT** < *contopt* > < *classopt* > < *variable1=varopt* < *variable2=varopt*... > >

where *contopt*= **MEAN** | **MIN** | **MAX** | **MIDRANGE**

*classopt*= **ALL** | **REF**

*varopt*= *contopts* | *number-list* | *classopts* | 'class-level'... 'class-level'

specifies values at which to fix continuous and class variables when they are not used in **X=**, **Y=**, **SLICEBY=**, or **PLOTBY=** effects. The *contopt* keyword fixes continuous variables at their mean, minimum, maximum, or midrange=  $\frac{1}{2}(\text{minimum} + \text{maximum})$ ; the default is to use the mean. The *classopt* keyword either fixes a **CLASS** variable at its reference (last) level



or indicates that all levels of the CLASS variable should be processed; the default is to use the reference level. The *varopt* values enable you to specify *contopt* and *classopt* keywords, or to specify lists of numbers or class levels. You can specify a CLASS variable only once in the AT specification, but you can specify a continuous variable multiple times; for example, the following syntax is valid when X is a continuous variable:

```
effectplot / at(x=min max x=0 to 2 by 1 x=2 5 7);
```

Duplicate AT values are suppressed, so the last X=2 value is ignored.

You can also specify *plug-in values* for CLASS variable levels when computing the predicted values  $x'\beta$ . For example, suppose a CLASS variable A with two levels={0,1} is in the model. Then instead of using the coding for A in the *x* vector by specifying **AT (A=a11)**, **AT (A=ref)** or **AT (A='0' '1')**, you can specify a numeric list to plug in. For example, if the proportion of A's that equal 0 in the data set is 0.3, then you can input the proportions for all levels of the variable by specifying **AT (A=0.3 0.7)**. Under GLM coding, A=0 is coded as "1 0" and A=1 is coded as "0 1", so the plug-in specification replaces both of these codings with "0.3 0.7". Under REFERENCE coding A=0 is coded as "1" and A=1 is coded as "0", so this specification replaces both of these codings with "0.3" followed by "0.7"; however, if another variable is nested within A, then only "0.3" is used. To plug in values, you must specify a multiple of the number of parameters used for the CLASS variable or, if a variable is nested within the CLASS variable, a multiple of the number of levels of the CLASS variable.

The plug-in values are distributed through the rest of the model effects in the following fashion. If a variable is nested within a plug-in variable, then its coding is multiplied by the plug-in value for the level it is nested in. If a variable interacts with a plug-in variable, its coding is multiplied by the appropriate plug-in value for the level it is interacting with. Lag, multimember, polynomial, and spline constructed effects are affected only by interactions and nestings. If the plug-in variable is part of a collection effect, then its values are replaced by the plug-in values; collection effects are also affected by interactions and nestings.

The AT levels are used for computing the predicted values. If the **OBS** option is also specified, then all observations are still displayed on all of the plots. For example, if you specify the options **AT (A='1') OBS**, then the fitted values are computed with A=1, but all of the observations are displayed with their predicted values computed at their observed level of A. If you want to display only a subset of the observations based on the levels of a CLASS variable, then you must specify either the **PLOTBY=** option or the **OBS(BYAT)** option.

#### **ATLEN=*n***

specifies the maximum length ( $1 \leq n \leq 256$ ) of the levels of the **AT** variables that are displayed in footnotes and headers. By default, up to 256 characters of the CLASS levels are displayed, and the continuous AT levels are displayed with a BEST format that has a width greater than or equal to 5, which distinguishes each level. **CAUTION:** If the levels of your **AT** variables are not unique when the first *n* characters are displayed, then the levels are combined in the plots but not in the underlying computations. Also, at most *n* characters for continuous **AT** variables are displayed.

#### **ATORDER=ASCENDING | DESCENDING**

uses the AT values for continuous variables in ascending or descending order as specified. By default, values are used in the order of their first appearance in the **AT** option.

**CLI**

displays normal (Wald) prediction limits. This option is available only for normal distributions with identity links. If your model is from a Bayesian analysis, then sampling-based intervals are computed; see the section “[Analysis Based on Posterior Estimates](#)” on page 5434 in Chapter 66, “[The PLM Procedure](#),” for more information.

**CLM**

displays confidence limits. These are computed as the normal (Wald) confidence limits for the linear predictor, and if the **ILINK** option is specified, the limits are also back-transformed by the inverse link function. If your model is from a Bayesian analysis, then sampling-based intervals are computed; see the section “[Analysis Based on Posterior Estimates](#)” on page 5434 in Chapter 66, “[The PLM Procedure](#),” for more information.

**EXTEND=DATA | value**

extends continuous covariate axes by  $value \times \frac{1}{2}range$  in both directions, where *range* is the range of the X axis. Specifying the **DATA** keyword displays curves to the range of the data within the appropriate **SLICEBY=**, **PLOTBY=**, and **AT** level. For the **CONTOUR** *plot-type*, *value*=0.05 by default; other *plot-types* set the default value to 0. When constructed effects are present, only the **EXTEND=DATA** option is available.

**GRIDSIZE=n**

specifies the resolution of curves by computing the predicted values at *n* equally spaced x-values and specifies the resolution of surfaces by computing the predicted values on an  $n \times n$  grid of points. Default values are  $n=200$  for curves and bands,  $n=50$  for surfaces, and  $n=2$  for lines. If results of a Bayesian or bootstrap analysis are being displayed, then the defaults are  $n=500000/B$ , where *B* is the number of samples, the upper limit is equal to the usual defaults, and the lower limit equal to 20.

**ILINK**

displays the fit on the scale of the inverse link function. In particular, the results are displayed on the probability scale for logistic regression. By default, a procedure displays the fit on either the [link](#) or inverse link scale.

**INDIVIDUAL**

displays individual probabilities for polytomous response models with cumulative links on the scale of the inverse link function. This option is not available when the **LINK** option is specified, and confidence limits are not available with this option.

**LIMITS**

invokes the [CLI](#) and [CLM](#) options.

**LINK**

displays the fit on the scale of the link function; that is, the linear predictor. Note that probabilities or observed proportions near 0 and 1 are transformed to  $\pm 20$ . By default, a procedure displays the fit on either the link or [inverse link](#) scale.

**MOFF**

moves the offset for a Poisson regression model to the response side of the equation. If the **ILINK** option is also in effect, then the rate is displayed on the Y axis, while the **LINK** option displays the log of the rate on the Y axis. Without this option, the predicted values are computed and displayed only for the observations.

**NCOLS=*n***

specifies the maximum number of columns in a paneled plot. This option is not available with the **BOX** *plot-type*.

The default choice of **NROWS=** and **NCOLS=** is based on the number of **PLOTBY=** and **AT** levels. If there is only one plot being displayed in a panel, then **NROWS=1** and **NCOLS=1** and the plots are produced as if you specified only the **UNPACK** option. If only two plots are displayed in a panel, then **NROWS=1** and **NCOLS=2**. For all other cases, a 2x2, 2x3, or 3x3 panel is chosen based on how much of the last panel is used, with ties going to the larger panels. For example, if 14 plots are being created, then this requires either four 2x2 panels with 50% of the last panel filled, three 2x3 panels with 33% of the last panel filled, or two 3x3 panels with 55% of the last panel filled; in this case, the 3x3 panels are chosen.

If you specify both of the **NROWS=** and **NCOLS=** options, then those are the values used. However, if you only specify one of the options but have fewer plots, then the panel size is reduced; for example, if you specify **NROWS=6** but only have four plots, then a plot with four rows and one column is produced.

**NOCLI**

suppresses the prediction limits.

**NOCLM**

suppresses the confidence limits.

**NOLIMITS**

invokes the **NOCLI** and **NOCLM** options.

**NOOBS**

suppresses the display of observations and overrides the specification of the **OBS=** option.

**NROWS=*n***

specifies the maximum number of rows in a paneled plot. This option is not available with the **BOX** *plot-type*. See the **NCOLS=** option for more details.

**OBS<(options)>**

displays observations on the effect plots. An input data set is required; hence the **OBS** option is not available with PROC PLM. The **OBS** option is overridden by the **NOOBS** option. When the **ILINK** option is specified with binary response variables, then either the observed proportions or a coded value of the response is displayed. For polytomous response variables, the observed values are overlaid onto the fitted curves unless the **LOCATION=** option is specified. Whether observations are displayed by default or not depends upon the procedure. If the **PLOTBY=** option is specified, then the observations displayed on each plot are from the corresponding **PLOTBY=** level for classification effects; for continuous effects, all observations are displayed on every plot.

The following *options* are available:

**BYAT** subsets the observations by **AT** level and by the **PLOTBY=** level. If you specify the **PLOTBY=** option without specifying this option, the observations are displayed on the plots that correspond to their **PLOTBY=** level without regard to any classification variables specified in the **AT** option. However, for **FIT** *plot-types* a distance can be computed and displayed (see the **DISTANCE** option for more information). This option is ignored when there are no **AT** variables.

**CDISPLAY=NONE | OUTLINE | GRADIENT | OUTLINEGRADIENT** controls the display of observations on contour plots. The keyword **OUTLINE** displays the observations as circles, **GRADIENT** displays gradient-colored dots, **OUTLINEGRADIENT** displays gradient-filled-circles, and **NONE** suppresses the display of the observations. The default is **CDISPLAY=OUTLINEGRADIENT**.

**CGRADIENT=RESIDUAL | DEPENDENT** specifies what the gradient-shading of the observed values on the **CONTOUR** *plot-type* represents. The **RESIDUAL** keyword shades the observations by the raw residual value and displays the fitted surface as a line contour plot. The **DEPENDENT** keyword shades the observations by the response variable value and displays the fitted surface as a contour shaded on the same scale. The default is **CGRADIENT=DEPENDENT**.

**DEPTH=depth** specifies the number of overlapping observations that can be distinguished by adjusting their transparency; you can specify  $1 \leq \text{depth} \leq 100$ . By default, **DEPTH=1**. The **DEPTH=** option is available with **FIT**, **SLICEFIT**, and **INTERACTION** *plot-types*.

**DISTANCE** displays observations on **FIT** *plot-types* with a color-gradient that indicates how far the observation is from the **AT** and **PLOTBY=** level. This option is ignored unless an **AT** or **PLOTBY=** option is specified.

The distance is computed as the square root of the following number: for each continuous **AT** and **PLOTBY=** variable, add the square of the difference from the observed value divided by the range of the variable; for each **CLASS AT** and **PLOTBY=** variable, add 1 if the **CLASS** levels are different. Thus the largest possible distance is the square root of the number of **AT** and **PLOTBY=** variables. Observations at zero distance are displayed with the darkest color, and the color fades as the distance increases.

Note that the **UNPACKED** panels compute the maximum distance within each panel and hence do not use the same gradient across all panels. Also, the **PANELS** *panel-type* computes the maximum distance within each **PLOTBY=** level, so a different gradient is used for each **PLOTBY=** level. All other *panel-types* compute the maximum distance across all observations and therefore use the same gradient on every plot.

**FITATCLASS** computes fitted values only for class levels that are observed in the data set. This option is ignored when the GLM parameterization is used.

**FRINGE** displays observations in a fringe (rug) plot at the bottom of the plot. This option is available only with **FIT** and **SLICEFIT** *plot-types*.

**JITTER**<(options)> shifts (*jitters*) the observations. By default, the jittering in the X direction is achieved by adding a random number that is generated according to a normal distribution with mean=0 and standard deviation=  $jitter/2$  and truncating at  $\pm jitter$ , where  $jitter=0.01$  times the range of the X axis; the jittering in the Y direction is performed independently but in the same fashion. The JITTER option is not available with the **BOX** plot-type. The following *options* are available:

**FACTOR**=*factor* sets the jitter to *factor* times the range of the axis, and jitters in both the X and Y directions. You can specify  $0 \leq factor \leq 1$ .

**SEED**=*seed* specifies an integer to use as the initial seed for the random number generator. If you do not specify a seed, or if you specify a value less than or equal to zero, then the time of day from the computer clock is used to generate an initial seed.

**X**=*x-jitter* sets the jitter to *x-jitter* for the X direction; the jitter in the Y direction is assumed to be 0 unless the Y= option is also specified. You can specify  $x-jitter \geq 0$ . The X= option is not available for the **INTERACTION** plot-type. This option is ignored if the **FACTOR**= option is also specified.

**Y**=*y-jitter* sets the jitter to *y-jitter* for the Y direction; the jitter in the X direction is assumed to be 0 unless the X= option is also specified. You can specify  $y-jitter \geq 0$ . This option is ignored if the **FACTOR**= option is also specified.

**LABEL**<=**OBS**> labels markers with their observation number.

**LOCATION**=*location* specifies where the observed values for polytomous response models are displayed when the **SLICEBY**= variable is the response. This option is available only with the **SLICEFIT** and **INTERACTION** plot-types. The observations are always displayed at their appropriate X-axis value, but their Y-axis location can depend on the specification of the **YRANGE**= option or on the minimum and maximum computed predicted values in addition to the specified *location*. The following *locations* are available:

**BOTTOM**<=*factor*> displays the first response level at the minimum predicted value, and displays succeeding response levels above the first level at  $factor \times range$  intervals, where *range* is the range of the predicted values. You can specify  $0 \leq factor \leq 1$ , but the largest usable value, which corresponds to **LOCATION**=**SPREAD**, is  $factor = \frac{1}{k}$ , where  $k + 1$  is the number of response levels that are displayed. By default,  $factor = 0.03$ .

**CURVE** displays the observations for polytomous response models at their predicted values. For displays on the **LINK** scale, the reference level is displayed at the maximum value. This method is the default.

**FIRST** displays the observations for a response level at the first displayed predicted value for that response level.

**MAX** displays the observations for a response level at the maximum displayed predicted value for that response level.

**MIDDLE** displays the observations for a response level at the middle of the displayed predicted values for that response level.

- MIN** displays the observations for a response level at the minimum displayed predicted value for that response level.
- SPREAD** displays the observations with the response levels evenly spread across the Y axis.
- TOP<=*factor*>** displays the last response level at the maximum predicted value, and displays preceding response levels below the last level at *factor* × *range* intervals, where *range* is the range of the predicted values. You can specify  $0 \leq \textit{factor} \leq 1$ , but the largest usable value, which corresponds to **LOCATION=SPREAD**, is  $\textit{factor} = \frac{1}{k}$ , where  $k + 1$  is the number of response levels that are displayed. By default, *factor* = 0.03.

**PLOTBY< (*panel-type*)>=*effect*<=*numeric-list*>**

specifies a variable or CLASS effect at whose levels the predicted values are computed and the plots are displayed. You can specify the response variable as the *effect* for polytomous response models. The *panel-type* argument specifies the method in which the plots are grouped for the display. The following *panel-types* are available.

- COLUMNS** specifies that the columns within each panel correspond to different levels of the PLOTBY= effect and hence the rows correspond to different **AT** levels.
- PACK** specifies that plots be displayed in the panels as they are produced with no control over the placement of the PLOTBY= and **AT** levels.
- PANELS | LEVELS** specifies that each level of the PLOTBY= effect begin a new panel of plots and the **AT** levels define the plots within the panels.
- ROWS** specifies that the rows within each panel correspond to different levels of the PLOTBY= effect and hence the columns correspond to different **AT** levels.

This option is ignored with the **BOX** *plot-type*; box plots are always displayed in an unpacked fashion, grouped by the PLOTBY= and **AT** levels. If you specify a continuous variable as the *effect*, then you can either specify a *numeric-list* of values at which to display that variable or, by default, five equally spaced values from the minimum variable value to its maximum are displayed.

The default *panel-type* is based on the number of PLOTBY= and **AT** levels as shown in the following table.

Number of PLOTBY Levels	Number of AT Levels	Resulting <i>panel-type</i>
1	1	(UNPACK)
>1	1	PACK
1	>1	PACK
2	>1	ROWS
3	>1	COLUMNS
>3	>1	PANELS

The default dimensions of the panels are also based on the number of PLOTBY= and **AT** levels; see the **NCOLS=** option for details.



Specification of the *panel-type* is honored except in the following cases. If you specify a *panel-type* but produce only one plot, specify the **NROWS=1** and **NCOLS=1** options, or specify the **UNPACK** option, then the plots are produced as if you specified only the **UNPACK** option. If you specify the **PANELS** *panel-type* with only one **AT** level, then the plots are produced with the **UNPACK** option. However, if you specify the **PANELS** *panel-type* but the **PLOTBY=** effect has only one level, then the *panel-type* is changed to **PACK**.

#### **PLOTBYLEN=*n***

specifies the maximum length ( $1 \leq n \leq 256$ ) of the levels of the **PLOTBY=** variables, which are displayed in footnotes and headers. By default, up to 256 characters of the **CLASS** levels are displayed. **CAUTION:** If the levels of your **PLOTBY=** variables are not unique when the first *n* characters are displayed, then the levels are combined in the plots but not in the underlying computations.

#### **POLYBAR**

displays polytomous response data as a stacked histogram with bar heights defined by the individual predicted value. Your response variable must be the **SLICEBY=** variable, and the **INDIVIDUAL** and **ILINK** options must be in effect; otherwise, the option is ignored. Confidence limits are ignored.

#### **PREDLABEL='label'**

specifies a label to be displayed on the Y axis. The default Y axis label is determined by your model. For the **CONTOUR** *plot-type*, this option changes the title to “*label* for Y.”

#### **SHOWCLEGEND**

displays the gradient-legend for the **CONTOUR** *plot-type*. This option has no effect when the **OBS(CGRADIENT=RESIDUAL)** option is also specified.

#### **SLICEBY=NONE** | *effect* <=*numeric-list*>

displays the fitted values at the different levels of the specified variable or **CLASS** effect. You can specify the response variable as the *effect* for polytomous response models. Use this option to modify **SLICEFIT** and **INTERACTION** *plot-types*. If you specify a continuous variable as the *effect*, then you can either specify a *numeric-list* of values at which to display that variable or, by default, five equally spaced values from the minimum variable value to its maximum are displayed. The **NONE** keyword is available for preventing the **INTERACTION** *plot-type* from slicing by a second class covariate. Note that the **SLICEBY=NONE** option is not available for the **SLICEFIT** *plot-type*, since that is the same as the **FIT** *plot-type*.

#### **SMOOTH**

overlays a loess smooth on the **FIT** *plot-type* for models that have only one continuous predictor. This option is not available for binary or polytomous response models.

#### **UNPACK**

suppresses paneling. By default, multiple plots can appear in some output *panels*. Specify **UNPACK** to display each plot separately.

#### **X=effect**

specifies values to display on the X axis. For **BOX** and **INTERACTION** *plot-types*, *effect* can be a **CLASS** effect in the **MODEL** statement. For **FIT**, **SLICEFIT**, and **CONTOUR** *plot-types*, *effect* can be any continuous variable in the model.

**Y=args**

specifies values to display on the Y axis for the **CONTOUR** *plot-type*. The Y= argument can be any continuous variable in the model.

**YRANGE=CLIP | (< min>< ,max>)**

displays the predicted values on the Y axis in the range [*min*,*max*]. The YRANGE=CLIP option has the same effect as specifying the minimum predicted value as *min* and the maximum predicted value as *max*. The axis might extend beyond your specified values. By default, when the Y axis displays predicted probabilities, the entire Y axis, [0,1], is displayed. This option is useful if your predicted probabilities are all contained in some subset of this range. This option is not available with the **CONTOUR** *plot-type*.

---

## ODS Graphics: EFFECTPLOT Statement

To produce the EFFECTPLOT displays, you must also specify the ODS GRAPHICS ON statement. For more information about the ODS GRAPHICS statement, see Chapter 21, “Statistical Graphics Using ODS.” The available graph names are provided in Table 19.15.

**Table 19.15** ODS Graphics Produced by the **EFFECTPLOT** Statement

ODS Graph Name	Plot Description
BoxFitPlot	A box plot of the responses at each level of one classification effect, overlaid with a plot of the predicted values
ContourFitPlot	A contour plot of the fitted surface against two continuous covariates
ContourFitPanel	A panel of ContourFitPlots
FitPlot	A curve of the predicted values plotted against one continuous covariate
FitPanel	A panel of FitPlots
InteractionPlot	A plot of the predicted values (connected by a line) against one classification effect, possibly for each level of a second classification effect
InteractionPanel	A panel of InteractionPlots
SliceFitPlot	A curve of the predicted values against one continuous covariate for each level of a second classification covariate
SliceFitPanel	A panel of SliceFitPlots



## Examples: EFFECTPLOT Statement

### Example 19.1: A Saddle Surface

Myers (1976) analyzes an experiment reported by Frankel (1961) which is aimed at maximizing the yield of mercaptobenzothiazole (MBT) by varying processing time and temperature. Myers uses a two-factor model in which the estimated surface does not have a unique optimum. The objective is to find the settings of time and temperature in the processing of a chemical that maximize the yield. The following statements create the data set d:

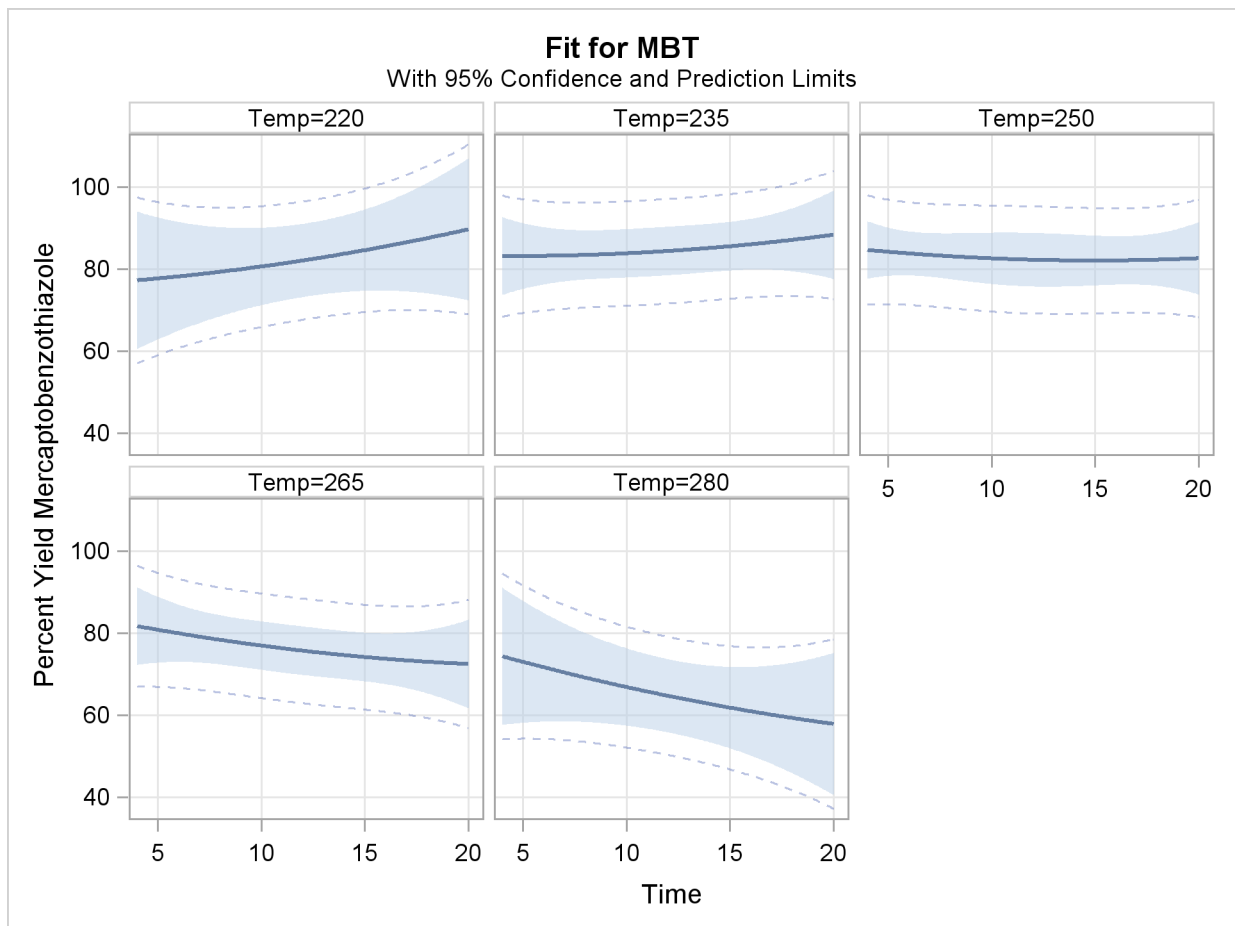
```
data d;
  input Time Temp MBT @@;
  label Time = "Reaction Time (Hours)"
        Temp = "Temperature (Degrees Centigrade)"
        MBT = "Percent Yield Mercaptobenzothiazole";
  datalines;
    4.0  250  83.8    20.0  250  81.7    12.0  250  82.4
    12.0  250  82.9    12.0  220  84.7    12.0  280  57.9
    12.0  250  81.2     6.3  229  81.3     6.3  271  83.1
    17.7  229  85.3    17.7  271  72.7     4.0  250  82.0
  ;
```

In the following statements, the ORTHOREG procedure fits a response surface regression model to the data and uses the EFFECTPLOT statement to create a slice of the response surface. The *FIT plot-type* requests plots of the predicted yield against the Time variable, and the **PLOTBY=** option specifies that the Temp variable is fixed at five equally spaced values so that five fitted regression curves are displayed in [Output 19.1.1](#).

```
ods graphics on;
proc orthoreg data=d;
  model MBT=Time|Time|Temp|Temp@2;
  effectplot fit(x=time plotby=temp);
run;
ods graphics off;
```

The displays in [Output 19.1.1](#) show that the slope of the surface changes as the temperature increases.

**Output 19.1.1** Panel of Fit Plots

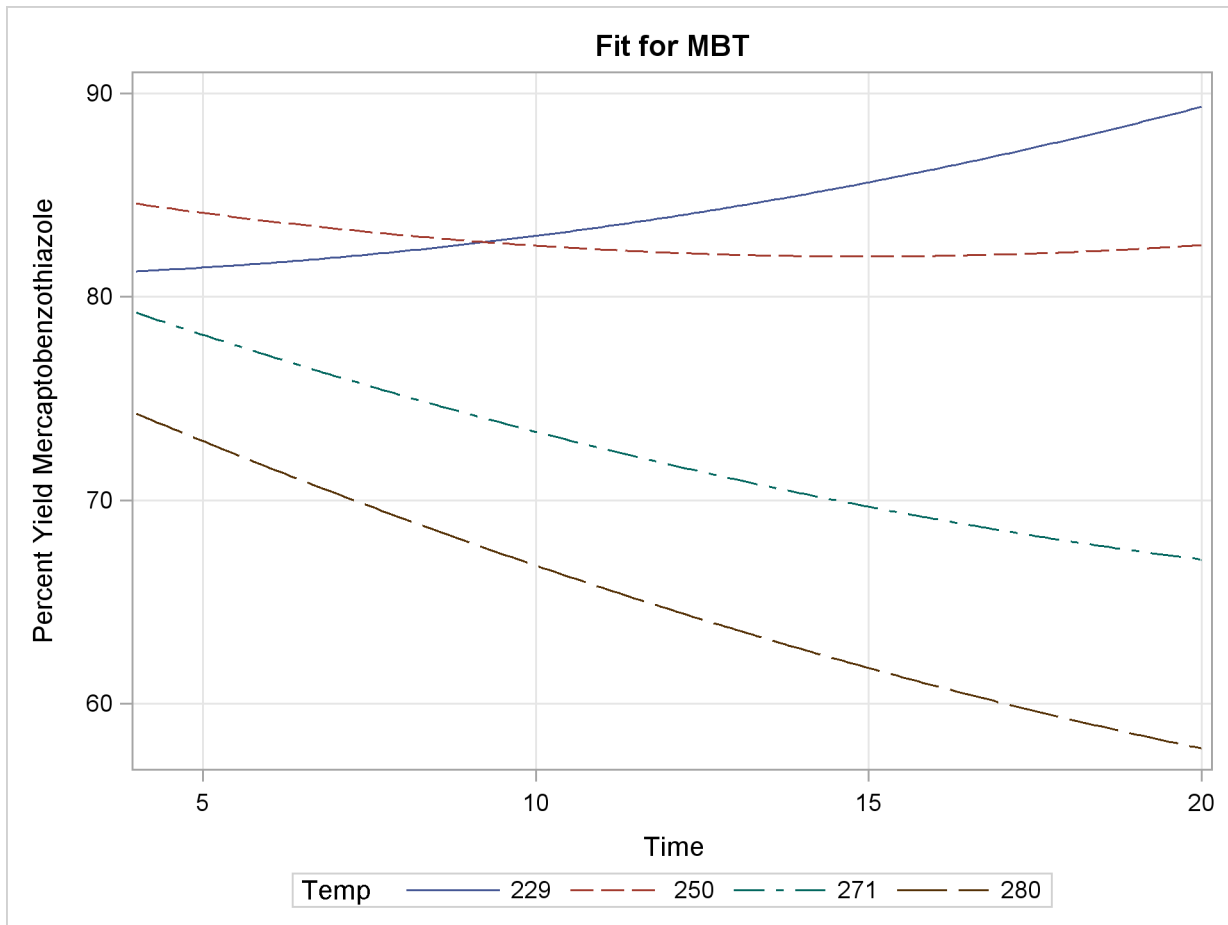


It might be more informative to see these results in one graphic, so the following statements specify the `SLICEFIT` *plot-type* to overlay plots of the predicted yield versus time, fixed at several values of temperature. In this case, the `SLICEBY=` option is specified to explicitly use the same four temperatures as used in the experiment.

```
ods graphics on;
proc orthoreg data=d;
  model MBT=Time|Time|Temp|Temp@2;
  effectplot slicefit(x=time sliceby=temp=229 250 271 280);
run;
ods graphics off;
```

Output 19.1.2 shows that you should choose either low temperatures and long times to optimize the yield, or maybe high temperatures and short times.

**Output 19.1.2** Fit Plot Grouped (Sliced) by Temp

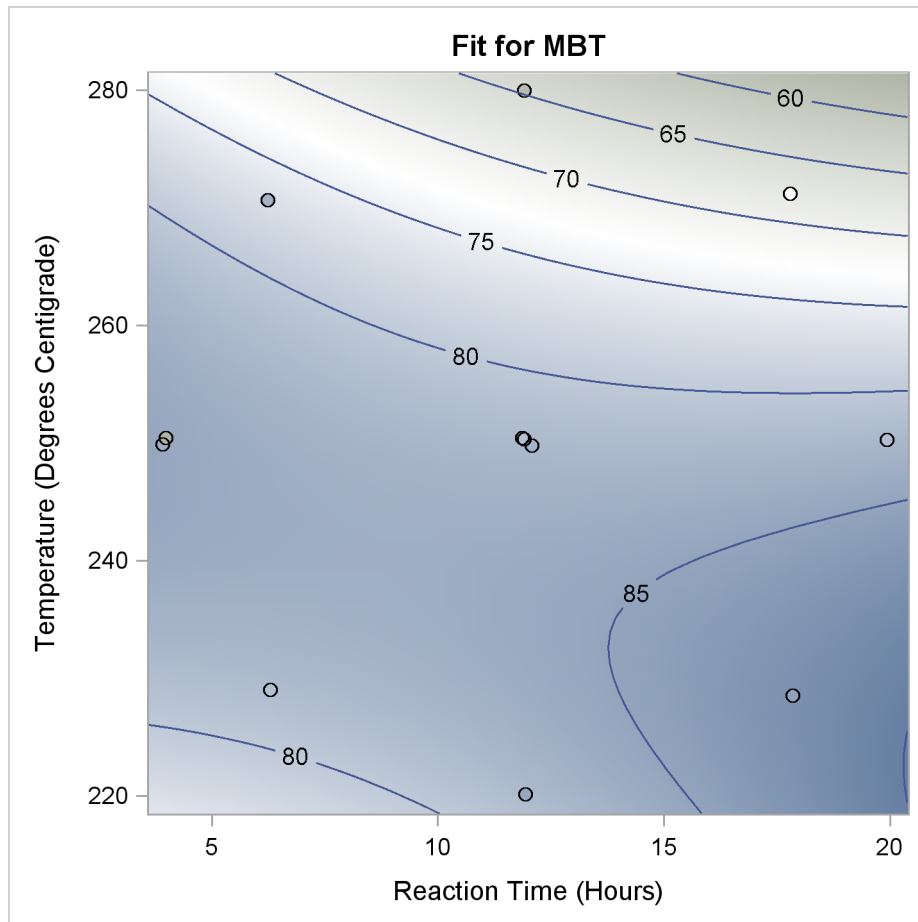


Another plot might explain the reason for this more clearly. The following statements produces the default EFFECTPLOT statement display, enhanced by the **OBS(JITTER)** option to jitter the observations so that you can see the replicated points.

```
ods graphics on;
proc orthoreg data=d;
  model MBT=Time|Time|Temp|Temp@2;
  effectplot / obs(jitter);
run;
ods graphics off;
```

Output 19.1.3 shows the reason for the changing slopes is that the surface is at a saddle point. This surface does not have an optimum point.

**Output 19.1.3** Contour Fit Plot with Jittered Observations



**Example 19.2: Unbalanced Two-Way ANOVA**

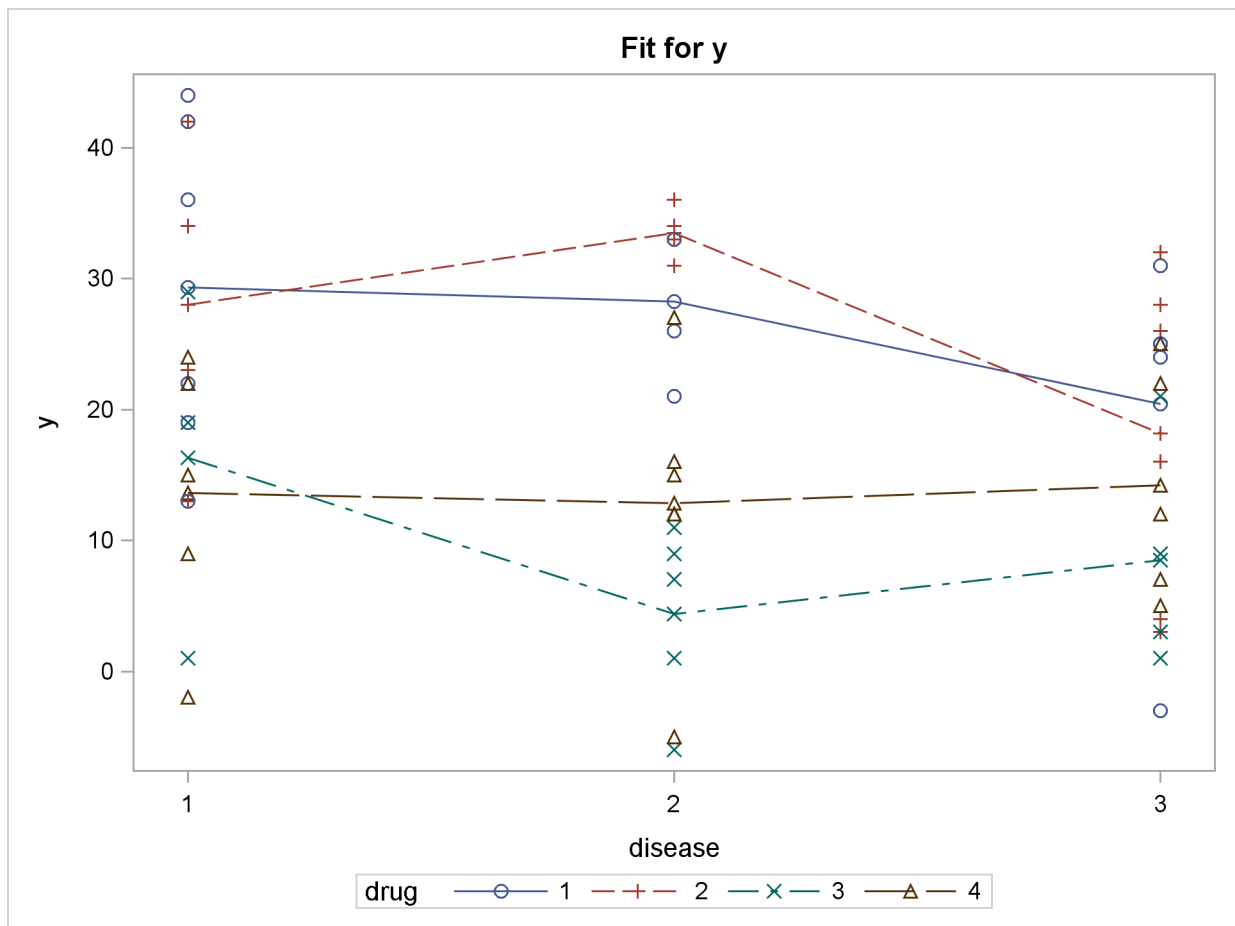
This example uses data from Kutner (1974, p. 98) to illustrate a two-way analysis of variance. The original data source is Afifi and Azen (1972, p. 166). The following statements create the data set a:

```
data a;
  input drug disease @;
  do i=1 to 6;
    input y @;
    output;
  end;
  datalines;
1 1 42 44 36 13 19 22
1 2 33 . 26 . 33 21
1 3 31 -3 . 25 25 24
2 1 28 . 23 34 42 13
2 2 . 34 33 31 . 36
2 3 3 26 28 32 4 16
3 1 . . 1 29 . 19
3 2 . 11 9 7 1 -6
3 3 21 1 . 9 3 .
4 1 24 . 9 22 -2 15
4 2 27 12 12 -5 16 15
4 3 22 7 25 5 12 .
;
```

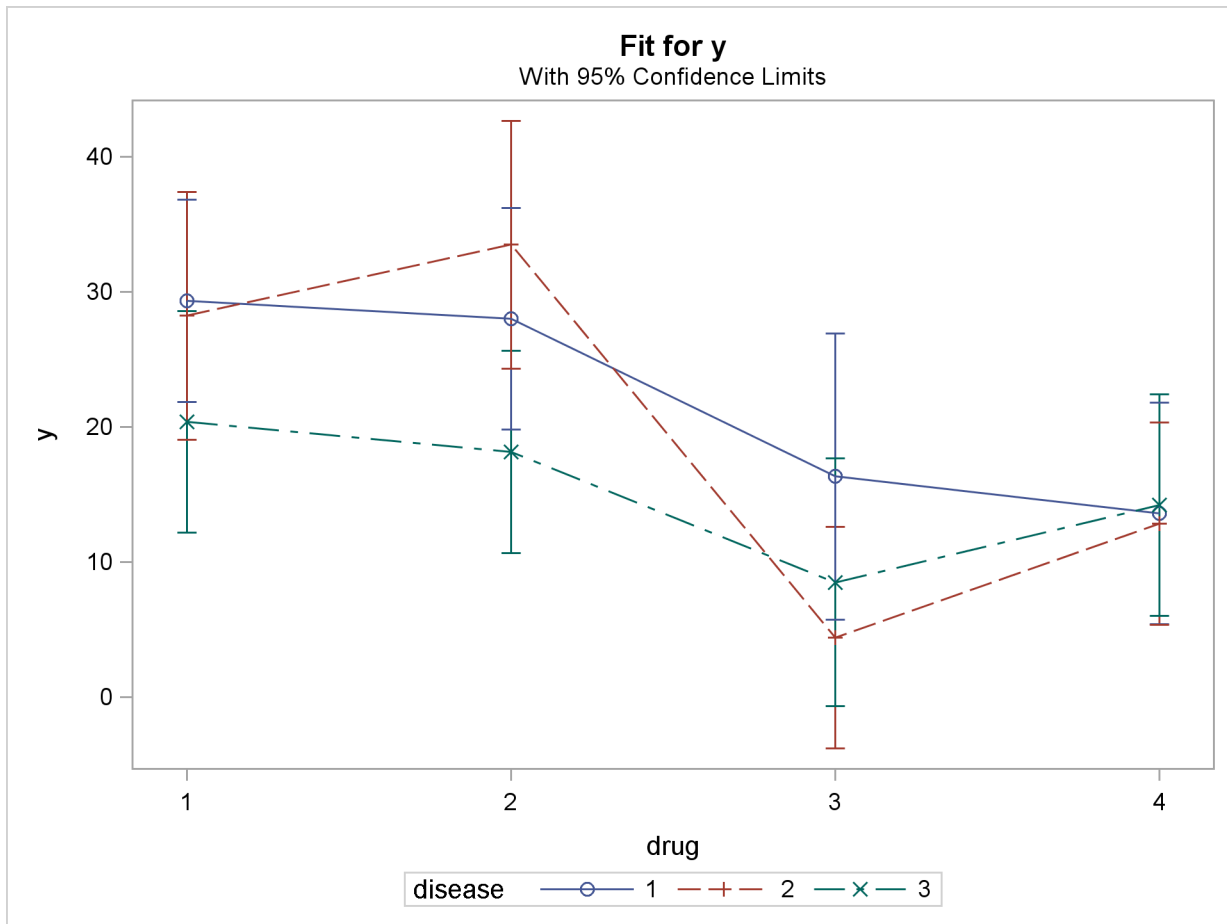
In the following statements, PROC GENMOD fits two classification variables and their interaction to Y. The first EFFECTPLOT statement displays the default graphic, which plots the predicted values against Disease for each of the four Drug levels. The **OBS** option also displays the observations on the plot. The second EFFECTPLOT statement modifies the default to plot the predicted values against Drug for each of the four Disease levels. The **CLM** option is specified to produce 95% confidence bars for the means.

```
ods graphics on;
proc genmod data=a;
  class drug disease;
  model y=disease drug disease*drug / d=n;
  effectplot / obs;
  effectplot interaction(sliceby=disease) / clm;
run;
ods graphics off;
```

In **Output 19.2.1**, the default interaction plot is produced, and the observations are also displayed. From this plot, you can compare the performance of the drugs for a given disease. The predicted values are connected with a line to provide something for your eye to follow—obviously a line has no intrinsic meaning in this graphic. Drugs 3 and 4 are consistently outperformed by the first two drugs.

**Output 19.2.1** Interaction Plot: Default with Observations

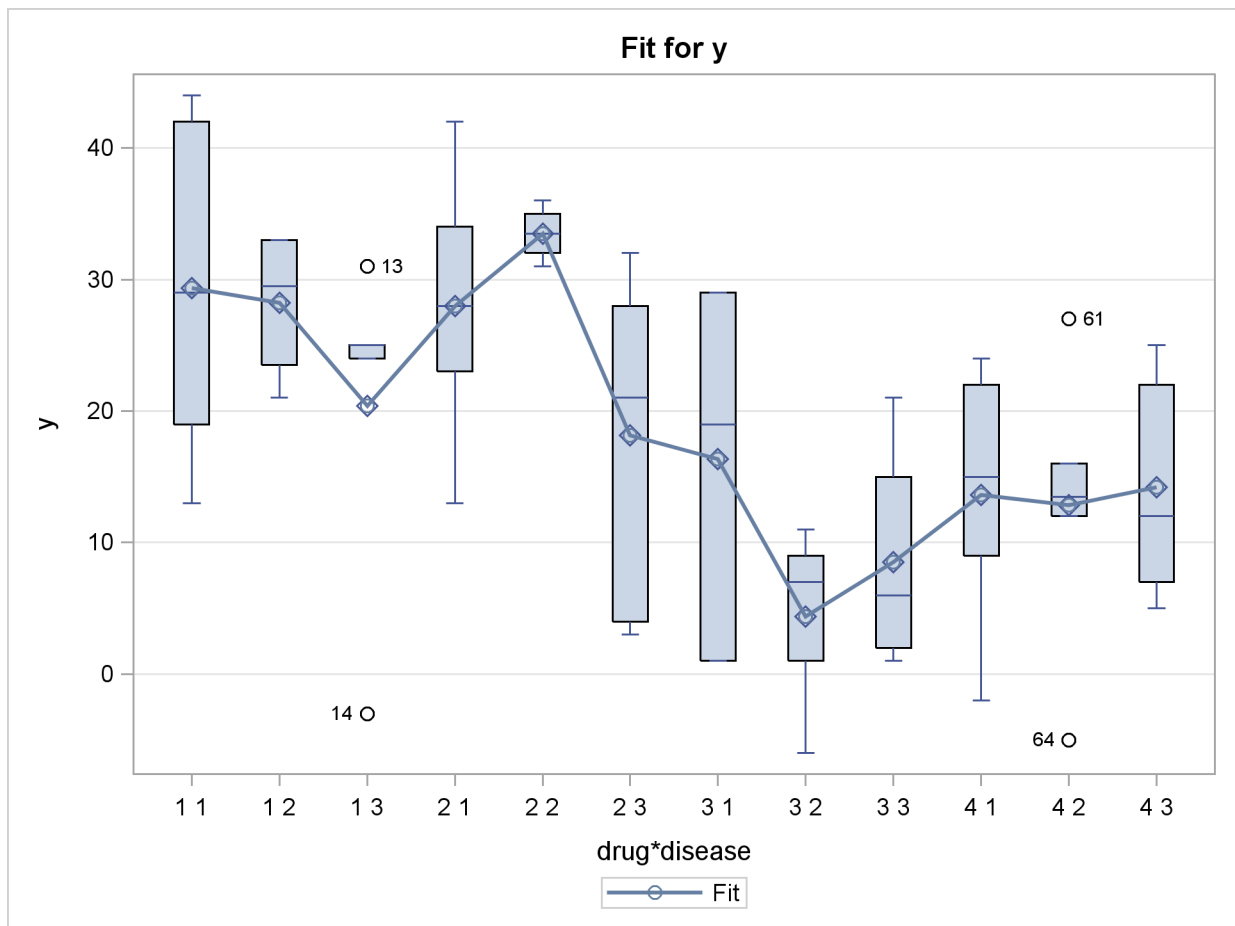
By default, the first classification variable is displayed on the X axis and the second classification variable is used for grouping. Specifying the `SLICEBY=DISEASE` option in the second `EFFECTPLOT` statement reverses this, displays the classification variable with the most levels on the X axis, and slices by fewer levels, resulting in a more readable display. [Output 19.2.2](#) shows how well a given drug performs on each disease.

**Output 19.2.2** Interaction Plot with Specified SLICEBY= Effect

In the following statements, the **BOX** *plot-type* is requested to display box plots of the predictions by each drug and disease combination. The second **EFFECTPLOT** statement displays the same information by using an **INTERACTION** *plot-type* and specifies the **OBS** option to display the individual observations. The third **EFFECTPLOT** statement creates an interaction plot of predictions versus drug for each of the Disease levels, and displays them in a panel.

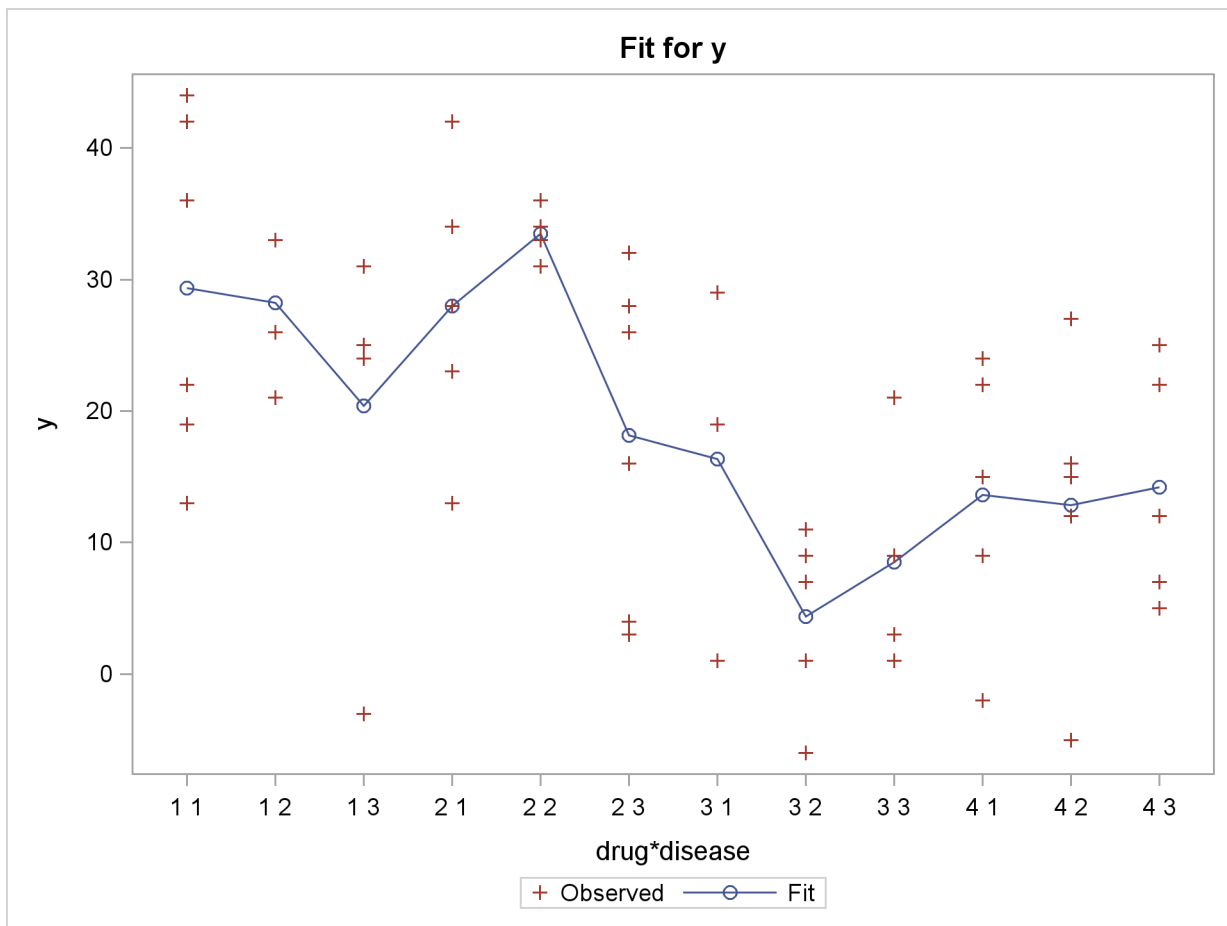
```
ods graphics on;
proc genmod data=a;
  class drug disease;
  model y=drug disease drug*disease / d=n;
  effectplot box;
  effectplot interaction(x=drug*disease) / obs;
  effectplot interaction(plotby=disease);
run;
ods graphics off;
```

In the box plot in [Output 19.2.3](#), the predicted values are displayed as circles; they coincide with the mean of the data at each level which are displayed as diamonds. The predicted values are again connected by lines. It is difficult to make any conclusions from this graphic.

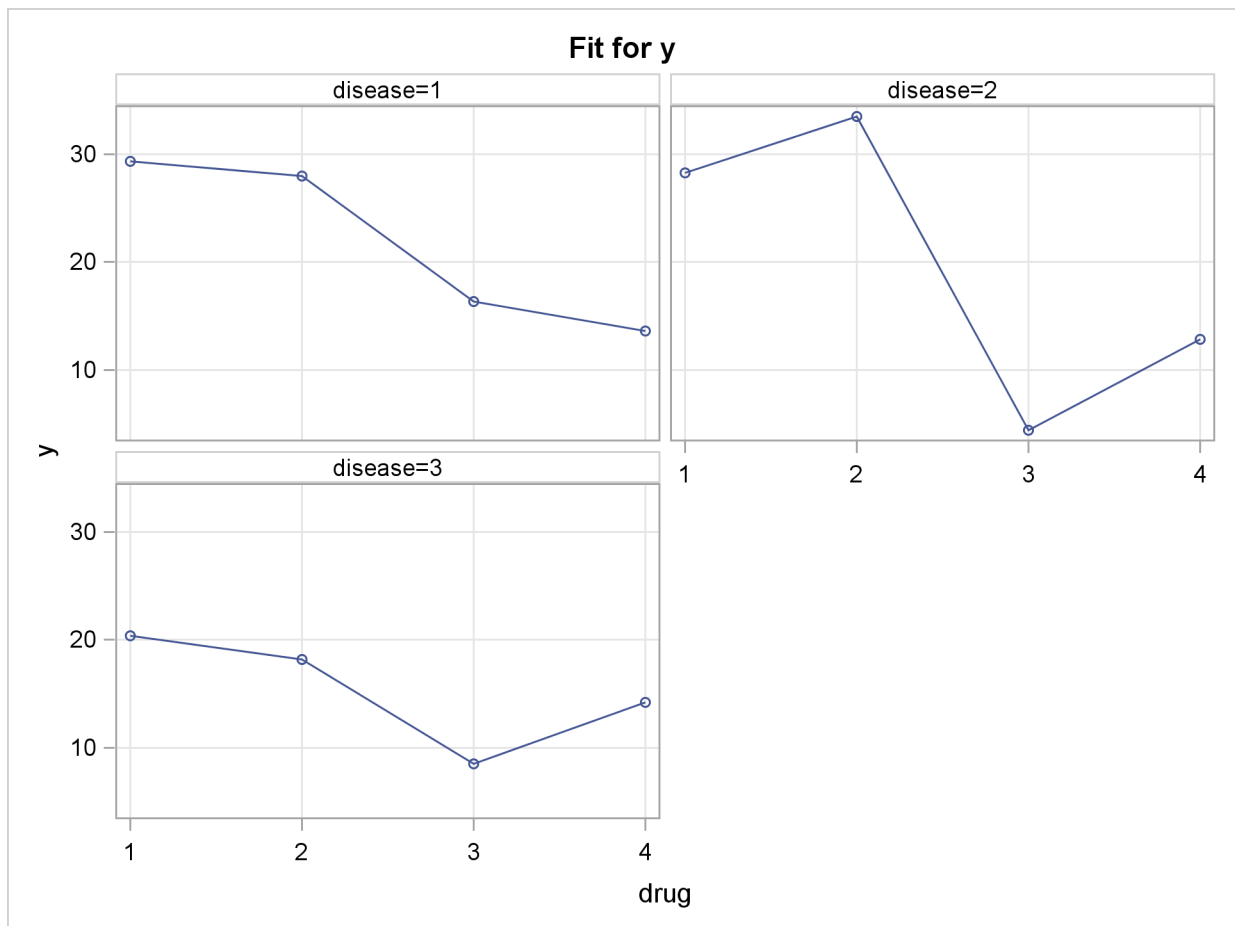
**Output 19.2.3** Box Fit Plot

Output 19.2.4 shows the interaction plot at every combination of Drug and Disease. This plot is identical to the preceding box plot, except the boxes are replaced by the actual observations. Again, it is difficult to see any pattern in the plot.



**Output 19.2.4** Interaction Plot with Specified X= Effect

**Output 19.2.5** groups the observations by Disease, and for each disease displays the effectiveness of the three drugs in a panel of plots.

**Output 19.2.5** Interaction Plot with Specified PLOTBY= Effect**Example 19.3: Logistic Regression**

Consider a study of the analgesic effects of treatments on elderly patients with neuralgia. Two test treatments and a placebo are compared. The response variable is whether the patient reported pain or not. Researchers recorded the age and gender of 60 patients and the duration of complaint before the treatment began.

The following DATA step creates the data set Neuralgia:

```

Data Neuralgia;
  input Treatment $ Sex $ Age Duration Pain $ @@;
  datalines;
P F 68 1 No B M 74 16 No P F 67 30 No
P M 66 26 Yes B F 67 28 No B F 77 16 No
A F 71 12 No B F 72 50 No B F 76 9 Yes
A M 71 17 Yes A F 63 27 No A F 69 18 Yes
B F 66 12 No A M 62 42 No P F 64 1 Yes
A F 64 17 No P M 74 4 No A F 72 25 No
P M 70 1 Yes B M 66 19 No B M 59 29 No
A F 64 30 No A M 70 28 No A M 69 1 No
B F 78 1 No P M 83 1 Yes B F 69 42 No
B M 75 30 Yes P M 77 29 Yes P F 79 20 Yes
A M 70 12 No A F 69 12 No B F 65 14 No
B M 70 1 No B M 67 23 No A M 76 25 Yes
P M 78 12 Yes B M 77 1 Yes B F 69 24 No
P M 66 4 Yes P F 65 29 No P M 60 26 Yes
A M 78 15 Yes B M 75 21 Yes A F 67 11 No
P F 72 27 No P F 70 13 Yes A M 75 6 Yes
B F 65 7 No P F 68 27 Yes P M 68 11 Yes
P M 67 17 Yes B M 70 22 No A M 65 15 No
P F 67 1 Yes A M 67 10 No P F 72 11 Yes
A F 74 1 No B M 80 21 Yes A F 69 3 No
;

```

The Neuralgia data set contains five variables. The Pain variable is the response. A specification of Pain=Yes indicates that the patient felt pain, and Pain=No indicates that the patient did not feel pain. The variable Treatment is a categorical variable with three levels: A and B represent the two test treatments, and P represents the placebo treatment. The gender of the patients is given by the categorical variable Sex. The variable Age is the age of the patients, in years, when treatment began. The duration of complaint, in months, before the treatment began is given by the variable Duration.

In the following statements, a complex model that includes classification and continuous covariates and an interaction term is fit to the Neuralgia data. When you try to create a default effect plot from this model, computations stop because the best type of plot cannot easily be determined.

```

ods graphics on;
proc logistic data=Neuralgia;
  class Treatment Sex / param=ref;
  model Pain= Treatment|Sex Age Duration;
  effectplot;
run;
ods graphics off;

```

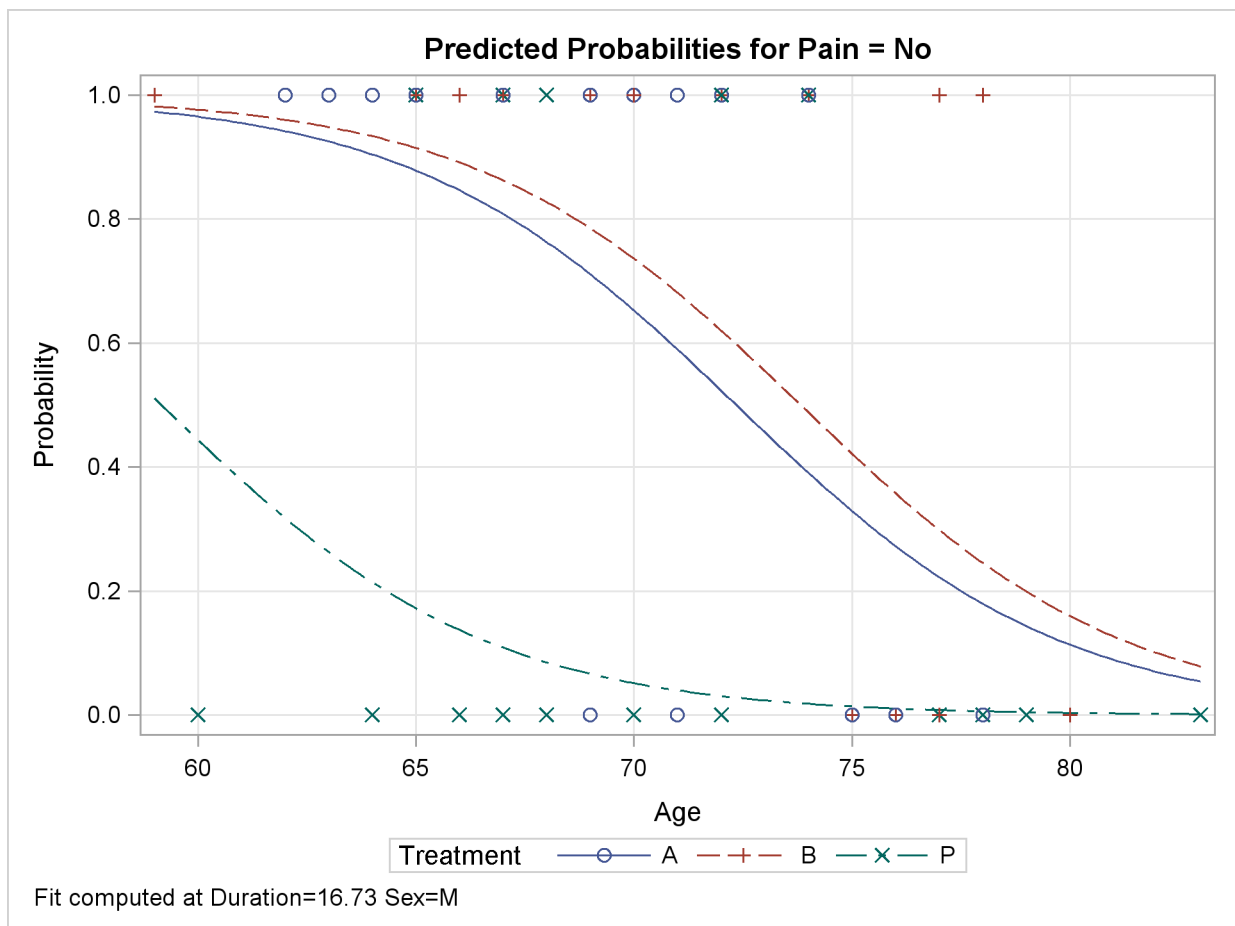
To produce an effect plot for this model, you need to first choose the type of plot to be created. In this case, since there are both classification and continuous covariates on the model, a [SLICEFIT plot-type](#) displays the first continuous covariate (Age) on the X axis and displays fit curves that correspond to each level of the first classification covariate (Treatment).

The following statements produce [Output 19.3.1](#).

```
ods graphics on;
proc logistic data=Neuralgia;
  class Treatment Sex / param=ref;
  model Pain= Treatment|Sex Age Duration;
  effectplot slice;
run;
ods graphics off;
```

By default, effect plots from PROC LOGISTIC are displayed on the probability scale. The predicted values are computed at the mean of the Duration variable, 16.73, and at the reference level of the Sex variable, M. Observations are also displayed on the sliced-fit plot in [Output 19.3.1](#). While the display of binary responses can give you a feel for the spread of the data, it does not enable you to evaluate the fit of the model.

**Output 19.3.1** Default Fit Plot Sliced by Treatment

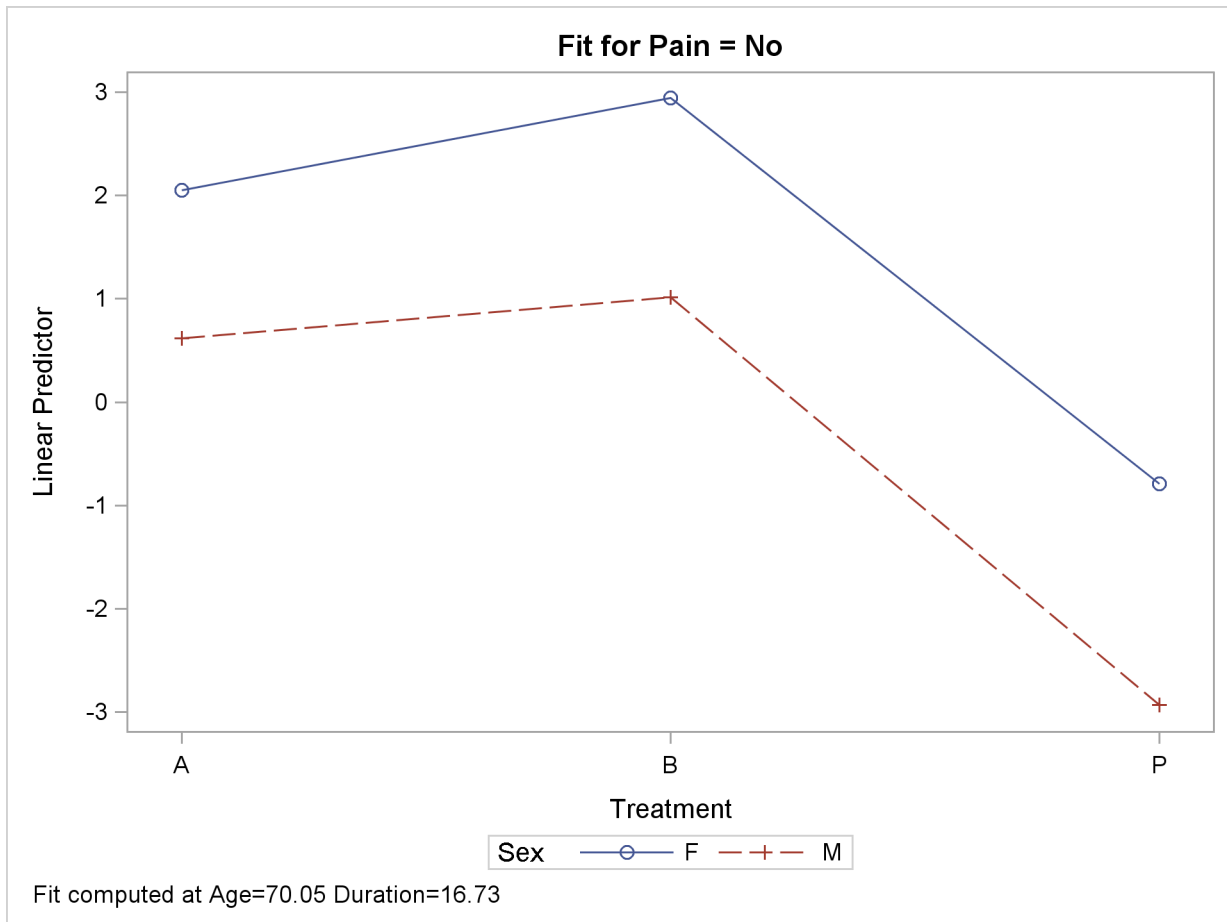


In the following statements, an [INTERACTION](#) *plot-type* is specified for the Treatment variable, with the Sex effect chosen for grouping the fits. The Age and Duration variables are set to their mean values for computing the predicted values. The [NOOBS](#) option suppresses the display of the binary observations on this plot. The [LINK](#) option is specified to display the fit on the LOGIT scale; if there is no interaction between Treatment and Sex, then the resulting curves shown in [Output 19.3.2](#) will have similar slopes across the treatments.

```
ods graphics on;
proc logistic data=Neuralgia;
  class Treatment Sex / param=ref;
  model Pain= Treatment|Sex Age Duration;
  effectplot interaction(x=Treatment sliceby=Sex) / noobs link;
run;
ods graphics off;
```

In [Output 19.3.2](#), the slopes of the lines seem “parallel” across the treatments, corroborating the nonsignificance of the interaction terms.

**Output 19.3.2** Interaction Plot of an Interaction Effect



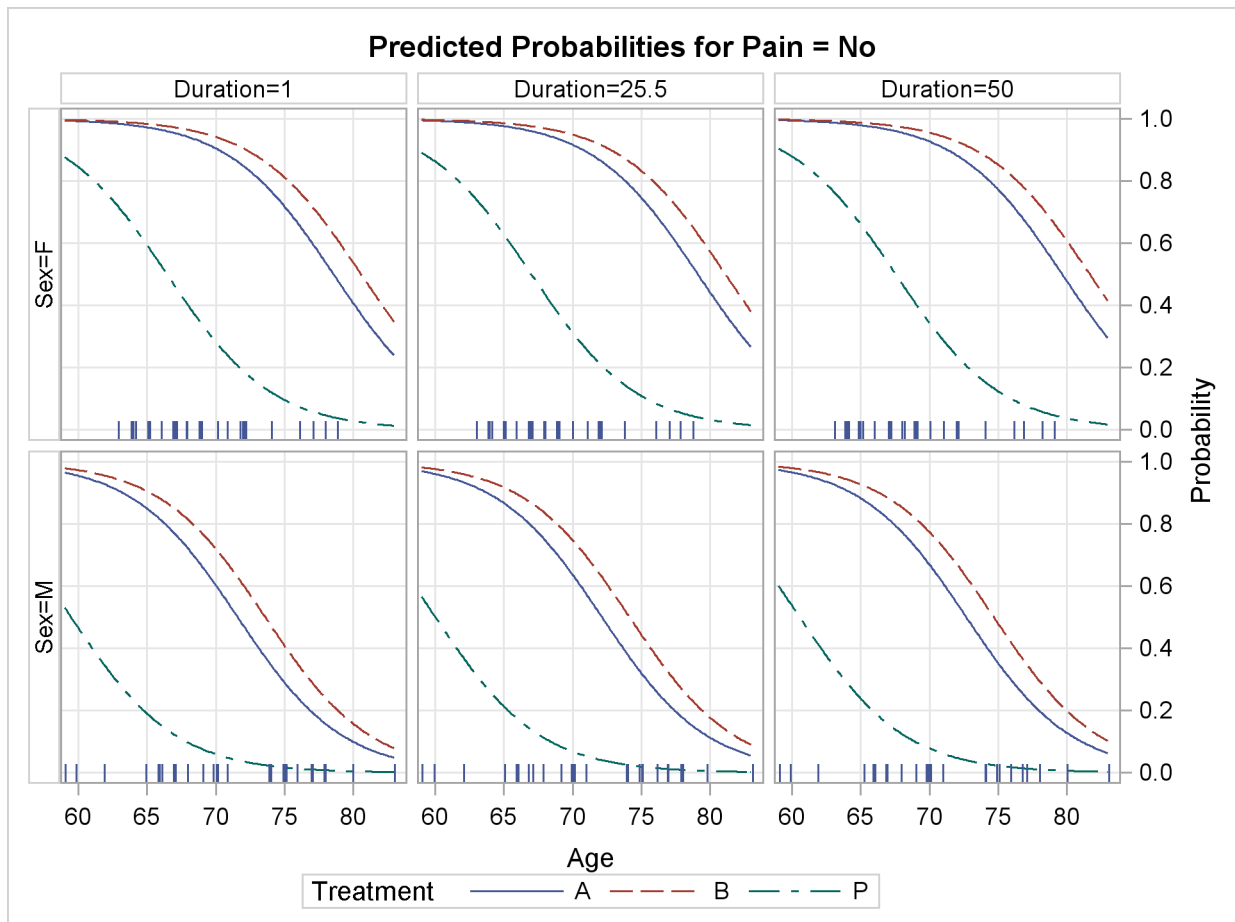
In the following statements, the interaction effect is removed, and the Duration variable is investigated further. The **PLOTBY(ROWS)=** option displays the Sex levels in the rows of a panel of plots, and the **AT** option computes the fits for several values of the Duration main effect in the columns of the panel. The **OBS(FRIDGE)** option moves the observations to a fringe (rug) plot at the bottom of the plot, the observations are subsetting and displayed according to the value of the **PLOTBY=** variable, and the **JITTER** option makes overlaid fringes more visible. A **STORE** statement is also specified to save the model information for a later display.

These statements produce [Output 19.3.3](#).

```
ods graphics on;
proc logistic data=Neuralgia;
  class Treatment Sex / param=ref;
  model Pain= Treatment Sex Age Duration;
  effectplot slicefit(sliceby=Treatment plotby(rows)=Sex)
    / at(Duration=min midrange max) obs(fringe jitter);
  store logimodel;
run;
ods graphics off;
```

The predicted probability curves in [Output 19.3.3](#) look very similar across the different values of the Duration variable, which agrees with the nonsignificance of Duration in this model. The fringe plot displays only female patients in the SEX=F row of the panel and displays only male patients in the SEX=M row, because the `PLOTBY=SEX` option subsets the observations.

**Output 19.3.3** Sliced-Fit Plot with AT Option

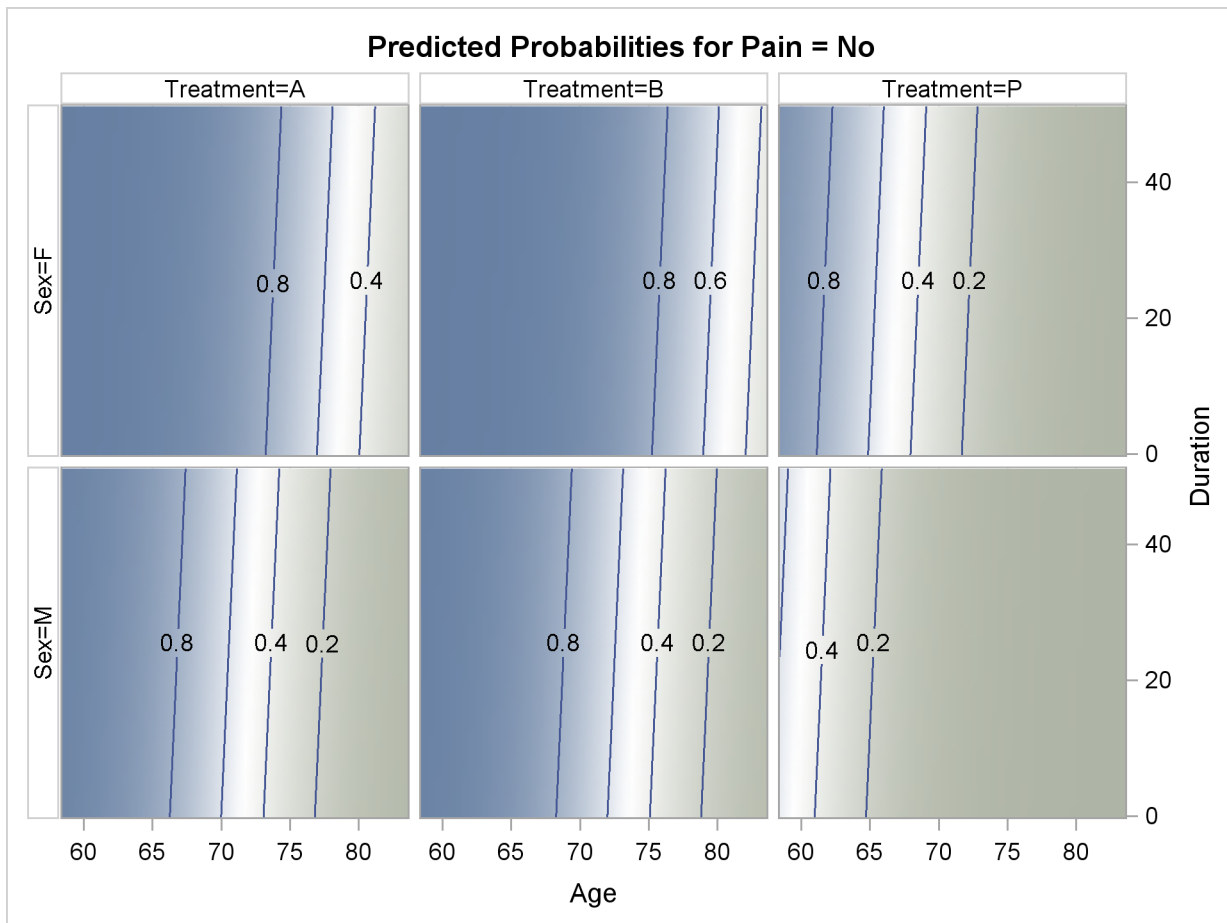


The following statements use the stored model and the PLM procedure to display a panel of contour plots:

```
ods graphics on;
proc plm source=logimodel;
  effectplot contour(plotby=Treatment) / at(Sex=all);
run;
ods graphics off;
```

Output 19.3.4 again confirms that Duration is not significant.

**Output 19.3.4** Contour Fit Panel



## ESTIMATE Statement

This statement documentation applies to the following procedures:

LOGISTIC, ORTHOREG, PHREG, PLM, SURVEYLOGISTIC, SURVEYPHREG, and SURVEYREG.

The ESTIMATE statement in the GENMOD, GLIMMIX, GLM, and MIXED procedures are documented in the respective procedure chapters.

The ESTIMATE statement provides a mechanism for obtaining custom hypothesis tests. Estimates are formed as linear estimable functions of the form  $\mathbf{L}\boldsymbol{\beta}$ . You can perform hypothesis tests for the estimable functions, construct confidence limits, and obtain specific nonlinear transformations.

### Syntax: ESTIMATE Statement

```
ESTIMATE <'label'> estimate-specification <(divisor=n)>
      <,<'label'> estimate-specification <(divisor=n)> > <,<...>
      </options>;
```

The basic element of the ESTIMATE statement is the *estimate-specification*, which consists of model effects and their coefficients. A *estimate-specification* takes the general form

*effect name < effect values ... >*

The following variables can appear in the ESTIMATE statement:

<i>label</i>	is an optional label that identifies the particular row of the estimate in the output.
<i>effect</i>	identifies an effect that appears in the MODEL statement. The keyword INTERCEPT can be used as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.
<i>values</i>	are constants that are elements of the $\mathbf{L}$ matrix and are associated with the fixed and random effects. There are two basic methods of specifying the entries of the $\mathbf{L}$ matrix. The traditional representation—also known as the positional syntax—relies on entering coefficients in the position they assume in the $\mathbf{L}$ matrix. For example, in the following statements the elements of $\mathbf{L}$ that are associated with the b main effect receive a 1 in the first position and a $-1$ in the second position:

```
class a b;
model y = a b a*b;
estimate 'B at A2' b 1 -1 a*b 0 0 1 -1;
```



The elements that are associated with the interaction receive a 1 in the third position and a  $-1$  in the fourth position. In order to specify coefficients correctly for the interaction term, you need to know how the levels of *a* and *b* vary in the interaction, which is governed by the order of the variables in the CLASS statement. The nonpositional syntax is designed to make it easier to enter coefficients for interactions and is necessary to enter coefficients for effects that are constructed with the EFFECT statement. In square brackets you enter the coefficient followed by the associated levels of the CLASS variables. If *B* has two levels and *A* has three levels, the previous ESTIMATE statement, by using nonpositional syntax for the interaction term, becomes the following statement:

```
estimate 'B at A2' b 1 -1 a*b [1, 2 1] [-1, 2 2];
```

The previous statement assigns value 1 to the interaction where *A* is at level 2 and *B* is at level 1, and it assigns  $-1$  to the interaction where both classification variables are at level 2. The comma that separates the entry for the **L** matrix from the level indicators is optional. Further details about the nonpositional contrast syntax and its use with constructed effects can be found in the section “[Positional and Nonpositional Syntax for Coefficients in Linear Functions](#)” on page 473.

Based on the *estimate-specifications* in your ESTIMATE statement, the procedure constructs the matrix **L** to test the hypothesis  $H: \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$ . The procedure supports nonpositional syntax for the coefficients of model effects in the ESTIMATE statement. For details see the section “[Positional and Nonpositional Syntax for Coefficients in Linear Functions](#)” on page 473.

The procedure then produces for each row **l** of **L** an approximate *t* test of the hypothesis  $H: \mathbf{l}\boldsymbol{\beta} = 0$ . You can also obtain multiplicity-adjusted *p*-values and confidence limits for multirow estimates with the **ADJUST=** option.

Note that multirow estimates are permitted. Unlike releases prior to SAS 9.22, you do not need to specify a ‘*label*’ for every row of the estimate; the procedure constructs a default label if a label is not specified.

If the procedure finds the estimate to be nonestimable, then it displays “Non-est” for the estimate entry.

Table 19.16 summarizes important options in the ESTIMATE statement. All ESTIMATE options are subsequently discussed in alphabetical order.

**Table 19.16** Important ESTIMATE Statement Options

Option	Description
<b>Construction and Computation of Estimable Functions</b>	
<b>DIVISOR=</b>	Specifies a list of values to divide the coefficients
<b>NOFILL</b>	Suppresses the automatic fill-in of coefficients for higher-order effects
<b>SINGULAR=</b>	Tunes the estimability checking difference

**Table 19.16** *continued*

Option	Description
<b>Degrees of Freedom and <math>p</math>-values</b>	
<b>ADJUST=</b>	Determines the method for multiple comparison adjustment of estimates
<b>ALPHA=<math>\alpha</math></b>	Determines the confidence level ( $1 - \alpha$ )
<b>LOWER</b>	Performs one-sided, lower-tailed inference
<b>STEPDOWN</b>	Adjusts multiplicity-corrected $p$ -values further in a step-down fashion
<b>TESTVALUE=</b>	Specifies values under the null hypothesis for tests
<b>UPPER</b>	Performs one-sided, upper-tailed inference
<b>Statistical Output</b>	
<b>CL</b>	Constructs confidence limits
<b>CORR</b>	Displays the correlation matrix of estimates
<b>COV</b>	Displays the covariance matrix of estimates
<b>E</b>	Prints the <b>L</b> matrix
<b>JOINT</b>	Produces a joint $F$ or chi-square test for the estimable functions
<b>PLOTS=</b>	Requests ODS statistical graphics if the analysis is sampling-based
<b>SEED=</b>	Specifies the seed for computations that depend on random numbers
<b>Generalized Linear Modeling</b>	
<b>CATEGORY=</b>	Specifies how to construct estimable functions with multinomial data
<b>EXP</b>	Exponentiates and displays estimates
<b>ILINK</b>	Computes and displays estimates and standard errors on the inverse linked scale

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

**ADJDFE=SOURCE****ADJDFE=ROW**

specifies how denominator degrees of freedom are determined when  $p$ -values and confidence limits are adjusted for multiple comparisons with the **ADJUST=** option. When you do not specify the **ADJDFE=** option, or when you specify **ADJDFE=SOURCE**, the denominator degrees of freedom for multiplicity-adjusted results are the denominator degrees of freedom for the final effect that is listed in the ESTIMATE statement from the “Type III” table.

The **ADJDFE=ROW** setting is useful if you want multiplicity adjustments to take into account that denominator degrees of freedom are not constant across estimates. For example, this can be the case when the denominator degrees of freedom are computed by the Satterthwaite method or according to Kenward and Roger (1997).

The **ADJDFE=** option has an effect only in mixed models that use these degree-of-freedom methods. It is not supported by the procedures that perform chi-square-based inference (LOGISTIC, PHREG, and SURVEYLOGISTIC).

**ADJUST=BON****ADJUST=SCHEFFE****ADJUST=SIDAK****ADJUST=SIMULATE**< (*simoptions*) >**ADJUST=T**

requests a multiple comparison adjustment for the  $p$ -values and confidence limits for the estimates. The adjusted quantities are produced in addition to the unadjusted quantities. Adjusted confidence limits are produced if the **CL** or **ALPHA=** option is in effect. For a description of the adjustments, see Chapter 39, “The GLM Procedure,” and Chapter 58, “The MULTTEST Procedure,” and the documentation for the **ADJUST=** option in the **LSMEANS** statement.

If the **STEPDOWN** option is in effect, the  $p$ -values are further adjusted in a step-down fashion.

**ALPHA=***number*

requests that a  $t$  type confidence interval be constructed with confidence level  $1 - \textit{number}$ . The value of *number* must be between 0 and 1; the default is 0.05. If the “Estimates” table shows infinite degrees of freedom, then the confidence interval is a  $z$  type interval.

**CATEGORY=***category-options*

specifies how to construct estimates and multiplicity corrections for models with multinomial data (ordinal or nominal). This option is also important for constructing sets of estimable functions for  $F$  or chi-square tests with the **JOINT** option.

The *category-options* are used to indicate how response variable levels are treated in constructing the estimable functions. Possible values for the *category-options* are the following:

**JOINT**

computes the estimable functions for every nonredundant category and treats them as a set. For example, a three-row ESTIMATE statement in a model with three response categories leads to six estimable functions.

**SEPARATE**

computes the estimable functions for every nonredundant category in turn. For example, a three-row ESTIMATE statement in a model with three response categories leads to two sets of three estimable functions.

*quoted-value-list*

computes the estimable functions only for the list of values given. The list must consist of formatted values of the response categories.

Consider the following ESTIMATE statements in the LOGISTIC procedure for an ordinal model with response categories ‘vg’, ‘g’, ‘m’, ‘b’, and ‘vb’. Because there are five response categories, there are four nonredundant categories for the cumulative link model.

```

proc logistic data=icecream;
  class brand / param=glm;
  model taste(order=data) = brand / link=logit;
  freq count;

  estimate brand 1 -1,
            intercept 1 brand 0 1 / category='m', 'vg';

  estimate intercept 1 brand 1      / category=joint
            adjust=simulate(seed=1);

  estimate brand 1 -1,
            brand 1 1 -2            / category=separate
            adjust=bon;

run;

```

The first ESTIMATE statement requests a two-row estimable function. The result is produced for two of the four nonredundant response categories. The second ESTIMATE statement produces four  $t$  tests, one for each nonredundant category. The multiplicity adjustment with  $p$ -value computation by simulation treats the four estimable functions as a unit for family-wise Type I error protection. The third ESTIMATE statement computes a two-row estimable function and reports its results separately for all nonredundant categories. The Bonferroni adjustment in this statement applies to a family of two tests that correspond to the two-row estimable function. Four Bonferroni adjustments for sets of size two are performed.

The CATEGORY= option is supported only by the procedures that support generalized linear modeling (LOGISTIC and SURVEYLOGISTIC) and by PROC PLM when it is used to perform statistical analyses on item stores created by these procedures.

### CHISQ

requests that chi-square tests be performed in addition to  $F$  tests, when you request an  $F$  test with the JOINT option. This option has no effect in procedures that produce chi-square statistics by default.

### CL

requests that  $t$  type confidence limits be constructed. If the procedure shows the degrees of freedom in the “Estimates” table as infinite, then the confidence limits are  $z$  intervals. The confidence level is 0.95 by default, and you can change the confidence level with the ALPHA= option. The confidence intervals are adjusted for multiplicity when you specify the ADJUST= option. However, if a step-down  $p$ -value adjustment is requested with the STEPDOWN option, only the  $p$ -values are adjusted for multiplicity.

### CORR

displays the estimated correlation matrix of the linear combination of the parameter estimates.

### COV

displays the estimated covariance matrix of the linear combination of the parameter estimates.

**DF=number**

specifies the degrees of freedom for the  $t$  test and confidence limits. This option is not supported by the procedures that perform chi-square-based inference (LOGISTIC, PHREG, and SURVEYLOGISTIC).

**DIVISOR=value-list**

specifies a list of values by which to divide the coefficients so that fractional coefficients can be entered as integer numerators. If you do not specify *value-list*, a default value of 1.0 is assumed. Missing values in the *value-list* are converted to 1.0.

If the number of elements in *value-list* exceeds the number of rows of the estimate, the extra values are ignored. If the number of elements in *value-list* is less than the number of rows of the estimate, the last value in *value-list* is copied forward.

If you specify a row-specific divisor as part of the specification of the estimate row, this value multiplies the corresponding divisor that is implied by the *value-list*. For example, the following statement divides the coefficients in the first row by 8, and the coefficients in the third and fourth row by 3:

```
estimate 'One vs. two'   A 2 -2 (divisor=2),
         'One vs. three' A 1  0 -1          ,
         'One vs. four'  A 3  0  0 -3        ,
         'One vs. five'  A 1  0  0  0 -1 / divisor=4,.,3;
```

Coefficients in the second row are not altered.

**E**

requests that the **L** matrix coefficients be displayed.

**EXP**

requests exponentiation of the estimate. When you model data with the logit, cumulative logit, or generalized logit link functions, and the estimate represents a log odds ratio or log cumulative odds ratio, the EXP option produces an odds ratio. In proportional hazards model, this option produces estimates of hazard ratios. If you specify the **CL** or **ALPHA=** option, the (adjusted) confidence bounds are also exponentiated.

The EXP option is supported only by PROC PHREG, PROC SURVEYPHREG, the procedures that support generalized linear modeling (LOGISTIC and SURVEYLOGISTIC), and by PROC PLM when it is used to perform statistical analyses on item stores created by these procedures.

**ILINK**

requests that the estimate and its standard error also be reported on the scale of the mean (the inverse linked scale). The computation of the inverse linked estimate depends on the estimation mode. For example, if the analysis is based on a posterior sample when a BAYES statement is present, the inversely linked estimate is the average of the inversely linked values across the sample of posterior parameter estimates. If the analysis is not based on a sample of parameter estimates, the procedure computes the value on the mean scale by applying the inverse link to the estimate.

The interpretation of this quantity depends on the *effect values* specified in your ESTIMATE statement and on the link function. For example, in a model for binary data with logit link the following statements compute

$$\frac{1}{1 + \exp\{-(\alpha_1 - \alpha_2)\}}$$

where  $\alpha_1$  and  $\alpha_2$  are the fixed-effects solutions that are associated with the first two levels of the classification effect A:

```
class A;
model y = A / dist=binary link=logit;
estimate 'A one vs. two' A 1 -1 / ilink;
```

This quantity is not the difference of the probabilities that are associated with the two levels,

$$\pi_1 - \pi_2 = \frac{1}{1 + \exp\{-\beta_0 - \alpha_1\}} - \frac{1}{1 + \exp\{-\beta_0 - \alpha_2\}}$$

The standard error of the inversely linked estimate is based on the delta method. If you also specify the **CL** option, the procedure computes confidence limits for the estimate on the mean scale. In multinomial models for nominal data, the limits are obtained by the delta method. In other models they are obtained from the inverse link transformation of the confidence limits for the estimate. The **ILINK** option is specific to an ESTIMATE statement.

The **ILINK** option is supported only by the procedures that support generalized linear modeling (**LOGISTIC** and **SURVEYLOGISTIC**) and by **PROC PLM** when it is used to perform statistical analyses on item stores created by these procedures.

#### **JOINT**<(joint-test-options)>

requests that a joint *F* or chi-square test be produced for the rows of the estimate. The **JOINT** option in the ESTIMATE statement essentially replaces the **CONTRAST** statement.

When the **LOWERTAILED** or the **UPPERTAILED** options are in effect, or if the **BOUNDS** option described below is in effect, the **JOINT** option produces the chi-bar-square statistic according to Silvapulle and Sen (2004). This statistic uses a simulation-based approach to compute *p*-values in situations where the alternative hypotheses of the estimable functions are not simple two-sided hypotheses. See the section “[Joint Hypothesis Tests with Complex Alternatives, the Chi-Bar-Square Statistic](#)” on page 476 for more information about this test statistic.

You can specify the following *joint-test-options* in parentheses:

#### **ACC**= $\gamma$

specifies the accuracy radius for determining the necessary sample size in the simulation-based approach of Silvapulle and Sen (2004) for tests with order restrictions. The value of  $\gamma$  must be strictly between 0 and 1; the default value is 0.005.

#### **EPS**= $\epsilon$

specifies the accuracy confidence level for determining the necessary sample size in the simulation-based approach of Silvapulle and Sen (2004) for tests with order restrictions. The value of  $\epsilon$  must be strictly between 0 and 1; the default value is 0.01.

**LABEL='label'**

assigns an identifying label to the joint test. If you do not specify a label, the first non-default label for the ESTIMATE rows is used to label the joint test.

**NOEST****ONLY**

performs only the  $F$  or chi-square test and suppresses other results from the ESTIMATE statement. This option is useful for emulating the CONTRAST statement that is available in other procedures.

**NSAMP= $n$** 

specifies the number of samples for the simulation-based method of Silvapulle and Sen (2004). If  $n$  is not specified, it is constructed from the values of the ALPHA= $\alpha$ , the ACC= $\gamma$ , and the EPS= $\epsilon$  options. With the default values for  $\gamma$ ,  $\epsilon$ , and  $\alpha$  (0.005, 0.01, and 0.05, respectively), NSAMP=12,604 by default.

**CHISQ**

adds a chi-square test if the procedure produces an  $F$  test by default.

**BOUNDS=value-list**

specifies boundary values for the estimable linear function. The null value of the hypothesis is always zero. If you specify a positive boundary value  $z$ , the hypotheses are  $H: \theta = 0$ ,  $H_a: \theta > 0$  with the added constraint that  $\theta < z$ . The same is true for negative boundary values. The alternative hypothesis is then  $H_a: \theta < 0$  subject to the constraint  $\theta > -|z|$ . If you specify a missing value, the hypothesis is assumed to be two-sided. The BOUNDS option enables you to specify sets of one- and two-sided joint hypotheses. If all values in *value-list* are set to missing, the procedure performs a simulation-based  $p$ -value calculation for a two-sided test.

**LOWER****LOWERTAILED**

requests that the  $p$ -value for the  $t$  test be based only on values that are less than the test statistic. A two-tailed test is the default. A lower-tailed confidence limit is also produced if you specify the CL or ALPHA= option.

Note that for ADJUST=SCHEFFE the one-sided adjusted confidence intervals and one-sided adjusted  $p$ -values are the same as the corresponding two-sided statistics, because this adjustment is based on only the right tail of the  $F$  distribution.

If you request a joint test with the JOINT option, then a one-sided left-tailed order restriction is applied to all estimable functions, and the corresponding chi-bar-square statistic of Silvapulle and Sen (2004) is computed in addition to the two-sided, standard,  $F$  or chi-square statistic. See the JOINT option for how to control the computation of the simulation-based chi-bar-square statistic.

**NOFILL**

suppresses the automatic fill-in of coefficients of higher-order effects.

**PLOTS=***plot-options*

produces ODS statistical graphics of the distribution of estimable functions if the procedure performs the analysis in a sampling-based mode. For example, this is the case when procedures support a BAYES statement and perform a Bayesian analysis. The estimable functions are then computed for each of the posterior parameter estimates, and the “Estimates” table reports simple descriptive statistics for the evaluated functions. The PLOTS= option enables you in this situation to visualize the distribution of the estimable function. The following *plot-options* are available:

**ALL**

produces all possible plots with their default settings.

**BOXPLOT**<(*boxplot-options*)>

produces box plots of the distribution of the estimable function across the posterior sample. A separate box is generated for each estimable function, and all boxes appear on a single graph by default. You can affect the appearance of the box plot graph with the following options:

**ORIENTATION=VERTICAL | HORIZONTAL**

**ORIENT=VERT | HORIZ** specifies the orientation of the boxes. The default is vertical orientation of the box plots.

**NPANELPOS=number** specifies how to break the series of box plots across multiple panels. If the NPANELPOS option is not specified, or if *number* equals zero, then all box plots are displayed in a single graph; this is the default. If a negative number is specified, then exactly up to  $|number|$  of box plots are displayed per panel. If *number* is positive, then the number of boxes per panel is balanced to achieve small variation in the number of box plots per graph.

**DISTPLOT**<(*distplot-options*)>**DIST**<(*distplot-options*)>

generates panels of histograms with a kernel density overlaid. A separate plot in each panel contains the results for each estimable function. You can specify the following *distplot-options* in parentheses:

**BOX | NOBOX** controls the display of a horizontal box plot of the estimable function’s distribution across the posterior sample below the graph. The BOX option is enabled by default.

**HIST | NOHIST** controls the display of the histogram of the estimable function’s distribution across the posterior sample. The HIST option is enabled by default.

**NORMAL | NONORMAL** controls the display of a normal density estimate on the graph. The NONORMAL option is enabled by default.

**KERNEL | NOKERNEL** controls the display of a kernel density estimate on the graph. The KERNEL option is enabled by default.

**NROWS=number** specifies the highest number of rows in a panel. The default is 3.



**NCOLS=number** specifies the highest number of columns in a panel. The default is 3.

**UNPACK** unpacks the panel into separate graphics.

#### **NONE**

does not produce any plots.

#### **SEED=number**

specifies the seed for the sampling-based components of the computations for the ESTIMATE statement (for example, chi-bar-square statistics and simulated  $p$ -values). *number* specifies an integer that is used to start the pseudo-random number generator for the simulation. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock. There could be multiple ESTIMATE statements with SEED= specifications and there could be other statements that can supply a random number seed. Since the procedure has only one random number stream, the initial seed is shown in the SAS log.

#### **SINGULAR=number**

tunes the estimability checking. If  $\mathbf{v}$  is a vector, define  $\text{ABS}(\mathbf{v})$  to be the largest absolute value of the elements of  $\mathbf{v}$ . If  $\text{ABS}(\mathbf{L} - \mathbf{LT})$  is greater than  $c * \text{number}$  for any row of  $\mathbf{L}$  in the contrast, then  $\mathbf{L}\boldsymbol{\beta}$  is declared nonestimable. Here,  $\mathbf{T}$  is the Hermite form matrix  $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}$ , and  $c$  is  $\text{ABS}(\mathbf{L})$ , except when it equals 0, and then  $c$  is 1. The value for *number* must be between 0 and 1; the default is 1E-4.

#### **STEPDOWN**< (*step-down-options*) >

requests that multiplicity adjustments for the  $p$ -values of estimates be further adjusted in a step-down fashion. Step-down methods increase the power of multiple testing procedures by taking advantage of the fact that a  $p$ -value is never declared significant unless all smaller  $p$ -values are also declared significant. The STEPDOWN adjustment combined with **ADJUST=BON** corresponds to the methods of Holm (1979) and “Method 2” of Shaffer (1986); this is the default. Using step-down-adjusted  $p$ -values combined with **ADJUST=SIMULATE** corresponds to the method of Westfall (1997).

If the ESTIMATE statement is applied with a STEPDOWN option in a mixed model where the degrees-of-freedom method is that of Kenward and Roger (1997) or of Satterthwaite, then step-down-adjusted  $p$ -values are produced only if the **ADJDFE=ROW** option is in effect.

Also, the STEPDOWN option affects only  $p$ -values, not confidence limits. For **ADJUST=SIMULATE**, the generalized least squares hybrid approach of Westfall (1997) is used to increase Monte Carlo accuracy. You can specify the following *step-down-options* in parentheses after the STEPDOWN option:

#### **MAXTIME=n**

specifies the time (in seconds) to be spent computing the maximal logically consistent sequential subsets of equality hypotheses for **TYPE=LOGICAL**. The default is **MAXTIME=60**. If the MAXTIME value is exceeded, the adjusted tests are not computed. When this occurs, you can try increasing the MAXTIME value. However, note that there are common multiple comparisons problems for which this computation requires a huge amount of time—for example, all pairwise comparisons between more than 10 groups. In such cases, try to use **TYPE=FREE** (the default) or **TYPE=LOGICAL(n)** for small  $n$ .

**ORDER=PVALUE****ORDER=ROWS**

specifies the order in which the step-down tests to be performed. ORDER=PVALUE is the default, with estimates being declared significant only if all estimates with smaller (unadjusted)  $p$ -values are significant. If you specify ORDER=ROWS, then significances are evaluated in the order in which they are specified in the syntax.

**REPORT**

specifies that a report on the step-down adjustment be displayed, including a listing of the sequential subsets (Westfall 1997) and, for ADJUST=SIMULATE, the step-down simulation results.

**TYPE=LOGICAL<(n)>****TYPE=FREE**

specifies how step-down adjustment are made. If you specify TYPE=LOGICAL, the step-down adjustments are computed by using maximal logically consistent sequential subsets of equality hypotheses (Shaffer 1986, Westfall 1997). Alternatively, for TYPE=FREE, sequential subsets are computed ignoring logical constraints. The TYPE=FREE results are more conservative than those for TYPE=LOGICAL, but they can be much more efficient to produce for many estimates. For example, it is not feasible to take logical constraints between all pairwise comparisons of more than about 10 groups. For this reason, TYPE=FREE is the default.

However, you can reduce the computational complexity of taking logical constraints into account by limiting the depth of the search tree used to compute them, specifying the optional depth parameter as a number  $n$  in parentheses after TYPE=LOGICAL. As with TYPE=FREE, results for TYPE=LOGICAL( $n$ ) are conservative relative to the true TYPE=LOGICAL results. But even for TYPE=LOGICAL(0) they can be appreciably less conservative than TYPE=FREE, and they are computationally feasible for much larger numbers of estimates. If you do not specify  $n$  or if  $n = -1$ , the full search tree is used.

**TESTVALUE=value-list****TESTMEAN=value-list**

specifies the value under the null hypothesis for testing the estimable functions in the ESTIMATE statement. The rules for specifying the *value-list* are very similar to those for specifying the divisor list in the DIVISOR= option. If no TESTVALUE= is specified, all tests are performed as  $H: \mathbf{L}\boldsymbol{\beta} = 0$ . Missing values in the *value-list* also are translated to zeros. If you specify fewer values than rows in the ESTIMATE statement, the last value in *value-list* is carried forward.

The TESTVALUE= option affects only  $p$ -values from individual, joint, and multiplicity-adjusted tests. It does not affect confidence intervals.

The TESTVALUE option is not available for the multinomial distribution, and the values are ignored when you perform a sampling-based (Bayesian) analysis.

**UPPER****UPPERTAILED**

requests that the  $p$ -value for the  $t$  test be based only on values that are greater than the test statistic. A two-tailed test is the default. An upper-tailed confidence limit is also produced if you specify the **CL** or **ALPHA=** option.

Note that for **ADJUST=SCHEFFE** the one-sided adjusted confidence intervals and one-sided adjusted  $p$ -values are the same as the corresponding two-sided statistics, because this adjustment is based on only the right tail of the  $F$  distribution.

If you request a joint test with the **JOINT** option, then a one-sided right-tailed order restriction is applied to all estimable functions, and the corresponding chi-bar-square statistic of Silvapulle and Sen (2004) is computed in addition to the two-sided, standard,  $F$  or chi-square statistic. See the **JOINT** option for how to control the computation of the simulation-based chi-bar-square statistic.

---

## Positional and Nonpositional Syntax for Coefficients in Linear Functions

When you define custom linear hypotheses with the **ESTIMATE** statement, the procedure sets up an **L** vector or matrix that conforms to the model effect solutions. (Note that the following remarks also apply to the **LSMESTIMATE** statement, where you specify coefficients of the matrix **K** which is then converted into a coefficient matrix that conforms to the model effects solutions.)

There are two methods for specifying the entries in a coefficient matrix (hereafter simply referred to as the **L** matrix); they are called the positional and nonpositional methods. In the positional form, which is the traditional method, you provide a list of values that occupy the elements of the **L** matrix that is associated with the effect in question in the order in which the values are listed. For traditional model effects that consist of continuous and classification variables, the positional syntax is simpler in some cases (main effects) and more cumbersome in others (interactions). When you work with effects that are constructed through the **EFFECT** statement, the nonpositional syntax is essential.

For example, consider the following two-way model with interactions where factors A and B have three and two levels, respectively:

```
proc logistic;
  class a b;
  model y = a b a*b;
run;
```

To test the difference of the B levels at the second level of A with an **ESTIMATE** statement (a slice), you need to assign coefficients 1 and  $-1$  to the levels of B and to the levels of the interaction where A is at the second level. Two examples of equivalent **ESTIMATE** statements that use positional and nonpositional syntax are as follows:

```
estimate 'B at A2' b 1 -1 a*b 0 0 1 -1 ;
estimate 'B at A2' b 1 -1 a*b [1 2 1] [-1 2 2];
```

Because A precedes B in the CLASS statement, the levels of the interaction are formed as  $\alpha_1\beta_1, \alpha_1\beta_2, \alpha_2\beta_1, \alpha_2\beta_2, \dots$ . If B precedes A in the CLASS statement, you need to modify the coefficients accordingly:

```
proc logistic;
  class b a;
  model y = a b a*b;
  estimate 'B at A2' b 1 -1 a*b 0 1 0 0 -1 ;
  estimate 'B at A2' b 1 -1 a*b [1 1 2] [-1 2 2];
  estimate 'B at A2' b 1 -1 a*b [1, 1 2] [-1, 2 2];
run;
```

You can optionally separate the **L** value entry from the level indicators with a comma, as in the last **ESTIMATE** statement.

The general syntax for defining coefficients with the nonpositional syntax is as follows:

*effect-name* [*multiplier* <,> *level-values*] ...<[*multiplier* <,> *level-values*]>

The first entry in square brackets is the multiplier that is applied to the elements of **L** for the effect after the *level-values* have been resolved and any necessary action that forms **L** has been taken.

The *level-values* are organized in a specific form:

- The number of entries should equal the number of terms that are needed to construct the effect. For effects that do not contain any constructed effects, this number is simply the number of terms in the name of the effect.
- Values of continuous variables that are needed for the construction of the **L** matrix precede the level indicators of CLASS variables.
- If the effect involves constructed effects, then you need to provide as many continuous and classification variables as are needed for the effect formation. For example, if a grouping effect is defined as

```
class c;
effect v = vars(x1 x2 c);
```

then a proper nonpositional syntax would be

```
v [0.5, 0.2 0.3 3]
```

- If an effect contains both regular terms (old-style effects) and constructed effects, then the order of the coefficients is as follows: continuous values for old-style effects, class levels for classification variables in old-style effects, continuous values for constructed effects, and finally class levels that are needed for constructed effects.

Assume that C has four levels so that effect v contributes six elements to the **L** matrix. When the procedure resolves this syntax, the values 0.2 and 0.3 are assigned to the positions for x1 and x2 and a 1 is associated with the third level of C. The resulting vector is then multiplied by 0.5 to produce

$$[0.1 \quad 0.15 \quad 0 \quad 0 \quad 0.5 \quad 0]$$

Note that you enter the **levels** of the classification variables in the square brackets, not their formatted values. The ordering of the levels of classification variables can be gleaned from the “Class Level Information” table.

To specify values for continuous variables, simply give their value as one of the terms in the effect. The nonpositional syntax in the following **ESTIMATE** statement is read as “1 times the value 0.4 in the column that is associated with level 2 of A”

```
proc phreg;
  class a / param=glm;
  model y = a a*x / s;
  lsmeans a / e at x=0.4;
  estimate 'A2 at x=0.4' intercept 1 a 0 1 a*x [1,0.4 2] / e;
run;
```

Because the value before the comma serves as a multiplier, the same estimable function could also be constructed with the following statements:

```
estimate 'A2 at x=0.4' intercept 1 a 0 1 a*x [ 4, 0.1 2];
estimate 'A2 at x=0.4' intercept 1 a 0 1 a*x [ 2, 0.2 2];
estimate 'A2 at x=0.4' intercept 1 a 0 1 a*x [-1, -0.4 2];
```

Note that continuous variables that are needed to construct an effect are always listed before any **CLASS** variables.

When you work with constructed effects, the nonpositional syntax works in the same way. For example, the following model contains a classification effect and a B-spline. The first two **ESTIMATE** statements produce predicted values for level 1 of C when the continuous variable x takes on the values 20 and 10, respectively.

```
proc orthoreg;
  class c;
  effect spl = spline(x / knotmethod=equal(5));
  model y = c spl;
  estimate 'C = 1 @ x=20' intercept 1 c 1 spl [1,20],
          'C = 1 @ x=10' intercept 1 c 1 spl [1,10];
  estimate 'Difference' spl [1,20] [-1,10];
run;
```

In this example, the **ORTHOREG** procedure computes the spline coefficients for the first **ESTIMATE** statement based on  $x = 20$ , and similarly in the second statement for  $x = 10$ . The third **ESTIMATE** statement computes the difference of the predicted values. Because the spline effect does not interact with the classification variable, this difference does not depend on the level of C. If such an interaction is present, you can estimate the difference in predicted values for a given level of C by using the nonpositional syntax. Because the effect C\*spl contains both old-style terms (C) and a constructed effect, you specify the values for the old-style terms before assigning values to constructed effects.

```

proc orthoreg;
  class c;
  effect spl = spline(x / knotmethod=equal(5));
  model y = spl*c;
  estimate 'C2 = 1, x=20' intercept 1 c*spl [1,1 20];
  estimate 'C2 = 2, x=20' intercept 1 c*spl [1,2 20];
  estimate 'C diff at x=20' c*spl [1,1 20] [-1,2 20];
run;

```

It is recommended that you add the E option to the [ESTIMATE](#) or [LSMESTIMATE](#) statement to verify that the **L** matrix is formed according to your expectations.

In any row of an [ESTIMATE](#) statement you can choose positional and nonpositional syntax separately for each effect. However, you cannot mix the two forms of syntax for coefficients of a single effect. For example, the following statement is not proper because both forms of syntax are used for the interaction effect:

```

estimate 'A1B1 - A1B2' b 1 -1 a*b 0 1 [-1, 1 2];

```

---

## Joint Hypothesis Tests with Complex Alternatives, the Chi-Bar-Square Statistic

Silvapulle and Sen (2004) propose a test statistic for testing hypotheses where the null or the alternative hypothesis or both involve inequalities. You can test special cases of these hypotheses with the JOINT option in the [ESTIMATE](#) and the [LSMESTIMATE](#) statement. Consider the  $k$  estimable functions  $\mathbf{L}\boldsymbol{\beta}$  and the hypotheses  $H_0: \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$  and  $H_a: \mathbf{L}\boldsymbol{\beta} \geq \mathbf{0}$ . The alternative hypothesis defines a convex cone  $\mathcal{C}$  at the origin. Suppose that under the null hypothesis  $\widehat{\mathbf{L}\boldsymbol{\beta}}$  follows a multivariate normal distribution with mean  $\mathbf{0}$  and variance  $\mathbf{V}$ . The restricted alternative prevents you from using the usual  $F$  or chi-square test machinery, since the distribution of the test statistic under the alternative might not follow the usual rules. Silvapulle and Sen (2004) coined a statistic that takes into account the projection of the observed estimate onto the convex cone formed by the alternative parameter space. This test statistic is called the chi-bar-square statistic, and  $p$ -values are obtained by simulation; see, in particular, Chapter 3.4 in Silvapulle and Sen (2004).

Briefly, let  $\mathbf{U}$  be a multivariate normal random variable with mean  $\mathbf{0}$  and variance matrix  $\mathbf{V}$ . The chi-bar-square statistic is the random variable

$$\begin{aligned}\bar{\chi}^2 &= \mathbf{U}'\mathbf{V}^{-1}\mathbf{U} - Q \\ Q &= \min_{\boldsymbol{\theta} \in \mathcal{C}} (\mathbf{U} - \boldsymbol{\theta})'\mathbf{V}^{-1}(\mathbf{U} - \boldsymbol{\theta})\end{aligned}$$

and it can be motivated by a geometric argument. The quadratic form in  $Q$  is the  $\mathbf{V}$ -projection of  $\mathbf{U}$  onto the cone  $\mathcal{C}$ . Suppose that this projected point is  $\tilde{\mathbf{U}}$ . If  $\mathbf{U} \in \mathcal{C}$ , then  $Q = 0$  and  $\tilde{\mathbf{U}} = \mathbf{U}$ . If  $\mathbf{U}$  is completely outside of the cone  $\mathcal{C}$ , then  $\tilde{\mathbf{U}}$  is a point on the surface of the cone. Similarly,  $\mathbf{U}'\mathbf{V}^{-1}\mathbf{U}$  is the length of the segment from the origin to  $\mathbf{U}$  in the  $\mathbf{V}$ -space with norm  $\|\mathbf{x}\| = (\mathbf{x}'\mathbf{V}^{-1}\mathbf{x})^{1/2}$ . If

you apply the Pythagorean theorem, you can see that the chi-bar-square statistic measures the length of the segment from the origin to the projected point  $\tilde{\mathbf{U}}$  in  $\mathcal{C}$ .

To calculate  $p$ -values for chi-bar-square statistics, a simulation-based approach is taken. Consider again the set of  $k$  estimable functions  $\mathbf{L}\boldsymbol{\beta}$  with estimate  $\mathbf{L}\hat{\boldsymbol{\beta}} = \mathbf{U}$  and variance  $\mathbf{L}\text{Var}[\hat{\boldsymbol{\beta}}]\mathbf{L}' = \mathbf{V}$ .

First, the observed value of the statistic is computed as

$$\bar{\chi}_{obs}^2 = \mathbf{U}'\mathbf{V}^{-1}\mathbf{U} - Q$$

Then,  $n$  independent random samples  $\mathbf{Z}_1, \dots, \mathbf{Z}_n$  are drawn from an  $N(\mathbf{0}, \mathbf{V})$  distribution and the following chi-bar-statistics are computed for the sample:

$$\begin{aligned} \bar{\chi}_1^2 &= \mathbf{Z}_1'\mathbf{V}^{-1}\mathbf{Z}_1 - \min_{\boldsymbol{\theta} \in \mathcal{C}} (\mathbf{Z}_1 - \boldsymbol{\theta})'\mathbf{V}^{-1}(\mathbf{Z}_1 - \boldsymbol{\theta}) \\ &\vdots \\ \bar{\chi}_n^2 &= \mathbf{Z}_n'\mathbf{V}^{-1}\mathbf{Z}_n - \min_{\boldsymbol{\theta} \in \mathcal{C}} (\mathbf{Z}_n - \boldsymbol{\theta})'\mathbf{V}^{-1}(\mathbf{Z}_n - \boldsymbol{\theta}) \end{aligned}$$

The  $p$ -value is estimated by the fraction of simulated statistics that are greater than or equal to the observed value  $\bar{\chi}_{obs}^2$ .

Notice that unless  $\mathbf{U}$  is interior to the cone  $\mathcal{C}$ , finding the value of  $Q$  requires the solution to a quadratic optimization problem. When  $k$  is large, or when many simulations are requested, the computation of  $p$ -values for chi-bar-square statistics might require considerable computing time.

## ODS Table Names: ESTIMATE Statement

Each table created by the [ESTIMATE](#) statement has a name associated with it, and you can use this name to refer to the table when you use the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in [Table 19.17](#). For more information about ODS, see Chapter 20, “Using the Output Delivery System.”

**Table 19.17** ODS Tables Produced by the [ESTIMATE](#) statement

Table Name	Description	Required Option
Coef	$\mathbf{L}$ matrix coefficients	<a href="#">E</a>
Estimates	<a href="#">ESTIMATE</a> statement results	Default
Contrasts	Joint test results	<a href="#">JOINT</a>

## ODS Graphics: ESTIMATE Statement

This section describes the use of ODS Graphics for creating statistical graphs of the distribution of estimable functions with the **ESTIMATE** statement. The plots can be produced only in association with the PHREG procedure, which can perform Bayesian analysis. The plots are available via these procedures directly, and also via PROC PLM when it is run using an item store that was created by these procedures.

To request these graphs you must do the following:

- specify the ODS GRAPHICS ON statement
- use a BAYES statement with PROC PHREG, or use PROC PLM to perform statistical analysis on an item store that was saved from a Bayesian analysis
- request plots with the **PLOTS=** option in the **ESTIMATE** statement

For more information about the ODS GRAPHICS statement, see Chapter 21, “Statistical Graphics Using ODS.” The available graphs are summarized in Table 19.18.

**Table 19.18** ODS Graphics Produced by the **ESTIMATE** statement

ODS Graph Name	Plot Description	Required Option
BoxPlot	Displays box plots of estimable functions across a posterior sample.	PLOTS=BOXPLOT
DistPanel	Displays panels of histograms with kernel density curves overlaid. Each plot contains the results for the posterior sample of each estimable function.	PLOTS=DISTPLOT
DistPlot	Displays a histogram with a kernel density curve overlaid. The plot contains the results for the posterior sample of the estimable function.	PLOTS=DISTPLOT(UNPACK)

For details about the *plot-options* of the **ESTIMATE** statement, see the **PLOTS=** option in the section “**ESTIMATE Statement**” on page 462.



## LSMEANS Statement

This statement documentation applies to the following procedures:

GENMOD, LOGISTIC, ORTHOREG, PHREG, PLM, SURVEYLOGISTIC, SURVEYPHREG, and SURVEYREG.

The GLIMMIX, GLM, and MIXED procedures also support LSMEANS statements. The relevant statement documentation for these procedures can be found in the specific procedure chapter.

The LSMEANS statement computes least squares means (LS-means) of fixed effects. In the GLM, MIXED, and GLIMMIX procedures, LS-means are *predicted population margins*—that is, they estimate the marginal means over a balanced population. In a sense, LS-means are to unbalanced designs as class and subclass arithmetic means are to balanced designs.

Thus it is important not to interpret the name with a strict association with least squares estimation. Least squares is the predominant estimation technique for the type of models in which LS-means were first applied. Their interpretation and importance reaches beyond the least squares principle, however. A more appropriate approach to LS-means views them as linear combinations of the parameter estimates that are constructed in such a way that they correspond to average predicted values in a population where the levels of classification variables are balanced.

This contemporary—and historically correct—interpretation of the concept of least squares means underlines their importance in all classes of models where predicted values are reasonably formed as linear combinations of the parameter estimates. LS-means distinguish themselves from general estimable functions in that they take the structure for the model and data into account through the structure of the  $\mathbf{X}$  and  $\mathbf{X}'\mathbf{X}$  matrix in your model. For example, in a generalized linear model the structure of the  $\mathbf{X}$  matrix informs the analysis about the possible levels of classification variables and predictions on the linear (the linked) scale are computed as  $\mathbf{x}'\boldsymbol{\beta}$ . LS-means are thus meaningful quantities in such models when the linear estimable function that corresponds to an averaged prediction is constructed on the linked scale. For example, in a binomial model with logit link, the least squares means are predicted population margins of the logits. You can then transform the least squares means to the data scale with the ILINK option, and you can display differences of least squares means in terms of odds ratios with the ODDSRATIO option. The underlying principle—unless you perform a Bayesian analysis—is to construct the estimates or their differences on the linked scale and to apply appropriate transformations in a second step.

Least squares means computations are also supported for multinomial models.

LS-means are computed as  $\mathbf{L}\boldsymbol{\beta}$  where the  $\mathbf{L}$  matrix that is constructed to compute the predicted values is the same as the  $\mathbf{L}$  matrix that is formed in PROC GLM.

Each LS-mean is computed as  $\mathbf{L}\hat{\boldsymbol{\beta}}$ , where  $\mathbf{L}$  is the coefficient matrix that is associated with the least squares mean and  $\hat{\boldsymbol{\beta}}$  is the estimate of the fixed-effects parameter vector. The approximate standard error for the LS-mean is computed as the square root of  $\mathbf{L}\widehat{\text{Var}}[\hat{\boldsymbol{\beta}}]\mathbf{L}'$ . The approximate variance matrix of the fixed-effects estimates depends on the estimation method.

## Syntax: LSMEANS Statement

**LSMEANS** < *model-effects* > < / *options* > ;

LS-means can be computed for any effect in the statistical model that involves only CLASS variables. You can specify multiple effects in one LSMEANS statement or in multiple LSMEANS statements, and all LSMEANS statements must appear after the MODEL statement. If you do not specify *model-effects*, the options in the LSMEANS statement are applied to all suitable model effects.

As in the **ESTIMATE** statement, the **L** matrix is tested for estimability; if this test fails, the procedure displays “Non-est” for the LS-means entries. Note that linear functions of LS-means, such as differences, can be estimable, even if the means themselves are not estimable. Estimability checks for differences are thus applied separately from checks for the means.

Assuming the LS-mean is estimable, the procedure constructs an approximate *t* test to test the null hypothesis that the associated population quantity equals zero.

Table 19.19 summarizes important options in the LSMEANS statement. All LSMEANS options are subsequently discussed in alphabetical order.

**Table 19.19** Important LSMEANS Statement Options

Option	Description
<b>Construction and Computation of LS-Means</b>	
AT	Modifies the covariate value in computing LS-means
BYLEVEL	Computes separate margins
DIFF	Requests differences of LS-means
OM=	Specifies the weighting scheme for LS-means computation as determined by the input data set
SINGULAR=	Tunes estimability checking
<b>Degrees of Freedom and <i>p</i>-values</b>	
ADJUST=	Determines the method for multiple comparison adjustment of LS-means differences
ALPHA= $\alpha$	Determines the confidence level ( $1 - \alpha$ )
STEPDOWN	Adjusts multiple comparison <i>p</i> -values further in a step-down fashion
<b>Statistical Output</b>	
CL	Constructs confidence limits for means and mean differences
CORR	Displays the correlation matrix of LS-means
COV	Displays the covariance matrix of LS-means
E	Prints the <b>L</b> matrix
LINES	Produces a “Lines” display for pairwise LS-means differences
MEANS	Prints the LS-means
PLOTS=	Requests ODS statistical graphics of means and mean comparisons
SEED=	Specifies the seed for computations that depend on random numbers

Table 19.19 *continued*

Option	Description
<b>Generalized Linear Modeling</b>	
<b>EXP</b>	Exponentiates and displays estimates of LS-means or LS-means differences
<b>ILINK</b>	Computes and displays estimates and standard errors of LS-means (but not differences) on the inverse linked scale
<b>ODDSRATIO</b>	Reports (simple) differences of least squares means in terms of odds ratios if permitted by the link function

You can specify the following options in the LSMEANS statement after a slash (/):

#### **ADJDFE=ROW**

#### **ADJDFE=SOURCE**

specifies how denominator degrees of freedom are determined when *p*-values and confidence limits are adjusted for multiple comparisons with the **ADJUST=** option. When you do not specify the **ADJDFE=** option or when you specify **ADJDFE=SOURCE**, the denominator degrees of freedom for multiplicity-adjusted results are the denominator degrees of freedom for the LS-mean effect in the “Type III Tests of Fixed Effects” table. When you specify **ADJDFE=ROW**, the denominator degrees of freedom for multiplicity-adjusted results correspond to the degrees of freedom that are displayed in the DF column of the “Differences of Least Squares Means” table.

The **ADJDFE=ROW** setting is particularly useful if you want multiplicity adjustments to take into account that denominator degrees of freedom are not constant across LS-mean differences.

In one-way models with heterogeneous variance, combining certain **ADJUST=** options with the **ADJDFE=ROW** option corresponds to particular methods of performing multiplicity adjustments in the presence of heteroscedasticity. For example, the following statements fit a heteroscedastic one-way model and perform Dunnett’s T3 method (Dunnett 1980), which is based on the studentized maximum modulus (**ADJUST=SMM**):

```
proc glimmix;
  class A;
  model y = A / ddfm=satterth;
  random _residual_ / group=A;
  lsmeans A / adjust=smm adjdfe=row;
run;
```

If you combine the **ADJDFE=ROW** option with **ADJUST=SIDAK**, the multiplicity adjustment corresponds to the T2 method of Tamhane (1979), and **ADJUST=TUKEY** corresponds to the method of Games-Howell (Games and Howell 1976). Note that **ADJUST=TUKEY** gives the exact results for the case of fractional degrees of freedom in the one-way model, but it does not take into account that the degrees of freedom are subject to variability. A more conservative method, such as **ADJUST=SMM**, might protect the overall error rate better.

Unless the **ADJUST=** option is specified in the LSMEANS statement, the **ADJDFE=** option has no effect. The option is not supported by the procedures that perform chi-square-based inference (GENMOD, LOGISTIC, PHREG, and SURVEYLOGISTIC).

**ADJUST=BON**

**ADJUST=DUNNETT**

**ADJUST=NELSON**

**ADJUST=SCHEFFE**

**ADJUST=SIDAK**

**ADJUST=SIMULATE**< (*simoptions*) >

**ADJUST=SMM** | **GT2**

**ADJUST=TUKEY**

requests a multiple comparison adjustment for the  $p$ -values and confidence limits for the differences of LS-means. The adjusted quantities are produced in addition to the unadjusted quantities. By default, the procedure performs all pairwise differences. If you specify **ADJUST=DUNNETT**, the procedure analyzes all differences with a control level. If you specify **ADJUST=NELSON**, ANOM differences are taken. The **ADJUST=** option implies the **DIFF** option.

The **BON** (Bonferroni) and **SIDAK** adjustments involve correction factors described in Chapter 39, “[The GLM Procedure](#),” and Chapter 58, “[The MULTTEST Procedure](#)”; also see Westfall and Young (1993) and Westfall et al. (1999). When you specify **ADJUST=TUKEY** and your data are unbalanced, the procedure uses the approximation described in Kramer (1956) and identifies the adjustment as “Tukey-Kramer” in the results. Similarly, when you specify **ADJUST=DUNNETT** or **ADJUST=NELSON** and the LS-means are correlated, the procedure uses the factor-analytic covariance approximation described in Hsu (1992) and identifies the adjustment in the results as “Dunnnett-Hsu” or “Nelson-Hsu,” respectively. The approximation derives an approximate “effective sample sizes” for which exact critical values are computed. Computing the exact adjusted  $p$ -values and critical values for unbalanced designs can be computationally intensive, in particular for **ADJUST=NELSON**. A simulation-based approach, as specified by the **ADJUST=SIM** option, while nondeterministic, can provide inferences that are sufficiently accurate in much less time. The preceding references also describe the **SCHEFFE** and **SMM** adjustments.

Nelson’s adjustment applies only to the analysis of means (Ott 1967; Nelson 1982, 1991, 1993), where LS-means are compared against an average LS-mean. It does not apply to all pairwise differences of least squares means. See the **DIFF=ANOM** option for more details regarding the analysis of means with the procedure.

The **SIMULATE** adjustment computes adjusted  $p$ -values and confidence limits from the simulated distribution of the maximum or maximum absolute value of a multivariate  $t$  random vector. All covariance parameters, except the residual scale parameter, are fixed at their estimated values throughout the simulation, potentially resulting in some underdispersion. The simulation estimates  $q$ , the true  $(1-\alpha)$ th quantile, where  $1-\alpha$  is the confidence coefficient. The default  $\alpha$  is 0.05, and you can change this value with the **ALPHA=** option in the LSMEANS statement.

The number of samples is set so that the tail area for the simulated  $q$  is within  $\gamma$  of  $1 - \alpha$  with  $100(1 - \epsilon)\%$  confidence. In equation form,

$$\Pr(|F(\hat{q}) - (1 - \alpha)| \leq \gamma) = 1 - \epsilon$$

where  $\hat{q}$  is the simulated  $q$  and  $F$  is the true distribution function of the maximum; see Edwards and Berry (1987) for details. By default,  $\gamma = 0.005$  and  $\epsilon = 0.01$ , placing the tail area of  $\hat{q}$  within 0.005 of 0.95 with 99% confidence. You can specify the following *simoptions* in parentheses after the ADJUST=SIMULATE option:

<b>ACC=value</b>	specifies the target accuracy radius $\gamma$ of a $100(1 - \epsilon)\%$ confidence interval for the true probability content of the estimated $(1 - \alpha)$ th quantile. The default value is ACC=0.005.
<b>EPS=value</b>	specifies the value $\epsilon$ for a $100 \times (1 - \epsilon)\%$ confidence interval for the true probability content of the estimated $(1 - \alpha)$ th quantile. The default value for the accuracy confidence is 99%, which corresponds to EPS=0.01.
<b>NSAMP=n</b>	specifies the sample size for the simulation. By default, $n$ is set based on the values of the target accuracy radius $\gamma$ and accuracy confidence $100 \times (1 - \epsilon)\%$ for an interval for the true probability content of the estimated $(1 - \alpha)$ th quantile. With the default values for $\gamma$ , $\epsilon$ , and $\alpha$ (0.005, 0.01, and 0.05, respectively), NSAMP=12,604 by default.
<b>SEED=number</b>	specifies an integer that is used to start the pseudo-random number generator for the simulation. If you do not specify a seed, or specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer's clock.
<b>THREADS</b>	specifies that the computational work for the simulation be divided into parallel threads, where the number of threads is the value of the SAS system option CPUCOUNT=. For large simulations (as specified directly using the NSAMP= <i>simoption</i> or indirectly using the ACC= or EPS= <i>simoptions</i> ), parallel processing can markedly speed up the computation of adjusted $p$ -values and confidence intervals. However, because the parallel processing has different pseudo-random number streams, the precise results are different from the default ones, which are computed in sequence rather than in parallel. This option overrides the SAS system option THREADS   NOTHEADS.
<b>NOTHEADS</b>	specifies that the computational work for the simulation be performed in sequence rather than in parallel. NOTHEADS is the default. This option overrides the SAS system option THREADS   NOTHEADS.

If the **STEPDOWN** option is in effect, the  $p$ -values are further adjusted in a step-down fashion. For certain options and data, this adjustment is exact under an *iid*  $N(0, \sigma^2)$  model for the dependent variable, in particular for the following:

- for ADJUST=DUNNETT when the means are uncorrelated
- for ADJUST=TUKEY with STEPDOWN(TYPE=LOGICAL) when the means are balanced and uncorrelated.

The first case is a consequence of the nature of the successive step-down hypotheses for comparisons with a control; the second uses an extension of the maximum studentized range distribution appropriate for partition hypotheses (Royen 1989). Finally, for **STEP-DOWN(TYPE=FREE)**, **ADJUST=TUKEY** employs the Royen (1989) extension in such a way that the resulting *p*-values are conservative.

**ALPHA=number**

requests that a *t* type confidence interval be constructed for each of the LS-means with confidence level  $1 - \text{number}$ . The value of *number* must be between 0 and 1; the default is 0.05.

**AT variable=value**

**AT (variable-list)=(value-list)**

**AT MEANS**

modifies the values of the covariates that are used in computing LS-means. By default, all covariate effects are set equal to their mean values for computation of standard LS-means. The AT option enables you to assign arbitrary values to the covariates. Additional columns in the output table indicate the values of the covariates.

If there is an effect that contains two or more covariates, the AT option sets the effect equal to the product of the individual means rather than the mean of the product (as with standard LS-means calculations). The AT MEANS option sets covariates equal to their mean values (as with standard LS-means) and incorporates this adjustment to crossproducts of covariates.

As an example, consider the following statements:

```
class A;
model Y = A x1 x2 x1*x2;
lsmeans A;
lsmeans A / at means;
lsmeans A / at x1=1.2;
lsmeans A / at (x1 x2)=(1.2 0.3);
```

For the first two LSMEANS statements, the LS-means coefficient for *x1* is  $\bar{x}_1$  (the mean of *x1*) and for *x2* is  $\bar{x}_2$  (the mean of *x2*). However, for the first LSMEANS statement, the coefficient for *x1\*x2* is  $\bar{x}_1\bar{x}_2$ , but for the second LSMEANS statement, the coefficient is  $\bar{x}_1 \times \bar{x}_2$ . The third LSMEANS statement sets the coefficient for *x1* equal to 1.2 and leaves it at  $\bar{x}_2$  for *x2*, and the final LSMEANS statement sets these values to 1.2 and 0.3, respectively.

Even if you specify a **WEIGHT** variable, the unweighted covariate means are used for the covariate coefficients if there is no AT specification. If you specify the AT option, **WEIGHT** or **FREQ** variables are taken into account as follows. The weighted covariate means are then used for the covariate coefficients for which no explicit AT values are given, or if you specify **AT MEANS**. Observations that do not contribute to the analysis because of a missing dependent variable are included in computing the covariate means. Use the **E** option in conjunction with the AT option to check that the modified LS-means coefficients are the ones you want.

The AT option is disabled if you specify the **BYLEVEL** option.

**BYLEVEL**

requests that separate margins be computed for each level of the LSMEANS effect.

The standard LS-means have equal coefficients across classification effects. The BYLEVEL option changes these coefficients to be proportional to the observed margins. This adjustment is reasonable when you want your inferences to apply to a population that is not necessarily balanced but has the margins observed in the input data set. In this case, the resulting LS-means are actually equal to raw means for fixed-effects models and certain balanced random-effects models, but their estimated standard errors account for the covariance structure that you have specified. If a WEIGHT statement is specified, the procedure uses weighted margins to construct the LS-means coefficients.

If the **AT** option is specified, the BYLEVEL option disables it.

**CL**

requests that *t* type confidence limits be constructed for each of the LS-means. The confidence level is 0.95 by default; this can be changed with the **ALPHA=** option. If you specify an **ADJUST=** option, then the confidence limits are adjusted for multiplicity. But if you also specify **STEPDOWN**, then only *p*-values are step-down adjusted, not the confidence limits.

**CORR**

displays the estimated correlation matrix of the least squares means as part of the “Least Squares Means” table.

**COV**

displays the estimated covariance matrix of the least squares means as part of the “Least Squares Means” table.

**DF=number**

specifies the degrees of freedom for the *t* test and confidence limits. The default is the denominator degrees of freedom taken from the “Type III Tests” table that corresponds to the LS-means effect. The option is not supported by the procedures that perform chi-square-based inference (GENMOD, LOGISTIC, PHREG and SURVEYLOGISTIC).

**DIFF<=difftype>****PDIFF<=difftype>**

requests that differences of the LS-means be displayed. You can use one of the following optional *difftype* values to specify which differences to produce:

**ALL** requests all pairwise differences; this is the default.

**ANOM** requests differences between each LS-mean and the average LS-mean, as in the *analysis of means* (Ott 1967). The average is computed as a weighted mean of the LS-means, the weights being inversely proportional to the diagonal entries of the  $\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}'$  matrix. If LS-means are nonestimable, this design-based weighted mean is replaced with an equally weighted mean. Note that the ANOM procedure in SAS/QC software implements both tables and graphics for the analysis of means with a variety of response types. For one-way designs and normal data with identity link, the DIFF=ANOM computations are equivalent to the results of PROC ANOM.



If the LS-means being compared are uncorrelated, exact adjusted  $p$ -values and critical values for confidence limits can be computed in the analysis of means; see Nelson (1982, 1991, 1993) and Guirguis and Tobias (2004) in addition to the documentation for the [ADJUST=NELSON](#) option.

### **CONTROL**

requests differences with a control, which, by default, is the first level of each of the specified LSMEANS effects. To specify which levels of the effects are the controls, list the quoted formatted values in parentheses after the CONTROL keyword. For example, if the effects A, B, and C are classification variables, each having two levels, 1 and 2, the following LSMEANS statement specifies the (1,2) level of A\*B and the (2,1) level of B\*C as controls:

```
lsmeans A*B B*C / diff=control('1' '2' '2' '1');
```

For multiple effects, the results depend upon the order of the list, and so you should check the output to make sure that the controls are correct.

Two-tailed tests and confidence limits are associated with the CONTROL *diff*type. For one-tailed results, use either the CONTROLL or CONTROLU *diff*type.

### **CONTROLL**

tests whether the noncontrol levels are significantly smaller than the control; the upper confidence limits for the control minus the noncontrol levels are considered to be infinity and are displayed as missing.

### **CONTROLU**

tests whether the noncontrol levels are significantly larger than the control; the upper confidence limits for the noncontrol levels minus the control are considered to be infinity and are displayed as missing.

If you want to perform multiple comparison adjustments on the differences of LS-means, you must specify the [ADJUST=](#) option.

The differences of the LS-means are displayed in a table titled “Differences of Least Squares Means.”

## **E**

requests that the **L** matrix coefficients for the LSMEANS effects be displayed.

## **EXP**

requests exponentiation of the LS-means or LS-mean differences. When you model data with the logit, cumulative logit, or generalized logit link functions, and the estimate represents a log odds ratio or log cumulative odds ratio, the EXP option produces an odds ratio. In proportional hazards model, the exponentiation of the LS-mean differences produces estimates of hazard ratios. If you specify the [CL](#) or [ALPHA=](#) option, the (adjusted) confidence bounds are also exponentiated.

The EXP option is supported only by PROC PHREG, PROC SURVEYPHREG, the procedures that support generalized linear modeling (GENMOD, LOGISTIC, and SURVEYLOGISTIC), and PROC PLM when it is used to perform statistical analyses on item stores that are created by these procedures.



**ILINK**

requests that estimates and their standard errors in the “Least Squares Means” table also be reported on the scale of the mean (the inverse linked scale). This enables you to obtain estimates of predicted probabilities and their standard errors in logistic models, for example. The option is specific to an LSMEANS statement. If you also specify the **CL** option, the procedure computes confidence intervals for the predicted means by applying the inverse link transform to the confidence limits on the linked (linear) scale. Standard errors on the inverse linked scale are computed by the delta method.

The ILINK option is supported only by the procedures that support generalized linear modeling (GENMOD, LOGISTIC and SURVEYLOGISTIC) and by PROC PLM when it is used to perform statistical analyses on item stores that are created by these procedures.

**LINES**

presents results of comparisons between all pairs of least squares means by listing the means in descending order and indicating nonsignificant subsets by line segments beside the corresponding LS-means. When all differences have the same variance, these comparison lines are guaranteed to accurately reflect the inferences that are based on the corresponding tests, which are made by comparing the respective  $p$ -values to the value of the **ALPHA=** option (0.05 by default). However, equal variances might not be the case for differences between LS-means. If the variances are not all the same, then the comparison lines might be conservative, in the sense that if you base your inferences on the lines alone, you will detect fewer significant differences than the tests indicate. If there are any such differences, the procedure lists the pairs of means that are inferred to be significantly different by the tests but not by the comparison lines. However, even though the variances in many cases are unequal, they are similar enough that the comparison lines accurately reflect the test inferences.

**MEANS | NOMEANS**

determines whether to print the least squares means themselves. For most procedure, MEANS is the default behavior. For example, the NOMEANS option is the default for the PHREG procedure. You can then use the MEANS option to produce the table of least squares means, if desired.

**ODDSRATIO****OR**

requests that LS-mean differences (**DIFF**, **ADJUST=** options) are also reported in terms of odds ratios. The ODDSRATIO option is ignored unless you use either the logit, cumulative logit, or generalized logit link function. If you specify the **CL** or **ALPHA=** option, confidence intervals for the odds ratios are also computed. These intervals are adjusted for multiplicity when you specify the **ADJUST=** option.

The ODDSRATIO option is supported only by the procedures that support generalized linear modeling (GENMOD, LOGISTIC and SURVEYLOGISTIC) and by PROC PLM when it is used to perform statistical analyses on item stores created by these procedures.

**OBSMARGINS** <=*OM-data-set*>

**OM** <=*OM-data-set*>

specifies a potentially different weighting scheme for the computation of LS-means coefficients. The standard LS-means have equal coefficients across classification effects; however, the OM option changes these coefficients to be proportional to those found in the *OM-data-set*. This adjustment is reasonable when you want your inferences to apply to a population that is not necessarily balanced but has the margins that are observed in *OM-data-set*.

By default, *OM-data-set* is the same as the analysis data set. You can optionally specify another data set that describes the population for which you want to make inferences. This data set must contain all model variables except for the dependent variable (which is ignored if it is present). In addition, the levels of all CLASS variables must be the same as those that occur in the analysis data set. Specifying an *OM-data-set* enables you to construct arbitrarily weighted LS-means.

In computing the observed margins, the procedure uses all observations for which there are no missing or invalid independent variables, including those for which there are missing dependent variables. Also, if you use a WEIGHT statement, the procedure computes weighted margins to construct the LS-means coefficients. If your data are balanced, the LS-means are unchanged by the OM option.

The **BYLEVEL** option modifies the observed-margins LS-means. Instead of computing the margins across all of the *OM-data-set*, the procedure computes separate margins for each level of the LSMEANS effect in question. In this case the resulting LS-means are actually equal to raw means for fixed-effects models and certain balanced random-effects models, but their estimated standard errors account for the covariance structure that you have specified.

You can use the **E** option in conjunction with either the OM or **BYLEVEL** option to verify that the modified LS-means coefficients are the ones you want. It is possible that the modified LS-means are not estimable when the standard ones are estimable, or vice versa.

**PDIF**

is the same as the **DIFF** option.

**PLOT | PLOTS** <=*plot-request*<(options)>>

**PLOT | PLOTS** <=(*plot-request*<(options)> <...*plot-request*<(options)> >)>

requests that graphics related to least squares means be produced via ODS Graphics, provided that the ODS GRAPHICS ON statement has been specified and the *plot-request* does not conflict with other options in the LSMEANS statement. For general information about ODS Graphics, see Chapter 21, “Statistical Graphics Using ODS.”

The available options and suboptions are as follows:

**ALL**

requests that the default plots that correspond to this LSMEANS statement be produced. The default plot depends on the options in the statement.

**ANOMPLOT****ANOM**

requests an analysis-of-means display in which least squares means are compared to an average least squares mean. Least squares mean ANOM plots are produced only for those model effects that are listed in LSMEANS statements and have options that do not contradict with the display. For example, the following statements produce analysis-of-mean plots for effects A and C:

```
lsmeans A / diff=anom plot=anom;
lsmeans B / diff      plot=anom;
lsmeans C /           plot=anom;
```

The **DIFF** option in the second LSMEANS statement implies all pairwise differences.

**BOXPLOT**< *boxplot-options* >

produces box plots of the distribution of the least squares mean or least squares mean differences across a posterior sample. For example, this plot is available in procedures that support a Bayesian analysis through the BAYES statement.

A separate box is generated for each estimable function, and all boxes appear on a single graph by default. You can affect the appearance of the box plot graph with the following options:

**ORIENTATION=VERTICAL | HORIZONTAL**

**ORIENT=VERT | HORIZ** specifies the orientation of the boxes. The default is vertical orientation of the box plots.

**NPANELPOS=number** specifies how to break the series of box plots across multiple panels. If the NPANELPOS option is not specified, or if *number* equals zero, then all box plots are displayed in a single graph; this is the default. If a negative number is specified, then exactly up to  $|number|$  of box plots are displayed per panel. If *number* is positive, then the number of boxes per panel is balanced to achieve small variation in the number of box plots per graph.

**CONTROLPLOT****CONTROL**

requests a display in which least squares means are visually compared against a reference level. These plots are produced only for statements with options that are compatible with control differences. For example, the following statements produce control plots for effects A and C:

```
lsmeans A / diff=control('1') plot=control;
lsmeans B / diff      plot=control;
lsmeans C              plot=control;
```

The **DIFF** option in the second LSMEANS statement implies all pairwise differences.

**DIFFPLOT**< (*diffplot-options*) >**DIFFOGRAM**< (*diffplot-options*) >**DIFF**< (*diffplot-options*) >

requests a display of all pairwise least squares mean differences and their significance. The display is also known as a “mean-mean scatter plot” when it is based on arithmetic means (Hsu 1996 and Hsu and Peruggia 1994). For each comparison a line segment, centered at the LS-means in the pair, is drawn. The length of the segment corresponds to the projected width of a confidence interval for the least squares mean difference. Segments that fail to cross the 45-degree reference line correspond to significant least squares mean differences.

LS-mean difference plots are produced only for statements with options that are compatible with the display. For example, the following statements request differences against a control level for the A effect, all pairwise differences for the B effect, and the least squares means for the C effect:

```
lsmeans A / diff=control('1') plot=diff;
lsmeans B / diff                plot=diff;
lsmeans C                      plot=diff;
```

The **DIFF=** type in the first statement is incompatible with a display of all pairwise differences.

You can specify the following *diffplot-options*:

<b>ABS</b>	determines the positioning of the line segments in the plot. This is the default <i>diffplot-options</i> . When the ABS option is in effect, all line segments are shown on the same side of the reference line.
<b>NOABS</b>	determines the positioning of the line segments in the plot. The NOABS option separates comparisons according to the sign of the difference.
<b>CENTER</b>	marks the center point for each comparison. This point corresponds to the intersection of two least squares means.
<b>NOLINE</b>	suppresses the display of the line segments that represent the confidence bounds for the differences of the least squares means. The NOLINES option implies the CENTER option. The default is to draw line segments in the upper portion of the plot area without marking the center point.

**DISTPLOT**< *distplot-options* >**DIST**< *distplot-options* >

generates panels of histograms with a kernel density overlaid if the analysis has access to a set of posterior parameter estimates. For example, this plot is available in procedures that support a Bayesian analysis through the BAYES statement. A separate plot in each panel contains the results for each least squares mean or least squares mean differences. You can specify the following *distplot-options* in parentheses:

- BOX | NOBOX** controls the display of a horizontal box plot of the estimable function's distribution across the posterior sample below the graph. The BOX option is enabled by default.
- HIST | NOHIST** controls the display of the histogram of the estimable function's distribution across the posterior sample. The HIST option is enabled by default.
- NORMAL | NONORMAL** controls the display of a normal density estimate on the graph. The NONORMAL option is enabled by default.
- KERNEL | NOKERNEL** controls the display of a kernel density estimate on the graph. The KERNEL option is enabled by default.
- NROWS=number** specifies the highest number of rows in a panel. The default is 3.
- NCOLS=number** specifies the highest number of columns in a panel. The default is 3.
- UNPACK** unpacks the panel into separate graphics.

#### **MEANPLOT**<(meanplot-options)>

requests displays of the least squares means.

The following *meanplot-options* control the display of the least squares means.

#### **ASCENDING**

displays the least squares means in ascending order. This option has no effect if means are displayed in separate plots.

#### **CL**

displays upper and lower confidence limits for the least squares means. By default, 95% limits are drawn. You can change the confidence level with the [ALPHA=](#) option. Confidence limits are drawn by default if the [CL](#) option is specified in the LSMEANS statement.

#### **CLBAND**

displays confidence limits as bands. This option implies the JOIN option.

#### **DESCENDING**

displays the least squares means in descending order. This option has no effect if means are displayed in separate plots.

#### **ILINK**

requests that means (and confidence limits) be displayed on the inverse linked scale.

#### **JOIN**

#### **CONNECT**

connects the least squares means with lines. This option is implied by the CLBAND option. If the effect contains nested variables and a SLICEBY= effect contains classification variables that appear as crossed effects, this option is ignored.

**SLICEBY=***fixed-effect*

specifies an effect by which to group the means in a single plot. For example, the following statement requests a plot in which the levels of A are placed on the horizontal axis and the means that belong to the same level of B are joined by lines:

```
lsmeans A*B / plot=meanplot(sliceby=b join);
```

Unless the LS-mean effect contains at least two classification variables, the SLICEBY= option has no effect. The SLICEBY= effect does not have to be an effect in your MODEL statement, but it must consist entirely of classification variables.

**PLOTBY=***fixed-effect*

specifies an effect by which to break interaction plots into separate displays. For example, the following statement requests for each level of C one plot of the A\*B cell means that are associated with that level of C:

```
lsmeans A*B*C / plot=meanplot(sliceby=b plotby=c clband);
```

In each plot, levels of A are displayed on the horizontal axis, and confidence bands are drawn around the means that share the same level of B.

The PLOTBY= option has no effect unless the LS-mean effect contains at least three classification variables. The PLOTBY= effect does not have to be an effect in the MODEL statement, but it must consist entirely of classification variables.

**NONE**

requests that no plots be produced.

When LS-mean calculations are adjusted for multiplicity by using the **ADJUST=** option, the plots are adjusted accordingly.

**SEED=***number*

specifies the seed for the sampling-based components of the computations for the LSMEANS statement (for example, chi-bar-square statistics and simulated *p*-values). *number* specifies an integer that is used to start the pseudo-random-number generator for the simulation. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock. Note that there could be multiple LSMEANS statements with SEED= specifications and there could be other statements that can supply a random number seed. Since the procedure has only one random number stream, the initial seed is shown in the SAS log.

**SINGULAR=***number*

tunes the estimability checking. If  $\mathbf{v}$  is a vector, define  $\text{ABS}(\mathbf{v})$  to be the largest absolute value of the elements of  $\mathbf{v}$ . If  $\text{ABS}(\mathbf{K}' - \mathbf{K}'\mathbf{T})$  is greater than  $c*\text{number}$  for any row of  $\mathbf{K}'$  in the contrast, then  $\mathbf{K}'\boldsymbol{\beta}$  is declared nonestimable. Here,  $\mathbf{T}$  is the Hermite form matrix  $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}$ , and  $c$  is  $\text{ABS}(\mathbf{K}')$ , except when it equals 0, and then  $c$  is 1. The value for *number* must be between 0 and 1; the default is 1E-4.

**STEPDOWN**<(step-down-options)>

requests that multiple comparison adjustments for the  $p$ -values of LS-mean differences be further adjusted in a step-down fashion. Step-down methods increase the power of multiple comparisons by taking advantage of the fact that a  $p$ -value is never declared significant unless all smaller  $p$ -values are also declared significant. The STEPDOWN adjustment combined with **ADJUST=BON** corresponds to the methods of Holm (1979) “Method 2” of Schaffer (1986); this is the default. Using step-down-adjusted  $p$ -values combined with **ADJUST=SIMULATE** corresponds to the method of Westfall (1997).

If the denominator degrees of freedom are computed by the Kenward-Roger (Kenward and Roger 1997) or Satterthwaite method in a mixed model, then step-down-adjusted  $p$ -values are produced only if the **ADJDFE=ROW** option is in effect.

Also, STEPDOWN affects only  $p$ -values, not confidence limits. For **ADJUST=SIMULATE**, the generalized least squares hybrid approach of Westfall (1997) is used to increase Monte Carlo accuracy.

You can specify the following *step-down-options* in parentheses:

**MAXTIME**= $n$ 

specifies the time (in seconds) to be spent computing the maximal logically consistent sequential subsets of equality hypotheses for **TYPE=LOGICAL**. The default is **MAXTIME=60**. If the **MAXTIME** value is exceeded, the adjusted tests are not computed. When this occurs, you can try increasing the **MAXTIME** value. However, note that there are common multiple comparisons problems for which this computation requires a huge amount of time—for example, all pairwise comparisons between more than 10 groups. In such cases, try to use **TYPE=FREE** (the default) or **TYPE=LOGICAL( $n$ )** for small  $n$ .

**REPORT**

specifies that a report on the step-down adjustment be displayed, including a listing of the sequential subsets (Westfall 1997) and, for **ADJUST=SIMULATE**, the step-down simulation results.

**TYPE=LOGICAL**<( $n$ )>**TYPE=FREE**

specifies how step-down adjustment are made. If you specify **TYPE=LOGICAL**, the step-down adjustments are computed by using maximal logically consistent sequential subsets of equality hypotheses (Shaffer 1986, Westfall 1997). Alternatively, for **TYPE=FREE**, sequential subsets are computed ignoring logical constraints. The **TYPE=FREE** results are more conservative than those for **TYPE=LOGICAL**, but they can be much more efficient to produce for many comparisons. For example, it is not feasible to take logical constraints between all pairwise comparisons of more than 10 groups. For this reason, **TYPE=FREE** is the default.

However, you can reduce the computational complexity of taking logical constraints into account by limiting the depth of the search tree used to compute them, specifying the optional depth parameter as a number  $n$  in parentheses after **TYPE=LOGICAL**. As with **TYPE=FREE**, results for **TYPE=LOGICAL( $n$ )** are conservative relative to the true **TYPE=LOGICAL** results. But even for **TYPE=LOGICAL(0)** they can be appreciably less conservative than **TYPE=FREE**, and they are computationally feasible for much larger numbers of comparisons. If you do not specify  $n$  or if  $n = -1$ , the full search tree is used.



## ODS Table Names: LSMEANS Statement

Each table created by the **LSMEANS** statement has a name associated with it, and you can use this name to refer to the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in [Table 19.20](#). For more information about ODS, see Chapter 20, “[Using the Output Delivery System](#).”

**Table 19.20** ODS Tables Produced by the **LSMEANS** statement

Table Name	Description	Required Option
Coef	L matrix coefficients	E
Diffs	Differences of LS-means	DIFF or ADJUST= or STEPDOWN
LSMeans	LS-means	Default
LSMLines	Lines display for LS-means	LINES

## ODS Graphics: LSMEANS Statement

This section describes the use of ODS Graphics for creating graphics that are related to LS-means in procedures that support the common **LSMEANS** or **SLICE** statement. There are two groups of available plots: those that can be produced by all procedures that support these two statements, and those that can be produced only in association with the two procedures that can perform Bayesian analysis (PROC GENMOD and PROC PHREG). Plots that are associated with the Bayesian analysis are available via these procedures directly, and also by using PROC PLM with an item store that was created by these procedures.

Plots in the first group depict the LS-means and their differences; when LS-mean comparisons are adjusted for multiplicity by using the **ADJUST=** option, the plots are adjusted accordingly. To request plots in this group, you must specify the ODS GRAPHICS ON statement and request plots with the appropriate **PLOTS=** option in the **LSMEANS** or **SLICE** statement. Plots in the second group depict the posterior sample distribution of LS-means and their differences. To request plots in this group, you must also use a BAYES statement with PROC GENMOD or PROC PHREG, or use PROC PLM to perform statistical analysis on an item store that was saved from a Bayesian analysis.

For more information about the ODS GRAPHICS statement, see Chapter 21, “[Statistical Graphics Using ODS](#).” The available graphs are summarized in [Table 19.21](#) and [Table 19.22](#).

**Table 19.21** ODS Graphics Produced by All Procedures That Support the Common **LSMEANS** or **SLICE** Statement

ODS Graph Name	Plot Description	Required Option
AnomPlot	Requests an analysis of means display in which least squares means are compared to an average least squares mean.	PLOTS=ANOM



**Table 19.21** *continued*

ODS Graph Name	Plot Description	Required Option
ControlPlot	Requests a display in which least squares means are compared to a reference level.	PLOTS=CONTORL
DiffPlot	Displays all pairwise least squares mean differences and their significance. This plot is also known as a “mean-mean scatter plot” when based on arithmetic means.	PLOTS=DIFF
MeanPlot	Displays least squares means.	PLOTS=MEANPLOT

**Table 19.22** ODS Graphics Produced by Procedures That Support the LSMEANS or SLICE Statement and Bayesian Analysis

ODS Graph Name	Plot Description	Required Option
BoxPlot	Displays box plots of LS-means or LS-mean differences across a posterior sample.	PLOTS=BOXPLOT
DistPanel	Displays panels of histograms with kernel density curves overlaid. Each plot contains the results for the posterior sample of each LS-mean or LS-mean difference.	PLOTS=DISTPLOT
DistPlot	Displays a histogram with a kernel density curve overlaid. The plot contains the results for the posterior sample of the LS-mean or LS-mean difference.	PLOTS=DISTPLOT(UNPACK)

You can supply the same *plot-options* to the SLICE statement to produce these graphs. For details about the *plot-options* of the LSMEANS or SLICE statement, see the PLOTS= option in the section “LSMEANS Statement” on page 479. For more details about the DIFFPLOT in particular, see the section “Graphics for LS-Mean Comparisons” on page 2843 in Chapter 38, “The GLIMMIX Procedure.”

## LSMESTIMATE Statement

This statement documentation applies to the following procedures:

GENMOD, LOGISTIC, MIXED, ORTHOREG, PHREG, PLM, SURVEYLOGISTIC, SURVEYPHREG, and SURVEYREG.

The LSMESTIMATE statement in the GLIMMIX procedure is documented in Chapter 38, “[The GLIMMIX Procedure](#).”

The LSMESTIMATE statement provides a mechanism for obtaining custom hypothesis tests among least squares means. In contrast to the [LSMEANS](#) statement, the LSMESTIMATE statement does not produce the least squares means or their differences; instead, you can estimate any linear function of the least squares means (including the means themselves or their differences). In contrast to the linear functions that are constructed with the [ESTIMATE](#) statement, you do not specify coefficients for the individual parameter estimates. Instead, with the LSMESTIMATE statement you specify coefficients for the least squares means; these are then converted for you into estimable functions for the parameter estimates.

The LSMESTIMATE statement thus combines important and convenient features of the [LSMEANS](#) and the [ESTIMATE](#) statement. As with the [LSMEANS](#) statement, the following conditions are true:

- You need to specify only a single effect; the mapping into linear estimable functions in terms of the parameter estimates is performed by the procedure.
- You can use the [AT=](#), [BYLEVEL](#), and [OBSMARGINS](#) options to affect the computation of the underlying least squares means.

As with the [ESTIMATE](#) statement you can do the following:

- specify multiple-row linear combinations.
- perform multiplicity corrections to control the familywise Type I error probability with the [ADJUST=](#) option.
- construct general linear functions of the least squares means.
- perform joint  $F$  or chi-square tests with or without order restrictions through the [JOINT](#) option.
- rely on positional or nonpositional syntax to specify coefficients for linear functions. For details about using nonpositional syntax, see the section “[Positional and Nonpositional Syntax for Coefficients in Linear Functions](#)” on page 473.

The computation of an LSMESTIMATE involves two coefficient matrices. Suppose that there are  $n_l$  levels for a valid least squares means effect (an effect that is part of your model and consists of classification variables only). Then the LS-means are formed as  $\mathbf{L}_1 \hat{\boldsymbol{\beta}}$ , where  $\mathbf{L}_1$  is a  $(n_l \times p)$  coefficient matrix. The  $(k \times n_l)$  coefficient matrix  $\mathbf{K}$  is formed from the *values* that you supply in

the  $k$  rows of the LSMESTIMATE statement. The least squares means estimates then represent the  $(k \times 1)$  vector

$$\mathbf{KL}_1\boldsymbol{\beta} = \mathbf{L}\boldsymbol{\beta}$$

Because the analytic features and capabilities of the LSMESTIMATE statement are an amalgam of the LSMEANS and the ESTIMATE statement, the syntax of the statement follows the same pattern.

## Syntax: LSMESTIMATE Statement

```
LSMESTIMATE model-effect < 'label' > values < divisor=n >
            < , < 'label' > values < divisor=n > > < , ... >
            < / options > ;
```

In contrast to a multirow estimate in the ESTIMATE statement, you specify only a single effect in the LSMESTIMATE statement. The row labels are optional and follow the *model-effect* specification. For example, the following statements fit a split-split-plot design and compare the average of the third and fourth LS-mean of the whole-plot factor A to the first LS-mean of the factor:

```
proc glimmix;
  class a b block;
  model y = a b a*b / s;
  random int a / sub=block;
  lsestimate A 'a1 vs avg(a3,a4)' 2 0 -1 -1 divisor=2;
run;
```

The order in which coefficients are assigned to the least squares means corresponds to the order in which they are displayed in the “Least Squares Means” table. You can use the ELSM option to see how coefficients are matched to levels of the fixed effect.

The optional *divisor=n* specification enables you to assign a separate divisor to each row of the LSMESTIMATE. You can also assign divisor values through the DIVISOR= option. See the description of the DIVISOR= option that follows for the interaction between the two ways of specifying divisors.

Table 19.23 summarizes important options in the LSMESTIMATE statement. All LSMESTIMATE options are subsequently discussed in alphabetical order.

**Table 19.23** Important LSMESTIMATE Statement Options

Option	Description
<b>Construction and Computation of LS-Means</b>	
AT	Modifies covariate values in computing LS-means
BYLEVEL	Computes separate margins
DIVISOR=	Specifies a list of values to divide the coefficients
OM=	Specifies the weighting scheme for LS-means computation as determined by a data set
SINGULAR=	Tunes estimability checking

**Table 19.23** *continued*

Option	Description
<b>Degrees of Freedom and <math>p</math>-values</b>	
ADJUST=	Determines the method for multiple comparison adjustment of LS-means differences
ALPHA= $\alpha$	Determines the confidence level ( $1 - \alpha$ )
LOWER	Performs one-sided, lower-tailed inference
STEPDOWN	Adjusts multiple comparison $p$ -values further in a step-down fashion
TESTVALUE=	Specifies values under the null hypothesis for tests
UPPER	Performs one-sided, upper-tailed inference
<b>Statistical Output</b>	
CL	Constructs confidence limits for means and mean differences
CORR	Displays the correlation matrix of LS-means
COV	Displays the covariance matrix of LS-means
E	Prints the <b>L</b> matrix
ELSM	Prints the <b>K</b> matrix
JOINT	Produces a joint $F$ or chi-square test for the LS-means and LS-means differences
PLOTS=	Requests ODS statistical graphics of means and mean comparisons
SEED=	Specifies the seed for computations that depend on random numbers
<b>Generalized Linear Modeling</b>	
CATEGORY=	Specifies how to construct estimable functions with multinomial data
EXP	Exponentiates and displays LS-means estimates
ILINK	Computes and displays estimates and standard errors of LS-means (but not differences) on the inverse linked scale

You can specify the following options in the LSMESTIMATE statement after a slash (/):

**ADJDFE=SOURCE****ADJDFE=ROW**

specifies how denominator degrees of freedom are determined when  $p$ -values and confidence limits are adjusted for multiple comparisons with the ADJUST= option. When you do not specify the ADJDFE= option or when you specify ADJDFE=SOURCE, the denominator degrees of freedom for multiplicity-adjusted results are the denominator degrees of freedom for the LS-mean effect in the “Type III Tests of Fixed Effects” table.

The ADJDFE=ROW setting is useful if you want multiplicity adjustments to take into account that denominator degrees of freedom are not constant across estimates. For example, this can be the case when the denominator degrees of freedom are computed by the Satterthwaite or Kenward-Roger method (Kenward and Roger 1997) in a mixed model.

The `ADJDFE=` option is not supported by the procedures that perform chi-square-based inference (`GENMOD`, `LOGISTIC`, `PHREG` and `SURVEYLOGISTIC`).

**ADJUST=BON**

**ADJUST=SCHEFFE**

**ADJUST=SIDAK**

**ADJUST=SIMULATE** < (*simoptions*) >

**ADJUST=T**

requests a multiple comparison adjustment for the  $p$ -values and confidence limits for the LS-mean estimates. The adjusted quantities are produced in addition to the unadjusted  $p$ -values and confidence limits. Adjusted confidence limits are produced if the `CL` or `ALPHA=` option is in effect. For a description of the adjustments, see Chapter 39, “[The GLM Procedure](#),” and Chapter 58, “[The MULTTEST Procedure](#),” in addition to the documentation for the `ADJUST=` option in the `LSMEANS` statement.

Not all adjustment methods of the `LSMEANS` statement are available for the `LSMESTIMATE` statement. Multiplicity adjustments in the `LSMEANS` statement are designed specifically for differences of least squares means.

If you specify the `STEPDOWN` option, the  $p$ -values are further adjusted in a step-down fashion.

**ALPHA=number**

requests that a  $t$  type confidence interval be constructed for each of the LS-means with confidence level  $1 - \text{number}$ . The value of *number* must be between 0 and 1; the default is 0.05.

**AT variable=value**

**AT (variable-list)=(value-list)**

**AT MEANS**

modifies the values of the covariates used in computing LS-means. See the `AT` option in the `LSMEANS` statement for details.

**BYLEVEL**

requests that the procedure compute separate margins for each level of the `LSMEANS` effect.

The standard LS-means have equal coefficients across classification effects. The `BYLEVEL` option changes these coefficients to be proportional to the observed margins. This adjustment is reasonable when you want your inferences to apply to a population that is not necessarily balanced but has the margins observed in the input data set. In this case, the resulting LS-means are actually equal to raw means for fixed-effects models and certain balanced random-effects models, but their estimated standard errors account for the covariance structure that you have specified. If a `WEIGHT` statement is specified, the procedure uses weighted margins to construct the LS-means coefficients.

If the `AT` option is specified, the `BYLEVEL` option disables it.

**CATEGORY=category-options**

specifies how to construct estimates and multiplicity corrections for models with multinomial data (ordinal or nominal). This option is also important for constructing sets of estimable functions for  $F$  tests with the **JOINT** option.

The *category-options* indicate how response variable levels are treated in constructing the estimable functions. Possible value for the *category-options* are the following:

**JOINT**

computes the estimable functions for every nonredundant category and treats them as a set. For example, a three-row LSMESTIMATE statement in a model with three response categories leads to six estimable functions.

**SEPARATE**

computes the estimable functions for every nonredundant category in turn. For example, a three-row LSMESTIMATE statement in a model with three response categories leads to two sets of three estimable functions.

*quoted-value-list*

computes the estimable functions only for the list of values given. The list must consist of formatted values of the response categories.

For further details about using the CATEGORY= option in models for multinomial data, see the documentation for the **CATEGORY=** option in the **ESTIMATE** statement.

The CATEGORY= option is supported only by the procedures that support generalized linear modeling (GENMOD, LOGISTIC, and SURVEYLOGISTIC) and by PROC PLM when it is used to perform statistical analyses on item stores that were created by these procedures.

**CHISQ**

requests that chi-square tests be performed in addition to  $F$  tests, when you request an  $F$  test with the **JOINT** option. This option has no effect in procedures that produce chi-square statistics by default.

**CL**

requests that  $t$  type confidence limits be constructed for each of the LS-means. The confidence level is 0.95 by default; this can be changed with the **ALPHA=** option. If you specify an **ADJUST=** option, then the confidence limits are adjusted for multiplicity. But if you also specify **STEPDOWN**, then only  $p$ -values are step-down adjusted, not the confidence limits.

**CORR**

displays the estimated correlation matrix of the linear combination of the least squares means.

**COV**

displays the estimated covariance matrix of the linear combination of the least squares means.

**DF=number**

specifies the degrees of freedom for the tests and confidence limits. The option is not supported by the procedures that perform chi-square-based inference (GENMOD, LOGISTIC, PHREG, and SURVEYLOGISTIC).

**DIVISOR=value-list**

specifies a list of values by which to divide the coefficients so that fractional coefficients can be entered as integer numerators. If you do not specify *value-list*, a default value of 1.0 is assumed. Missing values in the *value-list* are converted to 1.0.

If the number of elements in *value-list* exceeds the number of rows of the estimate, the extra values are ignored. If the number of elements in *value-list* is less than the number of rows of the estimate, the last value in *value-list* is carried forward.

If you specify a row-specific divisor as part of the specification of the estimate row, this value multiplies the corresponding value in the *value-list*. For example, the following statement divides the coefficients in the first row by 8, and the coefficients in the third and fourth row by 3:

```
lsmestimate A 'One vs. two' 8 -8 divisor=2,
              'One vs. three' 1 0 -1
              'One vs. four' 3 0 0 -3
              'One vs. five' 3 0 0 0 -3 / divisor=4,.,3;
```

Coefficients in the second row are not altered.

**E**

requests that the **L** coefficients of the estimable function be displayed. These are the coefficients that apply to the fixed-effect parameter estimates. The E option displays the coefficients that you would need to enter in an equivalent [ESTIMATE](#) statement.

**ELSM**

requests that the **K** matrix coefficients be displayed. These are the coefficients that apply to the LS-means. This option is useful to ensure that you assigned the coefficients correctly to the LS-means.

**EXP**

requests exponentiation of the least squares means estimate. When you model data with the logit link function and the estimate represents a log odds ratio, the EXP option produces an odds ratio. If you specify the [CL](#) or [ALPHA=](#) option, the (adjusted) confidence limits for the estimate are also exponentiated.

The EXP option is supported only by PROC PHREG, PROC SURVEYPHREG, the procedures that support generalized linear modeling (GENMOD, LOGISTIC, and SURVEYLOGISTIC), and by PROC PLM when it is used to perform statistical analyses on item stores that were created by these procedures.

**ILINK**

requests that the estimate and its standard error also be reported on the scale of the mean (the inverse linked scale). The computation of the inverse linked estimate depends on the estimation mode. For example, if the analysis is based on a posterior sample when a BAYES statement is present, the inversely linked estimate is the average of the inversely linked values across the sample of posterior parameter estimates. If the analysis is not based on a sample of parameter estimates, the procedure computes the value on the mean scale by applying the inverse link to the estimate.

The interpretation of the inversely linked quantity depends on the coefficients that are specified in your LSMESTIMATE statement and the link function. For example, in a model for binary data with logit link the following LSMESTIMATE statement computes

$$q = \frac{1}{1 + \exp\{-(\tau_1 - \tau_2)\}}$$

where  $\tau_1$  and  $\tau_2$  are the least squares means that are associated with the first two levels of the classification effect A:

```
proc logistic;
  class A / param=glm;
  model y = A / dist=binary link=logit;
  lsestimate A 1 -1 / ilink;
run;
```

The quantity  $q$  is not the difference of the probabilities associated with the two levels,

$$\pi_1 - \pi_2 = \frac{1}{1 + \exp\{-\tau_1\}} - \frac{1}{1 + \exp\{-\tau_2\}}$$

The standard error of the inversely linked estimate is based on the delta method. If you also specify the **CL** or **ALPHA=** option, the procedure computes confidence intervals for the inversely linked estimate. These intervals are obtained by applying the inverse link to the confidence intervals on the linked scale.

The **ILINK** option is supported only by the procedures that support generalized linear modeling (**GENMOD**, **LOGISTIC**, and **SURVEYLOGISTIC**) and by **PROC PLM** when it is used to perform statistical analyses on item stores that were created by these procedures.

#### **JOINT**<(joint-test-options)>

requests that a joint  $F$  or chi-square test be produced for the rows of the estimate. For more information about the simulation-based  $p$ -value calculation, see the section “[Joint Hypothesis Tests with Complex Alternatives, the Chi-Bar-Square Statistic](#)” on page 476. You can specify the following *joint-test-options* in parentheses:

##### **ACC**= $\gamma$

specifies the accuracy radius for determining the necessary sample size in the simulation-based approach of Silvapulle and Sen (2004) for tests with order restrictions. The value of  $\gamma$  must be strictly between 0 and 1; the default value is 0.005.

##### **EPS**= $\epsilon$

specifies the accuracy confidence level for determining the necessary sample size in the simulation-based approach of Silvapulle and Sen (2004) for  $F$  tests with order restrictions. The value of  $\epsilon$  must be strictly between 0 and 1; the default value is 0.01.

##### **LABEL**=‘label’

assigns an identifying label to the joint test. If you do not specify a label, the first non-default label for the ESTIMATE rows is used to label the joint test.



**NOEST  
ONLY**

performs only the joint test and suppresses other results from the ESTIMATE statement. This option is useful for emulating the CONTRAST statement that is available in other procedures.

**NSAMP=*n***

specifies the number of samples for the simulation-based method of Silvapulle and Sen (2004). If *n* is not specified, it is constructed from the values of the ALPHA= $\alpha$ , the ACC= $\gamma$ , and the EPS= $\epsilon$  options. With the default values for  $\gamma$ ,  $\epsilon$ , and  $\alpha$  (0.005, 0.01, and 0.05, respectively), NSAMP=12,604 by default.

**CHISQ**

adds a chi-square test if the procedure produces an *F* test by default.

**BOUNDS=*value-list***

specifies boundary values for the estimable linear function. The null value of the hypothesis is always zero. If you specify a positive boundary value *z*, the hypotheses are  $H: \theta = 0$ ,  $H_a: \theta > 0$  with the added constraint that  $\theta < z$ . The same is true for negative boundary values. The alternative hypothesis is then  $H_a: \theta < 0$  subject to the constraint  $\theta > -|z|$ . If you specify a missing value, the hypothesis is assumed to be two-sided. The BOUNDS option enables you to specify sets of one- and two-sided joint hypotheses. If all values in *value-list* are set to missing, the procedure performs a simulation-based *p*-value calculation for a two-sided test.

**LOWER****LOWERTAILED**

requests that the *p*-value for the *t* test be based only on values that are less than the test statistic. A two-tailed test is the default. A lower-tailed confidence limit is also produced if you specify the CL or ALPHA= option.

Note that for ADJUST=SCHEFFE the one-sided adjusted confidence intervals and one-sided adjusted *p*-values are the same as the corresponding two-sided statistics, because this adjustment is based on only the right tail of the *F* distribution.

If you request an *F* test with the JOINT option, then a one-sided left-tailed order restriction is applied to all estimable functions, and the corresponding chi-bar-square statistic of Silvapulle and Sen (2004) is computed in addition to the two-sided, standard, *F* or chi-square statistic. See the JOINT option for how to control the computation of the simulation-based chi-bar-square statistic.

**OBSMARGINS<=*OM-data-set*>****OM<=*OM-data-set*>**

specifies a potentially different weighting scheme for the computation of LS-means coefficients. The standard LS-means have equal coefficients across classification effects; however, the OM option changes these coefficients to be proportional to those found in the *OM-data-set*. This adjustment is reasonable when you want your inferences to apply to a population that is not necessarily balanced but has the margins observed in *OM-data-set*. See the OBSMARGINS option in the LSMEANS statement for further details.

**PLOTS=***plot-options*

produces ODS statistical graphics of the distribution of estimable functions if the procedure performs the analysis in a sampling-based mode. For example, this is the case when procedures support a BAYES statement and perform a Bayesian analysis. The estimable functions are then computed for each of the posterior parameter estimates, and the “Least Squares Means Estimates” table reports simple descriptive statistics for the evaluated functions. In this situation, the PLOTS= option enables you to visualize the distribution of the estimable function. The following *plot-options* are available:

**ALL**

produces all possible plots with their default settings.

**BOXPLOT**<(*boxplot-options*)>

produces box plots of the distribution of the estimable function across the posterior sample. A separate box plot is generated for each estimable function and all box plots appear on a single graph by default. You can affect the appearance of the box plot graph with the following options:

**ORIENTATION=VERTICAL | HORIZONTAL**

**ORIENT=VERT | HORIZ** specifies the orientation of the boxes. The default is vertical orientation of the box plots.

**NPANELPOS=***number* specifies how to break the series of box plots across multiple panels. If the NPANELPOS option is not specified, or if *number* equals zero, then all box plots are displayed in a single graph; this is the default. If a negative number is specified, then exactly up to  $|number|$  of box plots are displayed per panel. If *number* is positive, then the number of boxes per panel is balanced to achieve small variation in the number of box plots per graph.

**DISTPLOT**<(*distplot-options*)>**DIST**<(*distplot-options*)>

generates panels of histograms with a kernel density overlaid. A separate plot in each panel contains the results for each estimable function. You can specify the following *distplot-options* in parentheses:

**BOX | NOBOX** controls the display of a horizontal box plot below the histogram. The BOX option is enabled by default.

**HIST | NOHIST** controls the display of the histogram of the estimable function’s distribution across the posterior sample. The HIST option is enabled by default.

**NORMAL | NONORMAL** controls the display of a normal density estimate on the graph. The NONORMAL option is enabled by default.

**KERNEL | NOKERNEL** controls the display of a kernel density estimate on the graph. The KERNEL option is enabled by default.

**NROWS=***number* specifies the highest number of rows in a panel. The default is 3.

**NCOLS=number** specifies the highest number of columns in a panel. The default is 3.

**UNPACK** unpacks the panel into separate graphics.

**NONE**

does not produce any plots.

**SEED=number**

specifies the seed for the sampling-based components of the computations for the LSMESTIMATE statement (for example, chi-bar-square statistics and simulated  $p$ -values). *number* specifies an integer that is used to start the pseudo-random-number generator for the simulation. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock. Note that there could be multiple LSMESTIMATE statements with SEED= specifications and there could be other statements that can supply a random number seed. Since the procedure has only one random number stream, the initial seed is shown in the SAS log.

**SINGULAR=number**

tunes the estimability checking as documented for the [ESTIMATE](#) statement.

**STEPDOWN<(step-down-options)>**

requests that multiplicity adjustments for the  $p$ -values of estimable functions be further adjusted in a step-down fashion. Step-down methods increase the power of multiple testing procedures by taking advantage of the fact that a  $p$ -value is never declared significant unless all smaller  $p$ -values are also declared significant. The STEPDOWN adjustment combined with [ADJUST=BON](#) corresponds to the methods of Holm (1979) and “Method 2” of Shaffer (1986); this is the default. Using step-down-adjusted  $p$ -values combined with [ADJUST=SIMULATE](#) corresponds to the method of Westfall (1997).

If the ESTIMATE statement is applied with a STEPDOWN option in a mixed model where the degrees-of-freedom method is that of Kenward and Roger (1997) or of Satterthwaite, then step-down-adjusted  $p$ -values are produced only if the [ADJDFE=ROW](#) option is in effect.

Also, the STEPDOWN option affects only  $p$ -values, not confidence limits. For [ADJUST=SIMULATE](#), the generalized least squares hybrid approach of Westfall (1997) is used to increase Monte Carlo accuracy.

You can specify the following *step-down-options* in parentheses:

**MAXTIME= $n$**

specifies the time (in seconds) to be spent computing the maximal logically consistent sequential subsets of equality hypotheses for TYPE=LOGICAL. The default is MAXTIME=60. If the MAXTIME value is exceeded, the adjusted tests are not computed. When this occurs, you can try increasing the MAXTIME value. However, note that there are common multiple comparisons problems for which this computation requires a huge amount of time—for example, all pairwise comparisons between more than 10 groups. In such cases, try to use TYPE=FREE (the default) or TYPE=LOGICAL( $n$ ) for small  $n$ .

**ORDER=PVALUE****ORDER=ROWS**

specifies the order in which the step-down tests are performed. ORDER=PVALUE is the default, with LS-mean estimates being declared significant only if all LS-mean estimates with smaller (unadjusted)  $p$ -values are significant. If you specify ORDER=ROWS, then significances are evaluated in the order in which they are specified.

**REPORT**

specifies that a report on the step-down adjustment be displayed, including a listing of the sequential subsets (Westfall 1997) and, for ADJUST=SIMULATE, the step-down simulation results.

**TYPE=LOGICAL<(n)>****TYPE=FREE**

specifies how step-down adjustment are made. If you specify TYPE=LOGICAL, the step-down adjustments are computed by using maximal logically consistent sequential subsets of equality hypotheses (Shaffer 1986, Westfall 1997). Alternatively, for TYPE=FREE, sequential subsets are computed ignoring logical constraints. The TYPE=FREE results are more conservative than those for TYPE=LOGICAL, but they can be much more efficient to produce for many estimates. For example, it is not feasible to take logical constraints between all pairwise comparisons of more than about 10 groups. For this reason, TYPE=FREE is the default.

However, you can reduce the computational complexity of taking logical constraints into account by limiting the depth of the search tree used to compute them, specifying the optional depth parameter as a number  $n$  in parentheses after TYPE=LOGICAL. As with TYPE=FREE, results for TYPE=LOGICAL( $n$ ) are conservative relative to the true TYPE=LOGICAL results. But even for TYPE=LOGICAL(0), they can be appreciably less conservative than TYPE=FREE, and they are computationally feasible for much larger numbers of estimates. If you do not specify  $n$  or if  $n = -1$ , the full search tree is used.

**TESTVALUE=value-list****TESTMEAN=value-list**

specifies the value under the null hypothesis for testing the estimable functions in the LSMESTIMATE statement. The rules for specifying the *value-list* are very similar to those for specifying the divisor list in the DIVISOR= option. If no TESTVALUE= is specified, all tests are performed as  $H: \mathbf{L}\boldsymbol{\beta} = 0$ . Missing values in the *value-list* also are translated to zeros. If you specify fewer values than rows in the LSMESTIMATE statement, the last value in *value-list* is carried forward.

The TESTVALUE= option affects only  $p$ -values from individual, joint, and multiplicity-adjusted tests. It does not affect confidence intervals.

The TESTVALUE option is not available for the multinomial distribution, and the values are ignored when you perform a sampling-based (Bayesian) analysis.

**UPPER****UPPERTAILED**

requests that the  $p$ -value for the  $t$  test be based only on values that are greater than the test statistic. A two-tailed test is the default. An upper-tailed confidence limit is also produced if you specify the **CL** or **ALPHA=** option.

Note that for **ADJUST=SCHEFFE** the one-sided adjusted confidence intervals and one-sided adjusted  $p$ -values are the same as the corresponding two-sided statistics, because this adjustment is based on only the right tail of the  $F$  distribution.

If you request a joint test with the **JOINT** option, then a one-sided right-tailed order restriction is applied to all estimable functions, and the corresponding chi-bar-square statistic of Silvapulle and Sen (2004) is computed in addition to the two-sided, standard,  $F$  or chi-square statistic. See the **JOINT** option for how to control the computation of the simulation-based chi-bar-square statistic.

---

## ODS Table Names: LSMESTIMATE Statement

Each table created by the **LSMESTIMATE** statement has a name associated with it, and you can use this name to refer to the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 19.24. For more information about ODS, see Chapter 20, “Using the Output Delivery System.”

**Table 19.24** ODS Tables Produced by the **LSMESTIMATE** statement

Table Name	Description	Required Option
Coef	L matrix coefficients or K matrix coefficients	E or ELSM
LSMEstimates	Estimates among LS-means	Default
Contrasts	Joint test results for LS-means estimates	JOINT

---

## ODS Graphics: LSMESTIMATE Statement

This section describes the use of ODS for creating statistical graphs of the distribution of LS-means and LS-mean differences with the **LSMESTIMATE** statement. The plots can be produced only in association with the two procedures that can perform Bayesian analysis (PROC GENMOD and PROC PHREG). The plots are available via these procedures directly, and also via PROC PLM when run using an item store that was created by these procedures.

To request these graphs, you must do the following:

- specify the ODS GRAPHICS ON statement
- use a BAYES statement with PROC GENMOD or PROC PHREG, or use PROC PLM to perform statistical analysis on an item store that was saved from a Bayesian analysis
- request plots with the **PLOTS=** option in the **LSMESTIMATE** statement

For more information about the ODS GRAPHICS statement, see Chapter 21, “Statistical Graphics Using ODS.” The available graphs are summarized in Table 19.25.

**Table 19.25** ODS Graphics Produced by the **LSMESTIMATE** statement

ODS Graph Name	Plot Description	Required Option
BoxPlot	Displays box plots of LS-means or LS-mean differences across a posterior sample.	PLOTS=BOXPLOT
DistPanel	Displays panels of histograms with kernel density curves overlaid. Each plot contains the results for the posterior sample of each LS-mean or LS-mean difference.	PLOTS=DISTPLOT
DistPlot	Displays a histogram with a kernel density curve overlaid. The plot contains the results for the posterior sample of the LS-mean or LS-mean difference.	PLOTS=DISTPLOT(UNPACK)

For details about the *plot-options* of the **LSMESTIMATE** statement, see the **PLOTS=** option in the section “**LSMESTIMATE Statement**” on page 496.

## NLOPTIONS Statement

This section applies to the following procedures:

GLIMMIX, HPMIXED, and TCALIS. See the individual procedure chapters for deviations from the common syntax and defaults shown here.

## Syntax: NLOPTIONS Statement

The NLOPTIONS statement provides you with syntax to control aspects of the nonlinear optimizations in the GLIMMIX, HPMIXED, and TCALIS procedures.

**NLOPTIONS** < options > ;

The nonlinear optimization options are described in alphabetical order after Table 19.26, which summarizes the options by category. The notation used in describing the options is generic in the sense that  $\psi$  denotes the  $p \times 1$  vector of parameters for the optimization and  $\psi_i$  is its  $i$ th element. The objective function being minimized, its  $p \times 1$  gradient vector, and its  $p \times p$  Hessian matrix are denoted as  $f(\psi)$ ,  $g(\psi)$ , and  $H(\psi)$ , respectively. The gradient with respect to the  $i$ th parameter is denoted as  $g_i(\psi)$ . Superscripts in parentheses denote the iteration count; for example,  $f(\psi)^{(k)}$  is the value of the objective function at iteration  $k$ . In the mixed model procedures, the parameter vector  $\psi$  might consist of fixed effects only, covariance parameters only, or fixed effects and covariance parameters. In the TCALIS procedure,  $\psi$  consists of all independent parameters that are defined in the models and in the PARAMETERS statement.

**Table 19.26** Options to Control Aspects of the Optimization

Option	Description
<b>Optimization</b>	
HESCAL=	Determines the type of Hessian scaling
INHESIAN=	Specifies the start for approximated Hessian
LINESEARCH=	Specifies the line-search method
LSPRECISION=	Specifies the line-search precision
RESTART=	Specifies the iteration number for update restart
TECHNIQUE=	Determines the minimization technique
UPDATE=	Determines the update technique
<b>Termination Criteria</b>	
ABSCONV=	Tunes an absolute function convergence criterion
ABSFCNV=	Tunes an absolute function difference convergence criterion
ABSGCONV=	Tunes the absolute gradient convergence criterion
ABSXCONV=	Tunes the absolute parameter convergence criterion
FCONV=	Tunes the relative function convergence criterion
FCONV2=	Tunes another relative function convergence criterion
FSIZE=	Specifies the value used in the FCONV and GCONV criteria
GCONV=	Tunes the relative gradient convergence criterion
GCONV2=	Tunes another relative gradient convergence criterion
MAXFUNC=	Specifies the maximum number of function calls
MAXITER=	Specifies the maximum number of iterations
MAXTIME=	Specifies the upper limit for seconds of CPU time
MINITER=	Specifies the minimum number of iterations
XCONV=	Specifies the relative parameter convergence criterion
XSIZE=	Specifies the value used in the XCONV criterion
<b>Step Length</b>	

**Table 19.26** *continued*

Option	Description
DAMPSTEP=	Dampens steps in a line search
INSTEP=	Specifies the initial trust region radius
MAXSTEP=	Specifies the maximum trust region radius
<b>Printed Output</b>	
PALL	Displays (almost) all printed output
PHISTORY	Displays optimization history
NOPRINT	Suppresses all printed output
<b>Covariance Matrix Tolerances</b>	
ASINGULAR=	Specifies the absolute singularity for inertia
MSINGULAR=	Specifies the relative M singularity for inertia
VSINGULAR=	Specifies the relative V singularity for inertia
<b>Constraint Specifications</b>	
LCEPSILON=	Specifies the range for active constraints
LCDEACT=	Specifies the LM tolerance for deactivating
LCSINGULAR=	Specifies the tolerance for dependent constraints
<b>Remote Monitoring</b>	
SOCKET=	Specifies the fileref for remote monitoring

**ABSCONV=*r*****ABSTOL=*r***

specifies an absolute function convergence criterion: for minimization, termination requires  $f(\boldsymbol{\psi}^{(k)}) \leq r$ . 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* <*n*>****ABSFTOL=*r* <*n*>**

specifies an absolute function difference convergence criterion:

- For all techniques except NMSIMP (specified by the TECHNIQUE= option), termination requires a small change of the function value in successive iterations,

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

- 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  $r = 0$ . The optional integer value  $n$  specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.



**ABSGCONV= $r < n >$** **ABSGTOL= $r < n >$** 

specifies an absolute gradient convergence criterion:

- For all techniques except NMSIMP (specified by the **TECHNIQUE=** option), termination requires the maximum absolute gradient element to be small:

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

- This criterion is not used by the NMSIMP technique.

The default value is  $r = 1\text{E}-5$ . The optional integer value  $n$  specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

**ABSXCONV= $r < n >$** **ABSXTOL= $r < n >$** 

specifies an absolute parameter convergence criterion:

- For all techniques except NMSIMP, termination requires a small Euclidean distance between successive parameter vectors,

$$\|\boldsymbol{\psi}^{(k)} - \boldsymbol{\psi}^{(k-1)}\|_2 \leq r$$

- For the NMSIMP technique, termination requires either a small length  $\alpha^{(k)}$  of the vertices of a restart simplex,

$$\alpha^{(k)} \leq r$$

or a small simplex size,

$$\delta^{(k)} \leq r$$

where the simplex size  $\delta^{(k)}$  is defined as the L1 distance from the simplex vertex  $\boldsymbol{\xi}^{(k)}$  with the smallest function value to the other  $p$  simplex points  $\boldsymbol{\psi}_l^{(k)} \neq \boldsymbol{\xi}^{(k)}$ :

$$\delta^{(k)} = \sum_{\boldsymbol{\psi}_l^{(k)} \neq \boldsymbol{\xi}^{(k)}} \|\boldsymbol{\psi}_l^{(k)} - \boldsymbol{\xi}^{(k)}\|_1$$

The default is  $r = 1\text{E}-8$  for the NMSIMP technique and  $r = 0$  otherwise. The optional integer value  $n$  specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

**ASINGULAR= $r$** **ASING= $r$** 

specifies an absolute singularity criterion for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is the square root of the smallest positive double-precision value.

**DAMPSTEP=<r>**

specifies that the initial step length value  $\alpha^{(0)}$  for each line search (used by the QUANEW, CONGRA, or NEWRAP technique) cannot be larger than  $r$  times the step length value used in the former iteration. If the DAMPSTEP option is specified but  $r$  is not specified, the default is  $r = 2$ . The DAMPSTEP= option can prevent the line-search algorithm from repeatedly stepping into regions where some objective functions are difficult to compute or where they could lead to floating-point overflows during the computation of objective functions and their derivatives. The DAMPSTEP= option can save time-consuming function calls during the line searches of objective functions that result in very small steps.

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

specifies a relative function convergence criterion:

- For all techniques except NMSIMP, termination requires a small relative change of the function value in successive iterations,

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

where FSIZE is defined by the FSIZE= option.

- 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 is  $r = 10^{-\text{FDIGITS}}$ , where FDIGITS is by default  $-\log_{10}\{\epsilon\}$  and  $\epsilon$  is the machine precision. Some procedures, such as the GLIMMIX procedure, enable you to change the value with the FDIGITS= option in the PROC statement. The optional integer value  $n$  specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

**FCONV2=<r<n>****FTOL2=<r<n>**

specifies a second function convergence criterion:

- For all techniques except NMSIMP, termination requires a small predicted reduction,

$$df^{(k)} \approx f(\boldsymbol{\psi}^{(k)}) - f(\boldsymbol{\psi}^{(k)} + \mathbf{s}^{(k)})$$

of the objective function. The predicted reduction

$$\begin{aligned} df^{(k)} &= -\mathbf{g}^{(k)'} \mathbf{s}^{(k)} - \frac{1}{2} \mathbf{s}^{(k)'} \mathbf{H}^{(k)} \mathbf{s}^{(k)} \\ &= -\frac{1}{2} \mathbf{s}^{(k)'} \mathbf{g}^{(k)} \leq r \end{aligned}$$

is computed by approximating the objective function  $f$  by the first two terms of the Taylor series and substituting the Newton step,

$$\mathbf{s}^{(k)} = -[\mathbf{H}^{(k)}]^{-1} \mathbf{g}^{(k)}$$

- For the NMSIMP technique, termination requires a small standard deviation of the function values of the  $p + 1$  simplex vertices  $\psi_l^{(k)}$ ,  $l = 0, \dots, p$ ,

$$\sqrt{\frac{1}{n+1} \sum_l \left[ f(\psi_l^{(k)}) - \bar{f}(\psi^{(k)}) \right]^2} \leq r$$

where  $\bar{f}(\psi^{(k)}) = \frac{1}{p+1} \sum_l f(\psi_l^{(k)})$ . If there are  $p_{act}$  boundary constraints active at  $\psi^{(k)}$ , the mean and standard deviation are computed only for the  $n + 1 - p_{act}$  unconstrained vertices.

The default value is  $r = 1\text{E}-6$  for the NMSIMP technique and  $r = 0$  otherwise. The optional integer value  $n$  specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

#### **FSIZE= $r$**

specifies the FSIZE parameter of the relative function and relative gradient termination criteria. The default value is  $r = 0$ . For more details, see the **FCONV=** and **GCONV=** options.

#### **GCONV= $r < n >$**

#### **GTOL= $r < n >$**

specifies a relative gradient convergence criterion:

- For all techniques except CONGRA and NMSIMP, termination requires that the normalized predicted function reduction be small,

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

where FSIZE is defined by the **FSIZE=** option. For the CONGRA technique (where a reliable Hessian estimate  $\mathbf{H}$  is not available), the following criterion is used:

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

- This criterion is not used by the NMSIMP technique.

The default value is  $r = 1\text{E}-8$ . The optional integer value  $n$  specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

#### **GCONV2= $r < n >$**

#### **GTOL2= $r < n >$**

specifies another relative gradient convergence criterion:

- For least squares problems and the TRUREG, LEVMAR, NRRIDG, and NEWRAP techniques, the following criterion of Browne (1982) is used:

$$\max_j \frac{|\mathbf{g}_j(\psi^{(k)})|}{\sqrt{f(\psi^{(k)}) \mathbf{H}_{j,j}^{(k)}}} \leq r$$

- This criterion is not used by the other techniques.

The default value is  $r = 0$ . The optional integer value  $n$  specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

**HESCAL=0 | 1 | 2 | 3**

**HS=0 | 1 | 2 | 3**

specifies the scaling version of the Hessian (or crossproduct Jacobian) matrix used in NRRIDG, TRUREG, LEVMAR, NEWRAP, or DBLDOG optimization.

If HS is not equal to 0, the first iteration and each restart iteration set the diagonal scaling matrix  $D^{(0)} = \text{diag}(d_i^{(0)})$ :

$$d_i^{(0)} = \sqrt{\max(|H_{i,i}^{(0)}|, \epsilon)}$$

where  $H_{i,i}^{(0)}$  are the diagonal elements of the Hessian (or crossproduct Jacobian). In every other iteration, the diagonal scaling matrix  $D^{(0)} = \text{diag}(d_i^{(0)})$  is updated depending on the HS option:

HS=0 specifies that no scaling be done.

HS=1 specifies the Moré (1978) scaling update:

$$d_i^{(k+1)} = \max \left[ d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)} \right]$$

HS=2 specifies the Dennis, Gay, and Welsch (1981) scaling update:

$$d_i^{(k+1)} = \max \left[ 0.6 * d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)} \right]$$

HS=3 specifies that  $d_i$  be reset in each iteration:

$$d_i^{(k+1)} = \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}$$

In each scaling update,  $\epsilon$  is the relative machine precision. The default value is HS=0. Scaling of the Hessian can be time-consuming in the case where general linear constraints are active.

**INHESSIAN<=r>**

**INHESS<=r>**

specifies how the initial estimate of the approximate Hessian is defined for the quasi-Newton techniques QUANEW and DBLDOG. There are two alternatives:

- If you do not use the  $r$  specification, the initial estimate of the approximate Hessian is set to the Hessian at  $\psi^{(0)}$ .
- If you do use the  $r$  specification, the initial estimate of the approximate Hessian is set to the multiple of the identity matrix  $r\mathbf{I}$ .

By default (if you do not specify the option `INHESIAN= $r$` ), the initial estimate of the approximate Hessian is set to the multiple of the identity matrix  $r\mathbf{I}$ , where the scalar  $r$  is computed from the magnitude of the initial gradient.

**INSTEP= $r$**

**SALPHA= $r$**

**RADIUS= $r$**

reduces the length of the first trial step during the line search of the first iterations. For highly nonlinear objective functions, such as the EXP function, the default initial radius of the trust-region algorithm TRUREG or DBLDOG or the default step length of the line-search algorithms can result in arithmetic overflows. If this occurs, you should specify decreasing values of  $0 < r < 1$  such as `INSTEP=1E-1`, `INSTEP=1E-2`, `INSTEP=1E-4`, and so on, until the iteration starts successfully.

- For trust-region algorithms (TRUREG or DBLDOG), the `INSTEP=` option specifies a factor  $r > 0$  for the initial radius  $\Delta^{(0)}$  of the trust region. The default initial trust-region radius is the length of the scaled gradient. This step corresponds to the default radius factor of  $r = 1$ .
- For line-search algorithms (NEWRAP, CONGRA, or QUANEW), the `INSTEP=` option specifies an upper bound for the initial step length for the line search during the first five iterations. The default initial step length is  $r = 1$ .
- For the Nelder-Mead simplex algorithm, by using `TECH=NMSIMP`, the `INSTEP= $r$`  option defines the size of the start simplex.

**LCDEACT= $r$**

**LCD= $r$**

specifies a threshold  $r$  for the Lagrange multiplier that determines whether an active inequality constraint remains active or can be deactivated. For maximization,  $r$  must be greater than zero; for minimization,  $r$  must be smaller than zero. An active inequality constraint can be deactivated only if its Lagrange multiplier is less than the threshold value. The default value is

$$r = \pm \min(0.01, \max(0.1 \times \text{ABSGCONV}, 0.001 \times \text{gmax}^{(k)}))$$

where “+” is for maximization, “-” is for minimization, `ABSGCONV` is the value of the absolute gradient criterion, and  $\text{gmax}^{(k)}$  is the maximum absolute element of the gradient or the projected gradient.

**LCEPSILON= $r$**

**LCEPS= $r$**

**LCE= $r$**

specifies the range  $r$  for active and violated boundary constraints, where  $r \geq 0$ . If the point  $\psi^{(k)}$  satisfies the following condition, the constraint  $i$  is recognized as an active constraint:

$$\left| \sum_{j=1}^k a_{ij} \psi_j^{(k)} - b_i \right| \leq r \times (|b_i| + 1)$$

Otherwise, the constraint  $i$  is either an inactive inequality or a violated inequality or equality constraint. The default value is  $r = 1\text{E}-8$ . During the optimization process, the introduction

of rounding errors can force the optimization to increase the value of  $r$  by a factor of  $10^k$  for some  $k > 0$ . If this happens, it is indicated by a message displayed in the log.

**LCSINGULAR= $r$**

**LCSING= $r$**

**LCS= $r$**

specifies a criterion  $r$ , where  $r \geq 0$ , that is used in the update of the QR decomposition and that determines whether an active constraint is linearly dependent on a set of other active constraints. The default value is  $r = 1\text{E}-8$ . The larger  $r$  becomes, the more the active constraints are recognized as being linearly dependent. If the value of  $r$  is larger than 0.1, it is reset to 0.1.

**LINESEARCH= $i$**

**LIS= $i$**

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques. See Fletcher (1987) for an introduction to line-search techniques. The value of  $i$  can be 1, . . . , 8 as follows. The default is LIS=2.

- |       |   |
|-------|---|
| LIS=1 | specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is similar to one used by the Harwell subroutine library.   |
| LIS=2 | specifies a line-search method that needs more function than gradient calls for quadratic and cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the <b>LSPRECISION=</b> option. This is the default. |
| LIS=3 | specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the <b>LSPRECISION=</b> option.                       |
| LIS=4 | specifies a line-search method that needs the same number of function and gradient calls for stepwise extrapolation and cubic interpolation.  |
| LIS=5 | specifies a line-search method that is a modified version of LIS=4.   |
| LIS=6 | specifies a golden-section line search (Polak 1971), which uses only function values for linear approximation.  |
| LIS=7 | specifies a bisection line search (Polak 1971), which uses only function values for linear approximation.   |
| LIS=8 | specifies the Armijo line-search technique (Polak 1971), which uses only function values for linear approximation.  |

**LSPRECISION= $r$**

**LSP= $r$**

specifies the degree of accuracy that should be obtained by the line-search algorithms **LIS=2** and **LIS=3**. Usually an imprecise line search is inexpensive and successful. For more difficult optimization problems, a more precise and expensive line search might be necessary (Fletcher 1987). The LIS=2 line-search method (which is the default for the NEWRAP, QUANEW, and CONGRA techniques) and the LIS=3 line-search method approach exact line search for small

LSPRECISION= values. If you have numerical problems, try to decrease the LSPRECISION= value to obtain a more precise line search. The default values are shown in [Table 19.27](#).

**Table 19.27** Default Values for Line-Search Precision

TECH=	UPDATE=	LSP Default
QUANEW	DBFGS, BFGS	$r = 0.4$
QUANEW	DDFP, DFP	$r = 0.06$
CONGRA	All	$r = 0.1$
NEWRAP	No update	$r = 0.9$

For more details, see Fletcher (1987).

#### **MAXFUNC=*i***

##### **MAXFU=*i***

specifies the maximum number *i* of function calls in the optimization process. The default values are as follows:

- 125 for the TRUREG, NRRIDG, NEWRAP, and LEVMAR techniques
- 500 for the QUANEW and DBLDOG techniques
- 1000 for the CONGRA technique
- 3000 for the NMSIMP technique

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.

#### **MAXITER=*i***

##### **MAXIT=*i***

specifies the maximum number *i* of iterations in the optimization process. The default values are as follows:

- 50 for the TRUREG, NRRIDG, NEWRAP, and LEVMAR techniques
- 200 for the QUANEW and DBLDOG techniques
- 400 for the CONGRA technique
- 1000 for the NMSIMP technique

These default values are also valid when *i* is specified as a missing value.

#### **MAXSTEP=*r*<*n*>**

specifies an upper bound for the step length of the line-search algorithms during the first *n* iterations. By default, *r* is the largest double-precision value and *n* is the largest integer available. Setting this option can improve the speed of convergence for the CONGRA, QUANEW, and NEWRAP techniques.

**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 much longer than that specified by the MAXTIME= option. The actual running time includes the rest of the time needed to finish the iteration and the time needed to generate the output of the results.

**MINITER=*i*****MINIT=*i***

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.

**MSINGULAR=*r*****MSING=*r***

specifies a relative singularity criterion *r*, where  $R > 0$ , for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is 1E–12.

**NOPRINT**

suppresses output that is related to optimization, such as the iteration history. This option, along with all NOPTIONS statement options for displayed output, are ignored by the GLIMMIX and HPMIXED procedures.

**PALL**

displays all optional output for optimization. This option is supported only by the TCALIS procedure.

**PHISTORY****PHIST**

displays the optimization history. The PHISTORY option is implied if the [PALL](#) option is specified. The PHISTORY option is supported only by the TCALIS procedure.

**RESTART=*i*****REST=*i***

specifies that the QUANEW or CONGRA technique is restarted with a steepest search direction after at most *i* iterations, where  $i > 0$ . Default values are as follows:

- When TECHNIQUE=CONGRA and [UPDATE](#)=PB, restart is performed automatically; so *i* is not used.
- When TECHNIQUE=CONGRA and [UPDATE](#)≠PB,  $i = \min(10p, 80)$ , where *p* is the number of parameters.
- When TECHNIQUE=QUANEW, *i* is the largest integer available.



**SINGULAR=*r*****SING=*r***

specifies the singularity criterion  $r$ ,  $0 \leq r \leq 1$ , that is used for the inversion of the Hessian matrix. The default value is 1E–8.

**SOCKET=*fileref***

specifies the fileref that contains the information needed for remote monitoring.

**TECHNIQUE=*value*****TECH=*value*****OMETHOD=*value*****OM=*value***

specifies the optimization technique. You can find additional information about choosing an optimization technique in the section “[Choosing an Optimization Algorithm](#)” on page 522. Valid values for the TECHNIQUE= option are as follows:

- **CONGRA**  
performs a conjugate-gradient optimization, which can be more precisely specified with the **UPDATE=** option and modified with the **LINESEARCH=** option. When you specify this option, **UPDATE=PB** by default.
- **DBLDOG**  
performs a version of double-dogleg optimization, which can be more precisely specified with the **UPDATE=** option. When you specify this option, **UPDATE=DBFGS** by default.
- **LEVMAR**  
performs a highly stable, but for large problems memory- and time-consuming, Levenberg-Marquardt optimization technique, a slightly improved variant of the Moré (1978) implementation. You can also specify this technique with the alias **LM** or **MARQUARDT**. In the TCALIS procedure, this is the default optimization technique if there are fewer than 40 parameters to estimate. The GLIMMIX and HPMIXED procedures do not support this optimization technique.
- **NMSIMP**  
performs a Nelder-Mead simplex optimization. The TCALIS procedure does not support this optimization technique.
- **NONE**  
does not perform any optimization. This option can be used for the following:
  - to perform a grid search without optimization
  - to compute estimates and predictions that cannot be obtained efficiently with any of the optimization techniques
  - to obtain inferences for known values of the covariance parameters
- **NEWRAP**  
performs a Newton-Raphson optimization that combines a line-search algorithm with ridging. The line-search algorithm **LIS=2** is the default method.
- **NRRIDG**  
performs a Newton-Raphson optimization with ridging.

- **QUANEW**  
performs a quasi-Newton optimization, which can be defined more precisely with the **UPDATE=** option and modified with the **LINESEARCH=** option.
- **TRUREG**  
performs a trust-region optimization.

**UPDATE=method****UPD=method**

specifies the update method for the quasi-Newton, double-dogleg, or conjugate-gradient optimization technique. Not every update method can be used with each optimizer.

The following are the valid methods for the **UPDATE=** option:

- **BFGS**  
performs the original Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the inverse Hessian matrix.
- **DBFGS**  
performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default update method.
- **DDFP**  
performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.
- **DFP**  
performs the original DFP update of the inverse Hessian matrix.
- **PB**  
performs the automatic restart update method of Powell (1977) and Beale (1972).
- **FR**  
performs the Fletcher-Reeves update (Fletcher 1987).
- **PR**  
performs the Polak-Ribiere update (Fletcher 1987).
- **CD**  
performs a conjugate-descent update of Fletcher (1987).

**VERSION=1 | 2****VS=1 | 2**

specifies the version of the quasi-Newton optimization technique with nonlinear constraints.

**VS=1** specifies the update of the  $\mu$  vector as in Powell (1978a, 1978b) (update like VF02AD).

**VS=2** specifies the update of the  $\mu$  vector as in Powell (1982a, 1982b) (update like VMCWD).

The default is **VERSION=2**.

**VSINGULAR= $r$** **VSING= $r$** 

specifies a relative singularity criterion  $r$ , where  $r > 0$ , for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is  $r = 1\text{E}-8$ .

**XCONV= $r < n >$** **XTOL= $r < n >$** 

specifies the relative parameter convergence criterion:

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

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

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

The default value is  $r = 1\text{E}-8$  for the NMSIMP technique and  $r = 0$  otherwise. The optional integer value  $n$  specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

**XSIZE= $r$** 

specifies the XSIZE parameter  $r$  of the relative parameter termination criterion, where  $r \geq 0$ . The default value is  $r = 0$ . For more details, see the [XCONV=](#) option.

## 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 19.28 shows which derivatives are required for each optimization technique.

**Table 19.28** Derivatives Required

Algorithm	First-Order	Second-Order
LEVMAR	x	x
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 where 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 where the objective function and the gradient are much faster to evaluate 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 (essentially one matrix with  $p(p + 1)/2$  double words).

The first-derivative method CONGRA is best for large problems where the objective function and the gradient can be computed much faster than the Hessian and where 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 where 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 will converge if  $\text{ABSGCONV} < 1\text{E-}5$ ,  $\text{FCONV} < 10^{-\text{FDIGITS}}$ , or  $\text{GCONV} < 1\text{E-}8$ .

## Algorithm Descriptions

### **Trust Region Optimization (TRUEG)**

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

The trust region method iteratively optimizes a quadratic approximation to the nonlinear objective function within a hyperelliptic trust region with radius  $\Delta$  that constrains the step size that corresponds to the quality of the quadratic approximation. The trust region method is implemented based on Dennis, Gay, and Welsch (1981), Gay (1983), 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 default line-search method uses quadratic interpolation and cubic extrapolation (LIS=2).

### **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.

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.

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. Usually, however, NRRIDG requires fewer iterations than NEWRAP.

### **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 are much faster to compute than the Hessian. However, in general, it requires more iterations than the TRUREG, NEWRAP, and NRRIDG techniques, which compute second-order derivatives. QUANEW is the default optimization algorithm because it provides an appropriate balance between the speed and stability required for most nonlinear mixed model applications.

The QUANEW technique is one of the following, depending upon the value of the `UPDATE=` option:

- the original quasi-Newton algorithm, which updates an approximation of the inverse Hessian
- the dual quasi-Newton algorithm, which updates the Cholesky factor of an approximate Hessian (this is the default)

You can specify four update formulas with the `UPDATE=` option:

- DBFGS performs the dual Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the Cholesky factor of the Hessian matrix. This is the default.
- DDFP performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.
- BFGS performs the original BFGS update of the inverse Hessian matrix.
- DFP performs the original DFP update of the inverse Hessian matrix.

In each iteration, a line search is performed along the search direction to find an approximate optimum. The default line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size  $\alpha$  that satisfies the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit 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. You can specify line-search algorithms other than the default with the LIS= option.

The QUANEW algorithm uses its own line-search technique. Of the options and parameters that control the line search for other algorithms, only the INSTEP= option applies here. In several applications, large steps in the first iterations are troublesome. You can use the INSTEP= option to impose an upper bound for the step size  $\alpha$  during the first five iterations. You can also use the INHESSIAN= option to specify a different starting approximation for the Hessian. If you specify only the INHESSIAN option, the Cholesky factor of a (possibly ridged) finite-difference approximation of the Hessian is used to initialize the quasi-Newton update process.

### **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; see Fletcher (1987, p. 107). Thus, the DBLDOG subroutine uses the dual quasi-Newton update but does not perform a line search. You can specify two update formulas with the UPDATE= option:

- DBFGS performs the dual Broyden, Fletcher, Goldfarb, and Shanno update of the Cholesky factor of the Hessian matrix. This is the default.
- DDFP performs the dual Davidon, Fletcher, and Powell update of the Cholesky factor of the Hessian matrix.

The double-dogleg optimization technique works well for medium-sized to moderately large optimization problems, where the objective function and the gradient are much faster to compute 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. You can specify four different update formulas for generating the conjugate directions by using the UPDATE= option:

- PB performs the automatic restart update method of Powell (1977) and Beale (1972). This is the default.
- FR performs the Fletcher-Reeves update (Fletcher 1987).
- PR performs the Polak-Ribiere update (Fletcher 1987).
- CD performs a conjugate-descent update of Fletcher (1987).

The default often behaves best for typical examples, whereas `UPDATE=CD` can perform poorly.

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 default line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size  $\alpha$  that satisfies the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit for the step size. Other line-search algorithms can be specified with the `LIS=` option.

### ***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, which contributes to an increased speed of convergence. It uses a special termination criterion.

---

## **SLICE Statement**

This statement applies to the following procedures:

GENMOD, GLIMMIX, LOGISTIC, MIXED, ORTHOREG, PHREG, PLM, SURVEYLOGISTIC, SURVEYPHREG, and SURVEYREG.

The SLICE statement is similar to the `LSMEANS` statement. You use it to perform inferences on model effects that consist entirely of classification variables. With the SLICE statement, these effects must be higher-order effects of at least two classification variables. The effect is then partitioned into subsets that correspond to variables used in forming the effect. You can use the same options as



you use for the **LSMEANS** statement to perform an analysis for the partitions. This analysis is also known as an analysis of simple effects (Winer 1971).

By default, the interaction effect is partitioned by all main effects. For example, the following statements produce simple-effect differences among the A levels for each level of B and simple-effect differences among the B levels for each level of A:

```
class a b;
model y = a b a*b;
slice a*b / diff nof;
```

For example, if the *model-effect* is a three-way interaction effect, the default output includes comparisons of the two-way interaction means.

Suppose, for example, that the interaction effect A\*B is significant in your analysis and that you want to test the effect of A for each level of B. The appropriate statement is

```
slice A*B / sliceBy = B;
```

This produces an *F* test for each level of B that compares the equality of the levels of A.

For example, assume that in a balanced design factors A and B have  $a = 4$  and  $b = 3$  levels, respectively. Consider the following statements:

```
class a b;
model y = a b a*b;
slice a*b / sliceby=a diff;
```

The SLICE statement produces four *F* tests, one per level of A. The first of these tests is constructed by extracting the three rows that correspond to the first level of A from the coefficient matrix for the A\*B interaction. Call this matrix  $\mathbf{L}_{a1}$  and its rows  $\mathbf{l}_{a1}^{(1)}$ ,  $\mathbf{l}_{a1}^{(2)}$ , and  $\mathbf{l}_{a1}^{(3)}$ . The slice tests the two-degrees-of-freedom hypothesis

$$H: \begin{cases} (\mathbf{l}_{a1}^{(1)} - \mathbf{l}_{a1}^{(2)})\boldsymbol{\beta} = 0 \\ (\mathbf{l}_{a1}^{(1)} - \mathbf{l}_{a1}^{(3)})\boldsymbol{\beta} = 0 \end{cases}$$

In a balanced design, where  $\mu_{ij}$  denotes the mean response if A is at level  $i$  and B is at level  $j$ , this hypothesis is equivalent to  $H: \mu_{11} = \mu_{12} = \mu_{13}$ . The DIFF option considers the three rows of  $\mathbf{L}_{a1}$  in turn and performs tests of the difference between pairs of rows. By default, all pairwise differences within the subset of  $\mathbf{L}$  are considered; in the example this corresponds to tests of the form

$$\begin{aligned} H: (\mathbf{l}_{a1}^{(1)} - \mathbf{l}_{a1}^{(2)})\boldsymbol{\beta} &= 0 \\ H: (\mathbf{l}_{a1}^{(1)} - \mathbf{l}_{a1}^{(3)})\boldsymbol{\beta} &= 0 \\ H: (\mathbf{l}_{a1}^{(2)} - \mathbf{l}_{a1}^{(3)})\boldsymbol{\beta} &= 0 \end{aligned}$$

In the example, with  $a = 4$  and  $b = 3$ , this produces four sets of least squares means differences. Within each set, factor A is held fixed at a particular level and each set consists of three comparisons.

## Syntax: SLICE Statement

**SLICE** *model-effect* </ options> ;

You can specify all options of the **LSMEANS** statement in the **SLICE** statement. The philosophy of the **SLICE** statement is to apply the analysis according to the options to the subsets of the **L** matrix that correspond to chosen partitions.

The following behavior differences between the **SLICE** and the **LSMEANS** statement are noteworthy:

- The specification of the *model-effect* is optional in the **LSMEANS** statement and required in the **SLICE** statement.
- Only a single **SLICE model-effect** can be specified before the option slash (/). However, you can specify multiple partitioning rules with the **SLICEBY** option.
- The **MEANS** option is the default for most procedures in the **LSMEANS** statement. For the **SLICE** statement, the default is the **NOMEANS** option.

Also, the three generalized linear modeling options: **EXP**, **ILINK**, and **ODDSRATIO** in the **SLICE** statement are additionally supported by PROC GLIMMIX and by PROC PLM when it is used to perform statistical analyses on item stores that were created by PROC GLIMMIX.

In addition to the options in the **LSMEANS** statement, you can specify the following options in the **SLICE** statement after the slash (/):

**SLICEBY** <=> *slice-specification*

**SIMPLE** <=> *slice-specification*

**SLICEBY**(*slice-specification* <, *slice-specification* <, ...>>)

**SIMPLE**(*slice-specification* <, *slice-specification* <, ...>>)

determines how to construct the partition of the least squares means for the *model-effect*. A *slice-specification* consists of an effect name followed by an optional list of formatted values. For example, the following statements creates partitions of the A\*B interaction effect for all levels of variable A:

```
class a b;
model y = a b a*b;
slice a*b / sliceby=a;
```

The following statements produces two partitions of the interaction:

```
class a b;
model y = a b a*b;
slice a*b / sliceby(b='2' a='1') diff;
```

In the first partition the variable B takes on formatted value '2'. In the second partition the variable A takes on the formatted value '1'.

**NOF**

suppresses the *F* test for testing the mutual equality of the estimable functions in the partition.

## ODS Table Names: SLICE Statement

Each table created by the **SLICE** statement has a name associated with it, and you can use this name to refer to the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 19.29. For more information about ODS, see Chapter 20, “Using the Output Delivery System.”

**Table 19.29** ODS Tables Produced by the **SLICE** statement

Table Name	Description	Required Option
Coef	L matrix coefficients	E
Slices	LS-means slices	MEANS
SliceDiffs	Simple differences of LS-means slices	DIFF or ADJUST= or STEPDOWN or NOF
SliceLines	Lines display for LS-means slices	LINES
SliceTests	Tests for LS-means slices	Default

## STORE Statement

This statement applies to the following procedures:

GENMOD, GLIMMIX, GLM, LOGISTIC, MIXED, ORTHOREG, PHREG, SURVEYLOGISTIC, SURVEYPHREG, and SURVEYREG.

The **STORE** statement requests that the procedure save the context and results of the statistical analysis into an item store. An item store is a binary file format that cannot be modified by the user. The contents of the item store can be processed with the **PLM** procedure. One example of item store technology is to perform a time-consuming analysis and to store its results by using the **STORE** statement. At a later time you can then perform specific statistical analysis tasks based on the saved results of the previous analysis, without having to fit the model again. The following statements show an example in which a mixed model is fit with the **MIXED** procedure and the postprocessing analysis is performed with the **PLM** procedure:

```
proc mixed data=MyBigDataSet;
  class Env A B sub;
  model y = A B x / ddfm=KenwardRoger;
  random int A*B / sub=Env;
  repeated / subject=Env*A*B type=AR(1);
  store sasuser.mixed;
run;

proc plm source=sasuser.mixed;
  show cov Parm;
  lsmeans A B / diff;
  score data=NewData out=ScoreResults;
run;
```

The STORE statement in the PROC MIXED step requests that the MIXED procedure save those results that are needed to perform statistical tasks with the PLM procedure. For example, the MIXED procedure saves the necessary pieces of information that relate to the Kenward-Roger degree-of-freedom method. The results from the LSMEANS statement in the PROC PLM step thus apply this technique for calculating denominator degrees of freedom. The SHOW statement in the PLM procedure reveals the contents of the item store in terms of ODS tables, and the SCORE statement computes predicted values in a new data set. For more information about postprocessing tasks based on item stores, see the documentation for the PLM procedure.

---

## Syntax: STORE Statement

**STORE** <OUT=>*item-store-name* </ LABEL='label'> ;

The *item-store-name* is a usual one- or two-level SAS name, like the names that are used for SAS data sets. If you specify a one-level name, then the item store resides in the WORK library and is deleted at the end of the SAS session. Since item stores usually are used to perform postprocessing tasks, typical usage specifies a two-level name of the form *libname.membername*.

If an item store by the same name as specified in the STORE statement already exists, the existing store is replaced.

You can add a custom label with the LABEL= option in the STORE statement after the slash (/). When the PLM procedure processes an item store, the label appears in the PROC PLM output along with other identifying information.

---

## TEST Statement

This statement documentation applies to the following procedures:  
ORTHOREG, PLM, SURVEYPHREG, and SURVEYREG.

The TEST statement enables you to perform  $F$  tests for model effects that test Type I, II, or Type III hypotheses. See Chapter 15, “The Four Types of Estimable Functions,” for details about the construction of Type I, II, and III estimable functions.

## Syntax: TEST Statement

**TEST** < *model-effects* > < / *options* > ;

Table 19.30 summarizes options in the TEST statement.

**Table 19.30** TEST Statement Options

Option	Description
CHISQ	Requests chi-square tests
DDF=	Specifies denominator degrees of freedom for fixed effects
E	Requests Type I, Type II, and Type III coefficients
E1	Requests Type I coefficients
E2	Requests Type II coefficients
E3	Requests Type III coefficients
HTYPE=	Indicates the type of hypothesis test to perform
INTERCEPT	Adds a row that corresponds to the overall intercept

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

### CHISQ

requests that chi-square tests be performed for the relevant effects in addition to the *F* tests. Type III tests are the default; you can produce the Type I and Type II tests by using the **HTYPE=** option. This option has no effect when the procedure produces chi-square statistics by default.

### DDF=*value-list*

### DF=*value-list*

specifies the denominator degrees of freedom for the fixed effects. The *value-list* specification is a list of numbers or missing values (.) separated by commas. The order of degrees of freedom should match the order of the fixed effects that are specified in the TEST statement; otherwise it should match the order in which the effects appear in the “Type III Tests of Fixed Effects” table. If you want to retain the default degrees of freedom for a particular effect, use a missing value for its location in the list. In the following example, the first TEST statement assigns 3 denominator degrees of freedom to A and 4.7 to A\*B, while those for B remain the same, and the second TEST statement assigns 5 denominator degrees of freedom to A and uses the default degrees of freedom for B.

```
model Y = A B A*B;
test / ddf=3, ., 4.7;
test B A / ddf=., 5;
```

### E

requests that Type I, Type II, and Type III **L** matrix coefficients be displayed for all relevant effects.

**E1 | EI**

requests that Type I **L** matrix coefficients be displayed for all relevant effects.

**E2 | EII**

requests that Type II **L** matrix coefficients be displayed for all relevant effects.

**E3 | EIII**

requests that Type III **L** matrix coefficients be displayed for all relevant effects.

**HTYPE=***value-list*

indicates the type of hypothesis test to perform on the fixed effects. Valid entries for values in the *value-list* are 1, 2, and 3, which correspond to Type I, Type II, and Type III tests, respectively. The default value is 3.

**INTERCEPT****INT**

adds a row to the tables for Type I, II, and III tests that correspond to the overall intercept.

---

## ODS Table Names: TEST Statement

Each table created by the **TEST** statement has a name associated with it, and you can use this name to refer to the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in [Table 19.31](#). For more information about ODS, see Chapter 20, “Using the Output Delivery System.”

**Table 19.31** ODS Tables Produced by the **TEST** statement

Table Name	Description	Required Option
Coef	<b>L</b> matrix coefficients	<b>E</b>
Tests1	Type I tests of fixed effects	<b>HTYPE=1</b>
Tests2	Type II tests of fixed effects	<b>HTYPE=2</b>
Tests3	Type III tests of fixed effects	Default

## Programming Statements

This section applies to the following procedures:  
CALIS, GLIMMIX, MCMC, NLIN, NLMIXED, and TCALIS.

The majority of the SAS/STAT modeling procedures can take advantage of the fact that the statistical model can easily be translated into programming syntax (statements and options). However, several procedures require additional flexibility in specifying models—for example, when the model contains general nonlinear functions, when it is necessary to specify complicated restrictions, or when user-supplied expressions need to be evaluated. Procedures that are listed at the beginning of the section support—in addition to the usual procedure statements and options—programming statements that can be used in the SAS DATA step.

The following are valid statements:

```
ABORT;
CALL name < ( expression < , expression ... > ) >;
DELETE;
DO < variable = expression
    < TO expression > < BY expression >
    < , expression < TO expression > < BY expression > ... >
    >
    < WHILE expression > < UNTIL expression >;
END;
GOTO statement-label;
IF expression;
IF expression THEN program-statement;
    ELSE program-statement;
variable = expression;
variable + expression;
LINK statement-label;
PUT < variable > < = > < ... >;
RETURN;
SELECT < ( expression ) >;
STOP;
SUBSTR( variable, index, length ) = expression;
WHEN ( expression ) program-statement;
    OTHERWISE program-statement;
```

For the most part, these programming statements work the same as they do in the SAS DATA step, as documented in *SAS Language Reference: Concepts*. However, there are several differences:

- The ABORT statement does not allow any arguments.

- The DO statement does not allow a character index variable. Thus

```
do i = 1,2,3;
```

is supported, whereas the following statement is not supported:

```
do i = 'A', 'B', 'C';
```

- Not all procedures support LAG functionality. For example, the GLIMMIX procedure does not support lags.
- The PUT statement, used mostly for program debugging, supports only some of the features of the DATA step PUT statement, and it has some features that are not available with the DATA step PUT statement:
  - The PUT statement does not support line pointers, factored lists, iteration factors, overprinting, `_INFILE_`, the colon (:) format modifier, or “\$”.
  - The PUT statement does support expressions, but the expression must be enclosed in parentheses. For example, the following statement displays the square root of x:

```
put (sqrt(x));
```

- The PUT statement supports the item `_PDV_` to display a formatted listing of all variables in the program. For example:

```
put _pdv_;
```

- The WHEN and OTHERWISE statements enable you to specify more than one target statement. That is, DO/END groups are not necessary for multiple-statement WHENs. For example, the following syntax is valid:

```
select;
  when (exp1) stmt1;
               stmt2;
  when (exp2) stmt3;
               stmt4;
end;
```

- The LINK statement is used in a program to jump immediately to the label *statement\_label* and to continue program execution at that point. It is not used to specify a link function in a generalized linear model.

Please consult the individual chapters for other, procedure-specific differences between programming statements and the SAS DATA step and for procedure-specific details, limitations, and rules.

When coding your programming statements, avoid defining variables that begin with an underscore (`_`), because they might conflict with internal variables that are created by procedures that support programming statements.



## References

- Affi, A. A. and Azen, S. P. (1972), *Statistical Analysis: A Computer-Oriented Approach*, New York: Academic Press.
- Beale, E. M. L. (1972), "A Derivation of Conjugate Gradients," in *Numerical Methods for Nonlinear Optimization*, ed. F. A. Lootsma, London: Academic Press.
- Browne, M. W. (1982), "Covariance Structures," in *Topics in Multivariate Analyses*, ed. D. M. Hawkins, New York: Cambridge University Press.
- 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.
- Dunnnett, C. W. (1980), "Pairwise Multiple Comparisons in the Unequal Variance Case," *Journal of the American Statistical Association*, 75, 796–800.
- Edwards, D. and Berry, J. J. (1987), "The Efficiency of Simulation-Based Multiple Comparisons," *Biometrics*, 43, 913–928.
- Eskow, E. and Schnabel, R. B. (1991), "Algorithm 695: Software for a New Modified Cholesky Factorization," *Transactions on Mathematical Software*, 17(3), 306–312.
- Fox, J. (1987), "Effect Displays for Generalized Linear Models," in *Sociological Methodology*, ed. C. C. Clogg, American Sociological Association, Washington DC, 347–361.
- Fletcher, R. (1987), *Practical Methods of Optimization*, Second Edition, Chichester: John Wiley & Sons.
- Frankel, S. A. (1961), "Statistical Design of Experiments for Process Development of MBT," *Rubber Age*, 89, 453.
- Games, P. A. and Howell, J. F. (1976), "Pairwise Multiple Comparison Procedures with Unequal  $n$ 's and/or Variances: A Monte Carlo Study," *Journal of Educational Statistics*, 1, 113–125.
- Gay, D. M. (1983), "Subroutines for Unconstrained Minimization," *ACM Transactions on Mathematical Software*, 9, 503–524.
- Guirguis, G. H. and Tobias, R. D. (2004), "On the Computation of the Distribution for the Analysis of Means," *Communications in Statistics: Simulation and Computation*, 33, 861–888.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning*, New York: Springer-Verlag.
- Holm, S. (1979), "A Simple Sequentially Rejective Multiple Test Procedure," *Scandinavian Journal of Statistics*, 6, 65–70.

- Hsu, J. C. (1992), “The Factor Analytic Approach to Simultaneous Inference in the General Linear Model,” *Journal of Computational and Graphical Statistics*, 1, 151–168.
- Hsu, J. C. (1996), *Multiple Comparisons. Theory and Methods*, London: Chapman & Hall.
- Hsu, J. C. and Peruggia, M. (1994), “Graphical Representation of Tukey’s Multiple Comparison Method,” *Journal of Computational and Graphical Statistics*, 3: 143–161.
- Kenward, M. G. and Roger, J. H. (1997), “Small Sample Inference for Fixed Effects from Restricted Maximum Likelihood,” *Biometrics*, 53, 983–997.
- Kramer, C. Y. (1956), “Extension of Multiple Range Tests to Group Means with Unequal Numbers of Replications,” *Biometrics*, 12, 309–310.
- Kutner, M. H. (1974), “Hypothesis Testing in Linear Models (Eisenhart Model),” *American Statistician*, 28, 98–100.
- Moré, J. J. (1978), “The Levenberg-Marquardt Algorithm: Implementation and Theory,” in *Lecture Notes in Mathematics 630*, ed. G.A. Watson, Berlin-Heidelberg-New York: Springer Verlag.
- Moré, J. J. and Sorensen, D. C. (1983), “Computing a Trust-Region Step,” *SIAM Journal on Scientific and Statistical Computing*, 4, 553–572.
- Myers, R. H. (1976), *Response Surface Methodology*, Blacksburg VA: Virginia Polytechnic Institute and State University.
- Nelson, P. R. (1982), “Exact Critical Points for the Analysis of Means,” *Communications in Statistics*, 11, 699–709.
- Nelson, P. R. (1991), “Numerical Evaluation of Multivariate Normal Integrals with Correlations  $\rho_{lj} = -\alpha_l \alpha_j$ ,” *The Frontiers of Statistical Scientific Theory & Industrial Applications*, 97–114.
- Nelson, P. R. (1993), “Additional Uses for the Analysis of Means and Extended Tables of Critical Values,” *Technometrics*, 35, 61–71.
- Ott, E. R. (1967), “Analysis of Means—A Graphical Procedure,” *Industrial Quality Control*, 101–109. Reprinted in *Journal of Quality Technology*, 15 (1983), 10–18.
- Polak, E. (1971), *Computational Methods in Optimization*, New York: Academic Press.
- Powell, J. M. D. (1977), “Restart Procedures for the Conjugate Gradient Method,” *Mathematical Programming*, 12, 241–254.
- Powell, J. M. D. (1978a), “A Fast Algorithm for Nonlinearly Constraint Optimization Calculations,” in *Numerical Analysis, Dundee 1977, Lecture Notes in Mathematics 630*, ed. G. A. Watson, Berlin: Springer-Verlag, 144–175.
- Powell, J. M. D. (1978b), “Algorithms for Nonlinear Constraints That Use Lagrangian Functions,” *Mathematical Programming*, 14, 224–248.
- Powell, J. M. D. (1982a), “Extensions to Subroutine VF02AD,” in *Systems Modeling and Optimization, Lecture Notes in Control and Information Sciences 38*, ed. R. F. Drenick and F. Kozin, Berlin:

Springer-Verlag, 529–538.

Powell, J. M. D. (1982b), “VMCWD: A Fortran Subroutine for Constrained Optimization,” *DAMTP 1982/NA4*, Cambridge, England.

Royen, T. (1989), “Generalized Maximum Range Tests for Pairwise Comparisons of Several Populations,” *Biometrical Journal*, 31, 905–929.

Shaffer, J. P. (1986), “Modified Sequentially Rejective Multiple Test Procedures,” *Journal of the American Statistical Association*, 81, 329–335.

Silvapulle, M. J. and Sen, P. K. (2004), *Constrained Statistical Inference: Order, Inequality, and Shape Constraints*, New York: John Wiley & Sons.

Tamhane, A. C. (1979), “A Comparison of Procedures for Multiple Comparisons of Means with Unequal Variances,” *Journal of the American Statistical Association*, 74, 471–480.

Westfall, P. H. (1997), “Multiple Testing of General Contrasts Using Logical Constraints and Correlations,” *Journal of the American Statistical Association*, 92, 299–306.

Westfall, P. H., Tobias, R. D., Rom, D., Wolfinger, R. D., and Hochberg, Y. (1999), *Multiple Comparisons and Multiple Tests Using the SAS System*, Cary, NC: SAS Institute Inc.

Westfall, P. J. and Young, S. S. (1993), *Resampling-Based Multiple Testing*, New York: John Wiley & Sons.

Winer, B. J. (1971), *Statistical Principles in Experimental Design*, Second Edition, New York: McGraw-Hill.



# Subject Index

- B-spline
  - spline basis (Shared Concepts), 434
- B-spline basis
  - GLIMMIX procedure, 434
  - GLMSELECT procedure, 434
  - HPMIXED procedure, 434
  - LOGISTIC procedure, 434
  - ORTHOREG procedure, 434
  - PHREG procedure, 434
  - PLS procedure, 434
  - QUANTREG procedure, 434
  - ROBUSTREG procedure, 434
  - SURVEYLOGISTIC procedure, 434
  - SURVEYREG procedure, 434
- bar (|) operator
  - Shared Concepts, 410
- choosing optimization algorithm
  - Shared Concepts, 522
- CLASS statement
  - Shared Concepts, 407
- classification variables
  - Shared Concepts, 407
- collection effect
  - GLIMMIX procedure, 420
  - GLMSELECT procedure, 420
  - HPMIXED procedure, 420
  - LOGISTIC procedure, 420
  - ORTHOREG procedure, 420
  - PHREG procedure, 420
  - PLS procedure, 420
  - ROBUSTREG procedure, 420
  - SURVEYLOGISTIC procedure, 420
  - SURVEYREG procedure, 420
- conjugate
  - descent (GLIMMIX), 520
  - gradient (GLIMMIX), 519
- conjugate gradient method
  - Shared Concepts, 525
- continuous-by-class effects
  - Shared Concepts, 413
- continuous-nesting-class effects
  - Shared Concepts, 412
- convergence criterion
  - GLIMMIX procedure, 510, 511, 513, 521
- crossed effects
  - Shared Concepts, 411
- Davidon-Fletcher-Powell update, 520
- double dogleg
  - method (GLIMMIX), 519
- double-dogleg method
  - Shared Concepts, 525
- effect parameterization
  - Shared Concepts, 414
- effect plot
  - EFFECTPLOT statement, 436
- EFFECT statement
  - collection effect (Shared Concepts), 420
  - lag effect (Shared Concepts), 420
  - multimember effect (Shared Concepts), 423
  - polynomial effect (Shared Concepts), 425
  - spline effect (Shared Concepts), 428
  - syntax (Shared Concepts), 418
- EFFECTPLOT statement
  - ODS graph names, 446
  - syntax (Shared Concepts), 436
- ESTIMATE statement
  - chi-bar-square statistic, 476
  - estimate-specification (Shared Concepts), 462
  - joint hypothesis tests with complex alternatives, 476
  - multiple comparison adjustment (Shared Concepts), 464
  - positional and nonpositional syntax, 473
  - syntax (Shared Concepts), 462
- Estimate-specification
  - ESTIMATE statement, 462
- examples, GLIMMIX
  - multimember effect, 423
  - spline effect, 418
- examples, GLMSELECT
  - multimember effect, 423
- examples, HPMIXED
  - multimember effect, 423
- examples, LOGISTIC
  - multimember effect, 423
- examples, ORTHOREG
  - multimember effect, 423
- examples, PHREG
  - multimember effect, 423
- examples, PLS
  - multimember effect, 423
- examples, ROBUSTREG
  - multimember effect, 423

- examples, SURVEYLOGISTIC
  - multimember effect, 423
- examples, SURVEYREG
  - multimember effect, 423
- first-order algorithm
  - Shared Concepts, 522
- general effects
  - Shared Concepts, 413
- GENMOD procedure
  - analysis of means, 486
  - diffogram, 490
  - observed margins, 488
  - ODS graph names, 446
- GENMOD procedure, LSMEANS statement
  - ODS graph names, 494
  - ODS table names, 494
- GLIMMIX procedure
  - analysis of means, 486
  - B-spline basis, 434
  - collection effect, 420
  - convergence criterion, 510, 511, 513, 521
  - diffogram, 490
  - functional convergence criteria, 512
  - Hessian scaling, 514
  - lag effect, 420
  - lag functionality, 534
  - line-search methods, 516
  - line-search precision, 516
  - multimember effect, 423
  - Newton-Raphson algorithm, 519
  - Newton-Raphson algorithm with ridging, 519
  - observed margins, 488
  - optimization technique, 519
  - polynomial effect, 425
  - remote monitoring, 519
  - spline bases, 432
  - spline effect, 428
  - TPF basis, 433
  - truncated power function basis, 433
- GLIMMIX procedure, SLICE statement
  - ODS graph names, 494
- GLM parameterization
  - Shared Concepts, 415
- GLMSELECT procedure
  - B-spline basis, 434
  - collection effect, 420
  - lag effect, 420
  - multimember effect, 423
  - polynomial effect, 425
  - spline bases, 432
  - spline effect, 428
  - TPF basis, 433
  - truncated power function basis, 433
- Hessian scaling
  - GLIMMIX procedure, 514
- HPMIXED procedure
  - B-spline basis, 434
  - collection effect, 420
  - lag effect, 420
  - multimember effect, 423
  - polynomial effect, 425
  - spline bases, 432
  - spline effect, 428
  - TPF basis, 433
  - truncated power function basis, 433
- interaction effects
  - Shared Concepts, 411
- intercept
  - Shared Concepts, 410
- lag effect
  - GLIMMIX procedure, 420
  - GLMSELECT procedure, 420
  - HPMIXED procedure, 420
  - LOGISTIC procedure, 420
  - ORTHOREG procedure, 420
  - PHREG procedure, 420
  - PLS procedure, 420
  - ROBUSTREG procedure, 420
  - SURVEYLOGISTIC procedure, 420
  - SURVEYREG procedure, 420
- lag functionality
  - GLIMMIX procedure, 534
- levelization
  - Shared Concepts, 407
- line-search methods
  - GLIMMIX procedure, 516
- LOGISTIC procedure
  - analysis of means, 486
  - B-spline basis, 434
  - chi-bar-square statistic, 476
  - collection effect, 420
  - diffogram, 490
  - joint hypothesis tests with complex alternatives, 476
  - lag effect, 420
  - multimember effect, 423
  - observed margins, 488
  - ODS graph names, 446
  - polynomial effect, 425
  - positional and nonpositional syntax, 473
  - spline bases, 432
  - spline effect, 428
  - TPF basis, 433
  - truncated power function basis, 433

- LOGISTIC procedure, ESTIMATE statement
  - ODS table names, [477](#)
- LOGISTIC procedure, LSMEANS statement
  - ODS graph names, [494](#)
  - ODS table names, [494](#)
- LSMEANS statement
  - analysis of means (Shared Concepts), [486](#)
  - diffogram (Shared Concepts), [490](#)
  - least squares means (Shared Concepts), [479](#)
  - multiple comparison adjustment (Shared Concepts), [481](#)
  - observed margins (Shared Concepts), [488](#)
  - syntax (Shared Concepts), [480](#)
- LSMESTIMATE statement
  - syntax (Shared Concepts), [497](#)
- main effects
  - Shared Concepts, [410](#)
- MIXED procedure
  - analysis of means, [486](#)
  - diffogram, [490](#)
  - observed margins, [488](#)
- MIXED procedure, SLICE statement
  - ODS graph names, [494](#)
- multimember effect
  - GLIMMIX procedure, [423](#)
  - GLMSELECT procedure, [423](#)
  - HPMIXED procedure, [423](#)
  - LOGISTIC procedure, [423](#)
  - ORTHOREG procedure, [423](#)
  - PHREG procedure, [423](#)
  - PLS procedure, [423](#)
  - ROBUSTREG procedure, [423](#)
  - SURVEYLOGISTIC procedure, [423](#)
  - SURVEYREG procedure, [423](#)
- Nelder-Mead simplex
  - method (GLIMMIX), [519](#)
- Nelder-Mead simplex method
  - Shared Concepts, [526](#)
- nested effects
  - Shared Concepts, [412](#)
- nested versus crossed effects
  - Shared Concepts, [412](#)
- Newton-Raphson algorithm
  - GLIMMIX procedure, [519](#)
- Newton-Raphson algorithm with ridging
  - GLIMMIX procedure, [519](#)
- Newton-Raphson method
  - Shared Concepts, [523](#)
- Newton-Raphson with ridging
  - Shared Concepts, [523](#)
- NLOPTIONS statement
  - syntax (Shared Concepts), [509](#)

- ODS graph names
  - EFFECTPLOT statement, [446](#)
  - GENMOD procedure, [446](#)
  - LOGISTIC procedure, [446](#)
  - ORTHOREG procedure, [446](#)
  - PLM procedure, [446](#)
  - SLICE statement (GLIMMIX), [494](#)
  - SLICE statement (MIXED), [494](#)
- optimization technique
  - GLIMMIX procedure, [519](#)
- options summary
  - EFFECT statement, [419](#)
  - ESTIMATE statement, [463](#)
  - ESTIMATE statement (LOGISTIC), [463](#)
  - ESTIMATE statement (ORTHOREG), [463](#)
  - ESTIMATE statement (PHREG), [463](#)
  - ESTIMATE statement (PLM), [463](#)
  - ESTIMATE statement (SURVEYLOGISTIC), [463](#)
  - ESTIMATE statement (SURVEYPHREG), [463](#)
  - ESTIMATE statement (SURVEYREG), [463](#)
  - LSMEANS statement (GENMOD), [480](#)
  - LSMEANS statement (LOGISTIC), [480](#)
  - LSMEANS statement (ORTHOREG), [480](#)
  - LSMEANS statement (PHREG), [480](#)
  - LSMEANS statement (PLM), [480](#)
  - LSMEANS statement (SURVEYLOGISTIC), [480](#)
  - LSMEANS statement (SURVEYPHREG), [480](#)
  - LSMEANS statement (SURVEYREG), [480](#)
  - LSMESTIMATE statement (GENMOD), [497](#)
  - LSMESTIMATE statement (LOGISTIC), [497](#)
  - LSMESTIMATE statement (MIXED), [497](#)
  - LSMESTIMATE statement (ORTHOREG), [497](#)
  - LSMESTIMATE statement (PHREG), [497](#)
  - LSMESTIMATE statement (PLM), [497](#)
  - LSMESTIMATE statement (SURVEYLOGISTIC), [497](#)
  - LSMESTIMATE statement (SURVEYPHREG), [497](#)
  - LSMESTIMATE statement (SURVEYREG), [497](#)
  - NLOPTIONS statement (GLIMMIX), [509](#)
  - NLOPTIONS statement (HPMIXED), [509](#)
  - NLOPTIONS statement (TCALIS), [509](#)
  - SLICE statement (GENMOD), [480](#)
  - SLICE statement (GLIMMIX), [480](#)
  - SLICE statement (LOGISTIC), [480](#)
  - SLICE statement (MIXED), [480](#)
  - SLICE statement (ORTHOREG), [480](#)

- SLICE statement (PHREG), 480
- SLICE statement (PLM), 480
- SLICE statement (SURVEYLOGISTIC), 480
- SLICE statement (SURVEYPHREG), 480
- SLICE statement (SURVEYREG), 480
- TEST statement (ORTHOREG), 531
- TEST statement (PLM), 531
- TEST statement (SURVEYPHREG), 531
- TEST statement (SURVEYREG), 531
- ordering
  - of class levels (Shared Concepts), 408
- ordinal parameterization
  - Shared Concepts, 415
- ortheffect parameterization
  - Shared Concepts, 416
- orthordinal parameterization
  - Shared Concepts, 417
- ORTHOREG procedure
  - analysis of means, 486
  - B-spline basis, 434
  - chi-bar-square statistic, 476
  - collection effect, 420
  - diffogram, 490
  - joint hypothesis tests with complex
    - alternatives, 476
  - lag effect, 420
  - multimember effect, 423
  - observed margins, 488
  - ODS graph names, 446
  - polynomial effect, 425
  - positional and nonpositional syntax, 473
  - spline bases, 432
  - spline effect, 428
  - TPF basis, 433
  - truncated power function basis, 433
- ORTHOREG procedure, ESTIMATE statement
  - ODS table names, 477
- ORTHOREG procedure, LSMEANS statement
  - ODS graph names, 494
  - ODS table names, 494
- orthoterm parameterization
  - Shared Concepts, 417
- orthpoly parameterization
  - Shared Concepts, 417
- orthref parameterization
  - Shared Concepts, 417
- parameterization
  - effect (Shared Concepts), 414
  - GLM (Shared Concepts), 415
  - ordinal (Shared Concepts), 415
  - ortheffect (Shared Concepts), 416
  - orthordinal (Shared Concepts), 417
  - orthoterm (Shared Concepts), 417
  - orthpoly (Shared Concepts), 417
  - orthref (Shared Concepts), 417
  - polynomial (Shared Concepts), 415
  - reference (Shared Concepts), 416
  - Shared Concepts, 409
  - thermometer (Shared Concepts), 415
- PHREG procedure
  - analysis of means, 486
  - B-spline basis, 434
  - chi-bar-square statistic, 476
  - collection effect, 420
  - diffogram, 490
  - joint hypothesis tests with complex
    - alternatives, 476
  - lag effect, 420
  - multimember effect, 423
  - observed margins, 488
  - polynomial effect, 425
  - positional and nonpositional syntax, 473
  - spline bases, 432
  - spline effect, 428
  - TPF basis, 433
  - truncated power function basis, 433
- PHREG procedure, ESTIMATE statement
  - ODS graph names, 478
  - ODS table names, 477
- PHREG procedure, LSMEANS statement
  - ODS graph names, 494
  - ODS table names, 494
- PLM procedure
  - analysis of means, 486
  - chi-bar-square statistic, 476
  - diffogram, 490
  - joint hypothesis tests with complex
    - alternatives, 476
  - observed margins, 488
  - ODS graph names, 446
  - positional and nonpositional syntax, 473
- PLM procedure, ESTIMATE statement
  - ODS graph names, 478
  - ODS table names, 477
- PLM procedure, LSMEANS statement
  - ODS graph names, 494
  - ODS table names, 494
- PLS procedure
  - B-spline basis, 434
  - collection effect, 420
  - lag effect, 420
  - multimember effect, 423
  - polynomial effect, 425
  - spline bases, 432
  - spline effect, 428
  - TPF basis, 433
  - truncated power function basis, 433



- polynomial effect
  - GLIMMIX procedure, 425
  - GLMSELECT procedure, 425
  - HPMIXED procedure, 425
  - LOGISTIC procedure, 425
  - ORTHOREG procedure, 425
  - PHREG procedure, 425
  - PLS procedure, 425
  - ROBUSTREG procedure, 425
  - SURVEYLOGISTIC procedure, 425
  - SURVEYREG procedure, 425
- polynomial effects
  - Shared Concepts, 410
- polynomial parameterization
  - Shared Concepts, 415
- programming statements
  - Shared Concepts, 533
- QUANTREG procedure
  - B-spline basis, 434
  - spline bases, 432
  - spline effect, 428
  - TPF basis, 433
  - truncated power function basis, 433
- quasi-Newton method
  - Shared Concepts, 524
- reference parameterization
  - Shared Concepts, 416
- regression effects
  - Shared Concepts, 410
- remote monitoring
  - GLIMMIX procedure, 519
- ROBUSTREG procedure
  - B-spline basis, 434
  - collection effect, 420
  - lag effect, 420
  - multimember effect, 423
  - polynomial effect, 425
  - spline bases, 432
  - spline effect, 428
  - TPF basis, 433
  - truncated power function basis, 433
- second-order algorithm
  - Shared Concepts, 522
- Shared Concepts
  - bar (|) operator, 410
  - choosing optimization algorithm, 522
  - CLASS statement, 407
  - classification variables, 407
  - collection effect (EFFECT statement), 420
  - conjugate gradient method, 525
  - continuous-by-class effects, 413
  - continuous-nesting-class effects, 412
  - crossed effects, 411
  - double-dogleg method, 525
  - effect parameterization, 414
  - EFFECT statement, 418
  - EFFECTPLOT statement, 436
  - ESTIMATE statement, 462
  - first-order algorithm, 522
  - general effects, 413
  - GLM parameterization, 415
  - interaction effects, 411
  - intercept, 410
  - lag effect (EFFECT statement), 420
  - levelization, 407
  - LSMEANS statement, 480
  - LSMESTIMATE statement, 497
  - main effects, 410
  - missing values, class variables, 408
  - multimember effect (EFFECT statement), 423
  - Nelder-Mead simplex method, 526
  - nested effects, 412
  - nested versus crossed effects, 412
  - Newton-Raphson method, 523
  - Newton-Raphson with ridging, 523
  - NLOPTIONS statement, 509
  - ORDER= option, 408
  - ordering of class levels, 408
  - ordinal parameterization, 415
  - ortheffect parameterization, 416
  - orthordinal parameterization, 417
  - orthoterm parameterization, 417
  - orthpoly parameterization, 417
  - orthref parameterization, 417
  - parameterization, 409
  - polynomial effect (EFFECT statement), 425
  - polynomial effects, 410
  - polynomial parameterization, 415
  - programming statements, 533
  - quasi-Newton method, 524
  - reference parameterization, 416
  - regression effects, 410
  - second-order algorithm, 522
  - simplex method, 526
  - singular parameterization, 411
  - SLICE statement, 528
  - sort order of class levels, 408
  - spline bases, 432
  - spline basis, B-spline, 434
  - spline basis, truncated power function, 433
  - spline effect (EFFECT statement), 428
  - splines, 432
  - TEST statement, 531
  - thermometer parameterization, 415
  - trust region method, 523

- simplex method
  - Shared Concepts, 526
- singular parameterization
  - Shared Concepts, 411
- SLICE statement
  - syntax (Shared Concepts), 528
- sort order
  - of class levels (Shared Concepts), 408
- spline bases
  - GLIMMIX procedure, 432
  - GLMSELECT procedure, 432
  - HPMIXED procedure, 432
  - LOGISTIC procedure, 432
  - ORTHOREG procedure, 432
  - PHREG procedure, 432
  - PLS procedure, 432
  - QUANTREG procedure, 432
  - ROBUSTREG procedure, 432
  - Shared Concepts, 432
  - SURVEYLOGISTIC procedure, 432
  - SURVEYREG procedure, 432
- spline effect
  - GLIMMIX procedure, 428
  - GLMSELECT procedure, 428
  - HPMIXED procedure, 428
  - LOGISTIC procedure, 428
  - ORTHOREG procedure, 428
  - PHREG procedure, 428
  - PLS procedure, 428
  - QUANTREG procedure, 428
  - ROBUSTREG procedure, 428
  - SURVEYLOGISTIC procedure, 428
  - SURVEYREG procedure, 428
- splines
  - Shared Concepts, 432
- SURVEYLOGISTIC procedure
  - analysis of means, 486
  - B-spline basis, 434
  - chi-bar-square statistic, 476
  - collection effect, 420
  - diffogram, 490
  - joint hypothesis tests with complex alternatives, 476
  - lag effect, 420
  - multimember effect, 423
  - observed margins, 488
  - polynomial effect, 425
  - positional and nonpositional syntax, 473
  - spline bases, 432
  - spline effect, 428
  - TPF basis, 433
  - truncated power function basis, 433
- SURVEYLOGISTIC procedure, ESTIMATE statement
  - ODS table names, 477
- SURVEYLOGISTIC procedure, LSMEANS statement
  - ODS graph names, 494
  - ODS table names, 494
- SURVEYPHREG procedure
  - analysis of means, 486
  - chi-bar-square statistic, 476
  - diffogram, 490
  - joint hypothesis tests with complex alternatives, 476
  - observed margins, 488
  - positional and nonpositional syntax, 473
- SURVEYPHREG procedure, ESTIMATE statement
  - ODS table names, 477
- SURVEYPHREG procedure, LSMEANS statement
  - ODS graph names, 494
  - ODS table names, 494
- SURVEYREG procedure
  - analysis of means, 486
  - B-spline basis, 434
  - chi-bar-square statistic, 476
  - collection effect, 420
  - diffogram, 490
  - joint hypothesis tests with complex alternatives, 476
  - lag effect, 420
  - multimember effect, 423
  - observed margins, 488
  - polynomial effect, 425
  - positional and nonpositional syntax, 473
  - spline bases, 432
  - spline effect, 428
  - TPF basis, 433
  - truncated power function basis, 433
- SURVEYREG procedure, ESTIMATE statement
  - ODS table names, 477
- SURVEYREG procedure, LSMEANS statement
  - ODS graph names, 494
  - ODS table names, 494
- TEST statement
  - syntax (Shared Concepts), 531
- thermometer parameterization
  - Shared Concepts, 415
- TPF basis
  - GLIMMIX procedure, 433
  - GLMSELECT procedure, 433
  - HPMIXED procedure, 433
  - LOGISTIC procedure, 433
  - ORTHOREG procedure, 433
  - PHREG procedure, 433

- PLS procedure, [433](#)
- QUANTREG procedure, [433](#)
- ROBUSTREG procedure, [433](#)
- SURVEYLOGISTIC procedure, [433](#)
- SURVEYREG procedure, [433](#)
- truncated power function
  - spline basis (Shared Concepts), [433](#)
- truncated power function basis
  - GLIMMIX procedure, [433](#)
  - GLMSELECT procedure, [433](#)
  - HPMIXED procedure, [433](#)
  - LOGISTIC procedure, [433](#)
  - ORTHOREG procedure, [433](#)
  - PHREG procedure, [433](#)
  - PLS procedure, [433](#)
  - QUANTREG procedure, [433](#)
  - ROBUSTREG procedure, [433](#)
  - SURVEYLOGISTIC procedure, [433](#)
  - SURVEYREG procedure, [433](#)
- trust region method
  - Shared Concepts, [523](#)



# Syntax Index

- ABSCONV option
    - NLOPTIONS statement (GLIMMIX), 510
    - NLOPTIONS statement (HPMIXED), 510
    - NLOPTIONS statement (TCALIS), 510
  - ABSFCNV option
    - NLOPTIONS statement (GLIMMIX), 510
    - NLOPTIONS statement (HPMIXED), 510
    - NLOPTIONS statement (TCALIS), 510
  - ABSGCONV option
    - NLOPTIONS statement (GLIMMIX), 511
    - NLOPTIONS statement (HPMIXED), 511
    - NLOPTIONS statement (TCALIS), 511
  - ABSGTOL option
    - NLOPTIONS statement (GLIMMIX), 511
    - NLOPTIONS statement (HPMIXED), 511
    - NLOPTIONS statement (TCALIS), 511
  - ABSTOL option
    - NLOPTIONS statement (GLIMMIX), 510
    - NLOPTIONS statement (HPMIXED), 510
    - NLOPTIONS statement (TCALIS), 510
  - ABSXCONV option
    - NLOPTIONS statement (GLIMMIX), 511
    - NLOPTIONS statement (HPMIXED), 511
    - NLOPTIONS statement (TCALIS), 511
  - ABSXTOL option
    - NLOPTIONS statement (GLIMMIX), 511
    - NLOPTIONS statement (HPMIXED), 511
    - NLOPTIONS statement (TCALIS), 511
  - ADJDFE= option
    - ESTIMATE statement (ORTHOREG), 464
    - ESTIMATE statement (PLM), 464
    - ESTIMATE statement (SURVEYPHREG), 464
    - ESTIMATE statement (SURVEYREG), 464
    - LSMEANS statement (ORTHOREG), 481
    - LSMEANS statement (PLM), 481
    - LSMEANS statement (SURVEYPHREG), 481
    - LSMEANS statement (SURVEYREG), 481
    - LSMESTIMATE statement (MIXED), 498
    - LSMESTIMATE statement (ORTHOREG), 498
    - LSMESTIMATE statement (PLM), 498
    - LSMESTIMATE statement (SURVEYPHREG), 498
    - LSMESTIMATE statement (SURVEYREG), 498
    - SLICE statement (GLIMMIX), 481
  - SLICE statement (MIXED), 481
  - SLICE statement (ORTHOREG), 481
  - SLICE statement (PLM), 481
- ADJUST= option
- ESTIMATE statement (LOGISTIC), 465
  - ESTIMATE statement (ORTHOREG), 465
  - ESTIMATE statement (PHREG), 465
  - ESTIMATE statement (PLM), 465
  - ESTIMATE statement (SURVEYLOGISTIC), 465
  - ESTIMATE statement (SURVEYPHREG), 465
  - ESTIMATE statement (SURVEYREG), 465
  - LSMEANS statement (GENMOD), 482
  - LSMEANS statement (LOGISTIC), 482
  - LSMEANS statement (ORTHOREG), 482
  - LSMEANS statement (PHREG), 482
  - LSMEANS statement (PLM), 482
  - LSMEANS statement (SURVEYLOGISTIC), 482
  - LSMEANS statement (SURVEYPHREG), 482
  - LSMEANS statement (SURVEYREG), 482
  - LSMESTIMATE statement (GENMOD), 499
  - LSMESTIMATE statement (LOGISTIC), 499
  - LSMESTIMATE statement (MIXED), 499
  - LSMESTIMATE statement (ORTHOREG), 499
  - LSMESTIMATE statement (PHREG), 499
  - LSMESTIMATE statement (PLM), 499
  - LSMESTIMATE statement (SURVEYLOGISTIC), 499
  - LSMESTIMATE statement (SURVEYPHREG), 499
  - LSMESTIMATE statement (SURVEYREG), 499
  - SLICE statement (GENMOD), 482
  - SLICE statement (GLIMMIX), 482
  - SLICE statement (LOGISTIC), 482
  - SLICE statement (MIXED), 482
  - SLICE statement (ORTHOREG), 482
  - SLICE statement (PHREG), 482
  - SLICE statement (PLM), 482
  - SLICE statement (SURVEYLOGISTIC), 482
  - SLICE statement (SURVEYPHREG), 482
  - SLICE statement (SURVEYREG), 482
- ALPHA= option

- EFFECTPLOT statement, 438
- ESTIMATE statement (LOGISTIC), 465
- ESTIMATE statement (ORTHOREG), 465
- ESTIMATE statement (PHREG), 465
- ESTIMATE statement (PLM), 465
- ESTIMATE statement
  - (SURVEYLOGISTIC), 465
- ESTIMATE statement (SURVEYPHREG), 465
- ESTIMATE statement (SURVEYREG), 465
- LSMEANS statement (GENMOD), 484
- LSMEANS statement (LOGISTIC), 484
- LSMEANS statement (ORTHOREG), 484
- LSMEANS statement (PHREG), 484
- LSMEANS statement (PLM), 484
- LSMEANS statement (SURVEYLOGISTIC), 484
- LSMEANS statement (SURVEYPHREG), 484
- LSMEANS statement (SURVEYREG), 484
- LSMESTIMATE statement (GENMOD), 499
- LSMESTIMATE statement (LOGISTIC), 499
- LSMESTIMATE statement (MIXED), 499
- LSMESTIMATE statement (ORTHOREG), 499
- LSMESTIMATE statement (PHREG), 499
- LSMESTIMATE statement (PLM), 499
- LSMESTIMATE statement
  - (SURVEYLOGISTIC), 499
- LSMESTIMATE statement
  - (SURVEYPHREG), 499
- LSMESTIMATE statement (SURVEYREG), 499
- SLICE statement (GENMOD), 484
- SLICE statement (GLIMMIX), 484
- SLICE statement (LOGISTIC), 484
- SLICE statement (MIXED), 484
- SLICE statement (ORTHOREG), 484
- SLICE statement (PHREG), 484
- SLICE statement (PLM), 484
- SLICE statement (SURVEYLOGISTIC), 484
- SLICE statement (SURVEYPHREG), 484
- SLICE statement (SURVEYREG), 484

ATLEN= option

- EFFECTPLOT statement, 439

ATORDER= option

- EFFECTPLOT statement, 439

BASIS option

- EFFECT statement, spline (GLIMMIX), 429
- EFFECT statement, spline (GLMSELECT), 429
- EFFECT statement, spline (HPMIXED), 429
- EFFECT statement, spline (LOGISTIC), 429
- EFFECT statement, spline (ORTHOREG), 429
- EFFECT statement, spline (PHREG), 429
- EFFECT statement, spline (PLS), 429
- EFFECT statement, spline (QUANTREG), 429
- EFFECT statement, spline (ROBUSTREG), 429
- EFFECT statement, spline
  - (SURVEYLOGISTIC), 429
- EFFECT statement, spline (SURVEYREG), 429

ASINGULAR= option

- NLOPTIONS statement (GLIMMIX), 511
- NLOPTIONS statement (HPMIXED), 511
- NLOPTIONS statement (TCALIS), 511

AT option

- EFFECTPLOT statement, 438

AT= option

- LSMEANS statement (GENMOD), 484
- LSMEANS statement (LOGISTIC), 484
- LSMEANS statement (ORTHOREG), 484

#### BYLEVEL option

- LSMEANS statement (GENMOD), 485
- LSMEANS statement (LOGISTIC), 485
- LSMEANS statement (ORTHOREG), 485
- LSMEANS statement (PHREG), 485
- LSMEANS statement (PLM), 485
- LSMEANS statement (SURVEYLOGISTIC), 485
- LSMEANS statement (SURVEYPHREG), 485
- LSMEANS statement (SURVEYREG), 485
- LSMESTIMATE statement (GENMOD), 499
- LSMESTIMATE statement (LOGISTIC), 499
- LSMESTIMATE statement (MIXED), 499
- LSMESTIMATE statement (ORTHOREG), 499
- LSMESTIMATE statement (PHREG), 499
- LSMESTIMATE statement (PLM), 499
- LSMESTIMATE statement (SURVEYLOGISTIC), 499
- LSMESTIMATE statement (SURVEYPHREG), 499
- LSMESTIMATE statement (SURVEYREG), 499
- SLICE statement (GENMOD), 485
- SLICE statement (GLIMMIX), 485
- SLICE statement (LOGISTIC), 485
- SLICE statement (MIXED), 485
- SLICE statement (ORTHOREG), 485
- SLICE statement (PHREG), 485
- SLICE statement (PLM), 485
- SLICE statement (SURVEYLOGISTIC), 485
- SLICE statement (SURVEYPHREG), 485
- SLICE statement (SURVEYREG), 485

#### CATEGORY= option

- ESTIMATE statement (LOGISTIC), 465
- ESTIMATE statement (PLM), 465
- ESTIMATE statement (SURVEYLOGISTIC), 465
- LSMESTIMATE statement (GENMOD), 500
- LSMESTIMATE statement (LOGISTIC), 500
- LSMESTIMATE statement (PLM), 500
- LSMESTIMATE statement (SURVEYLOGISTIC), 500

#### CHISQ option

- ESTIMATE statement (ORTHOREG), 466
- ESTIMATE statement (PLM), 466
- ESTIMATE statement (SURVEYPHREG), 466
- ESTIMATE statement (SURVEYREG), 466
- LSMESTIMATE statement (MIXED), 500

- LSMESTIMATE statement (ORTHOREG), 500

- LSMESTIMATE statement (PLM), 500

- LSMESTIMATE statement (SURVEYPHREG), 500

- LSMESTIMATE statement (SURVEYREG), 500

- TEST statement (ORTHOREG), 531

- TEST statement (PLM), 531

- TEST statement (SURVEYPHREG), 531

- TEST statement (SURVEYREG), 531

#### CL option

- ESTIMATE statement (LOGISTIC), 466

- ESTIMATE statement (ORTHOREG), 466

- ESTIMATE statement (PHREG), 466

- ESTIMATE statement (PLM), 466

- ESTIMATE statement (SURVEYLOGISTIC), 466

- ESTIMATE statement (SURVEYPHREG), 466

- ESTIMATE statement (SURVEYREG), 466

- LSMEANS statement (GENMOD), 485

- LSMEANS statement (LOGISTIC), 485

- LSMEANS statement (ORTHOREG), 485

- LSMEANS statement (PHREG), 485

- LSMEANS statement (PLM), 485

- LSMEANS statement (SURVEYLOGISTIC), 485

- LSMEANS statement (SURVEYPHREG), 485

- LSMEANS statement (SURVEYREG), 485

- LSMESTIMATE statement (GENMOD), 500

- LSMESTIMATE statement (LOGISTIC), 500

- LSMESTIMATE statement (MIXED), 500

- LSMESTIMATE statement (ORTHOREG), 500

- LSMESTIMATE statement (PHREG), 500

- LSMESTIMATE statement (PLM), 500

- LSMESTIMATE statement (SURVEYLOGISTIC), 500

- LSMESTIMATE statement (SURVEYPHREG), 500

- LSMESTIMATE statement (SURVEYREG), 500

- SLICE statement (GENMOD), 485

- SLICE statement (GLIMMIX), 485

- SLICE statement (LOGISTIC), 485

- SLICE statement (MIXED), 485

- SLICE statement (ORTHOREG), 485

- SLICE statement (PHREG), 485

- SLICE statement (PLM), 485

- SLICE statement (SURVEYLOGISTIC), 485

- SLICE statement (SURVEYPHREG), 485



- SLICE statement (SURVEYREG), 485
- CLI option
  - EFFECTPLOT statement, 440
- CLM option
  - EFFECTPLOT statement, 440
- CORR option
  - ESTIMATE statement (LOGISTIC), 466
  - ESTIMATE statement (ORTHOREG), 466
  - ESTIMATE statement (PHREG), 466
  - ESTIMATE statement (PLM), 466
  - ESTIMATE statement
    - (SURVEYLOGISTIC), 466
  - ESTIMATE statement (SURVEYPHREG), 466
  - ESTIMATE statement (SURVEYREG), 466
  - LSMEANS statement (GENMOD), 485
  - LSMEANS statement (LOGISTIC), 485
  - LSMEANS statement (ORTHOREG), 485
  - LSMEANS statement (PHREG), 485
  - LSMEANS statement (PLM), 485
  - LSMEANS statement (SURVEYLOGISTIC), 485
  - LSMEANS statement (SURVEYPHREG), 485
  - LSMEANS statement (SURVEYREG), 485
  - LSMESTIMATE statement (GENMOD), 500
  - LSMESTIMATE statement (LOGISTIC), 500
  - LSMESTIMATE statement (MIXED), 500
  - LSMESTIMATE statement (ORTHOREG), 500
  - LSMESTIMATE statement (PHREG), 500
  - LSMESTIMATE statement (PLM), 500
  - LSMESTIMATE statement
    - (SURVEYLOGISTIC), 500
  - LSMESTIMATE statement
    - (SURVEYPHREG), 500
  - LSMESTIMATE statement (SURVEYREG), 500
  - SLICE statement (GENMOD), 485
  - SLICE statement (GLIMMIX), 485
  - SLICE statement (LOGISTIC), 485
  - SLICE statement (MIXED), 485
  - SLICE statement (ORTHOREG), 485
  - SLICE statement (PHREG), 485
  - SLICE statement (PLM), 485
  - SLICE statement (SURVEYLOGISTIC), 485
  - SLICE statement (SURVEYPHREG), 485
  - SLICE statement (SURVEYREG), 485
- COV option
  - ESTIMATE statement (LOGISTIC), 466
  - ESTIMATE statement (ORTHOREG), 466
  - ESTIMATE statement (PHREG), 466
  - ESTIMATE statement (PLM), 466
- ESTIMATE statement
  - (SURVEYLOGISTIC), 466
- ESTIMATE statement (SURVEYPHREG), 466
- ESTIMATE statement (SURVEYREG), 466
- LSMEANS statement (GENMOD), 485
- LSMEANS statement (LOGISTIC), 485
- LSMEANS statement (ORTHOREG), 485
- LSMEANS statement (PHREG), 485
- LSMEANS statement (PLM), 485
- LSMEANS statement (SURVEYLOGISTIC), 485
- LSMEANS statement (SURVEYPHREG), 485
- LSMEANS statement (SURVEYREG), 485
- LSMESTIMATE statement (GENMOD), 500
- LSMESTIMATE statement (LOGISTIC), 500
- LSMESTIMATE statement (MIXED), 500
- LSMESTIMATE statement (ORTHOREG), 500
- LSMESTIMATE statement (PHREG), 500
- LSMESTIMATE statement (PLM), 500
- LSMESTIMATE statement
  - (SURVEYLOGISTIC), 500
- LSMESTIMATE statement
  - (SURVEYPHREG), 500
- LSMESTIMATE statement (SURVEYREG), 500
- SLICE statement (GENMOD), 485
- SLICE statement (GLIMMIX), 485
- SLICE statement (LOGISTIC), 485
- SLICE statement (MIXED), 485
- SLICE statement (ORTHOREG), 485
- SLICE statement (PHREG), 485
- SLICE statement (PLM), 485
- SLICE statement (SURVEYLOGISTIC), 485
- SLICE statement (SURVEYPHREG), 485
- SLICE statement (SURVEYREG), 485



- EFFECT statement, spline (ROBUSTREG), 429
- EFFECT statement, spline (SURVEYLOGISTIC), 429
- EFFECT statement, spline (SURVEYREG), 429
- DDF= option
  - TEST statement (ORTHOREG), 531
  - TEST statement (PLM), 531
  - TEST statement (SURVEYPHREG), 531
  - TEST statement (SURVEYREG), 531
- DEGREE option
  - EFFECT statement, polynomial (GLIMMIX), 425
  - EFFECT statement, polynomial (GLMSELECT), 425
  - EFFECT statement, polynomial (HPMIXED), 425
  - EFFECT statement, polynomial (LOGISTIC), 425
  - EFFECT statement, polynomial (ORTHOREG), 425
  - EFFECT statement, polynomial (PHREG), 425
  - EFFECT statement, polynomial (PLS), 425
  - EFFECT statement, polynomial (ROBUSTREG), 425
  - EFFECT statement, polynomial (SURVEYLOGISTIC), 425
  - EFFECT statement, polynomial (SURVEYREG), 425
  - EFFECT statement, spline (GLIMMIX), 429
  - EFFECT statement, spline (GLMSELECT), 429
  - EFFECT statement, spline (HPMIXED), 429
  - EFFECT statement, spline (LOGISTIC), 429
  - EFFECT statement, spline (ORTHOREG), 429
  - EFFECT statement, spline (PHREG), 429
  - EFFECT statement, spline (PLS), 429
  - EFFECT statement, spline (QUANTREG), 429
  - EFFECT statement, spline (ROBUSTREG), 429
  - EFFECT statement, spline (SURVEYLOGISTIC), 429
  - EFFECT statement, spline (SURVEYREG), 429
- DESIGNROLE option
  - EFFECT statement, lag (GLIMMIX), 422
  - EFFECT statement, lag (GLMSELECT), 422
  - EFFECT statement, lag (HPMIXED), 422
  - EFFECT statement, lag (LOGISTIC), 422
  - EFFECT statement, lag (ORTHOREG), 422
  - EFFECT statement, lag (PHREG), 422
  - EFFECT statement, lag (PLS), 422
  - EFFECT statement, lag (ROBUSTREG), 422
  - EFFECT statement, lag (SURVEYLOGISTIC), 422
  - EFFECT statement, lag (SURVEYREG), 422
- DETAILS option
  - EFFECT statement, lag (GLIMMIX), 422
  - EFFECT statement, lag (GLMSELECT), 422
  - EFFECT statement, lag (HPMIXED), 422
  - EFFECT statement, lag (LOGISTIC), 422
  - EFFECT statement, lag (ORTHOREG), 422
  - EFFECT statement, lag (PHREG), 422
  - EFFECT statement, lag (PLS), 422
  - EFFECT statement, lag (ROBUSTREG), 422
  - EFFECT statement, lag (SURVEYLOGISTIC), 422
  - EFFECT statement, lag (SURVEYREG), 422
  - EFFECT statement, multimember (GLIMMIX), 424
  - EFFECT statement, multimember (GLMSELECT), 424
  - EFFECT statement, multimember (HPMIXED), 424
  - EFFECT statement, multimember (LOGISTIC), 424
  - EFFECT statement, multimember (ORTHOREG), 424
  - EFFECT statement, multimember (PHREG), 424
  - EFFECT statement, multimember (PLS), 424
  - EFFECT statement, multimember (ROBUSTREG), 424
  - EFFECT statement, multimember (SURVEYLOGISTIC), 424
  - EFFECT statement, multimember (SURVEYREG), 424
  - EFFECT statement, polynomial (GLIMMIX), 425
  - EFFECT statement, polynomial (GLMSELECT), 425
  - EFFECT statement, polynomial (HPMIXED), 425
  - EFFECT statement, polynomial (LOGISTIC), 425
  - EFFECT statement, polynomial (ORTHOREG), 425
  - EFFECT statement, polynomial (PHREG), 425
  - EFFECT statement, polynomial (PLS), 425
  - EFFECT statement, polynomial (ROBUSTREG), 425
  - EFFECT statement, polynomial (SURVEYLOGISTIC), 425

- EFFECT statement, polynomial  
(SURVEYREG), 425
- EFFECT statement, spline (GLIMMIX), 429
- EFFECT statement, spline (GLMSELECT),  
429
- EFFECT statement, spline (HPMIXED), 429
- EFFECT statement, spline (LOGISTIC), 429
- EFFECT statement, spline (ORTHOREG),  
429
- EFFECT statement, spline (PHREG), 429
- EFFECT statement, spline (PLS), 429
- EFFECT statement, spline (QUANTREG),  
429
- EFFECT statement, spline (ROBUSTREG),  
429
- EFFECT statement, spline  
(SURVEYLOGISTIC), 429
- EFFECT statement, spline (SURVEYREG),  
429
- DF= option
  - ESTIMATE statement (ORTHOREG), 467
  - ESTIMATE statement (PLM), 467
  - ESTIMATE statement (SURVEYPHREG),  
467
  - ESTIMATE statement (SURVEYREG), 467
  - LSMEANS statement (ORTHOREG), 485
  - LSMEANS statement (PLM), 485
  - LSMEANS statement (SURVEYPHREG),  
485
  - LSMEANS statement (SURVEYREG), 485
  - LSMESTIMATE statement (MIXED), 500
  - LSMESTIMATE statement (ORTHOREG),  
500
  - LSMESTIMATE statement (PLM), 500
  - LSMESTIMATE statement  
(SURVEYPHREG), 500
  - LSMESTIMATE statement (SURVEYREG),  
500
  - SLICE statement (GLIMMIX), 485
  - SLICE statement (MIXED), 485
  - SLICE statement (ORTHOREG), 485
  - SLICE statement (PLM), 485
  - SLICE statement (SURVEYPHREG), 485
  - SLICE statement (SURVEYREG), 485
- DIFF option
  - LSMEANS statement (GENMOD), 485
  - LSMEANS statement (LOGISTIC), 485
  - LSMEANS statement (ORTHOREG), 485
  - LSMEANS statement (PHREG), 485
  - LSMEANS statement (PLM), 485
  - LSMEANS statement (SURVEYLOGISTIC),  
485
  - LSMEANS statement (SURVEYPHREG),  
485
- LSMEANS statement (SURVEYREG), 485
- SLICE statement (GENMOD), 485
- SLICE statement (GLIMMIX), 485
- SLICE statement (LOGISTIC), 485
- SLICE statement (MIXED), 485
- SLICE statement (ORTHOREG), 485
- SLICE statement (PHREG), 485
- SLICE statement (PLM), 485
- SLICE statement (SURVEYLOGISTIC), 485
- SLICE statement (SURVEYPHREG), 485
- SLICE statement (SURVEYREG), 485
- DIVISOR= option
  - ESTIMATE statement (LOGISTIC), 467
  - ESTIMATE statement (ORTHOREG), 467
  - ESTIMATE statement (PHREG), 467
  - ESTIMATE statement (PLM), 467
  - ESTIMATE statement  
(SURVEYLOGISTIC), 467
  - ESTIMATE statement (SURVEYPHREG),  
467
  - ESTIMATE statement (SURVEYREG), 467
  - LSMESTIMATE statement (GENMOD), 501
  - LSMESTIMATE statement (LOGISTIC),  
501
  - LSMESTIMATE statement (MIXED), 501
  - LSMESTIMATE statement (ORTHOREG),  
501
  - LSMESTIMATE statement (PHREG), 501
  - LSMESTIMATE statement (PLM), 501
  - LSMESTIMATE statement  
(SURVEYLOGISTIC), 501
  - LSMESTIMATE statement  
(SURVEYPHREG), 501
  - LSMESTIMATE statement (SURVEYREG),  
501
- E option
  - ESTIMATE statement (LOGISTIC), 467
  - ESTIMATE statement (ORTHOREG), 467
  - ESTIMATE statement (PHREG), 467
  - ESTIMATE statement (PLM), 467
  - ESTIMATE statement  
(SURVEYLOGISTIC), 467
  - ESTIMATE statement (SURVEYPHREG),  
467
  - ESTIMATE statement (SURVEYREG), 467
  - LSMEANS statement (GENMOD), 486
  - LSMEANS statement (LOGISTIC), 486
  - LSMEANS statement (ORTHOREG), 486
  - LSMEANS statement (PHREG), 486
  - LSMEANS statement (PLM), 486
  - LSMEANS statement (SURVEYLOGISTIC),  
486

- LSMEANS statement (SURVEYPHREG), 486
- LSMEANS statement (SURVEYREG), 486
- LSMESTIMATE statement (GENMOD), 501
- LSMESTIMATE statement (LOGISTIC), 501
- LSMESTIMATE statement (MIXED), 501
- LSMESTIMATE statement (ORTHOREG), 501
- LSMESTIMATE statement (PHREG), 501
- LSMESTIMATE statement (PLM), 501
- LSMESTIMATE statement (SURVEYLOGISTIC), 501
- LSMESTIMATE statement (SURVEYPHREG), 501
- LSMESTIMATE statement (SURVEYREG), 501
- SLICE statement (GENMOD), 486
- SLICE statement (GLIMMIX), 486
- SLICE statement (LOGISTIC), 486
- SLICE statement (MIXED), 486
- SLICE statement (ORTHOREG), 486
- SLICE statement (PHREG), 486
- SLICE statement (PLM), 486
- SLICE statement (SURVEYLOGISTIC), 486
- SLICE statement (SURVEYPHREG), 486
- SLICE statement (SURVEYREG), 486
- TEST statement (ORTHOREG), 531
- TEST statement (PLM), 531
- TEST statement (SURVEYPHREG), 531
- TEST statement (SURVEYREG), 531
- E1 option
  - TEST statement (ORTHOREG), 532
  - TEST statement (PLM), 532
  - TEST statement (SURVEYPHREG), 532
  - TEST statement (SURVEYREG), 532
- E2 option
  - TEST statement (ORTHOREG), 532
  - TEST statement (PLM), 532
  - TEST statement (SURVEYPHREG), 532
  - TEST statement (SURVEYREG), 532
- E3 option
  - TEST statement (ORTHOREG), 532
  - TEST statement (PLM), 532
  - TEST statement (SURVEYPHREG), 532
  - TEST statement (SURVEYREG), 532
- EFFECT statement
  - collection effect, 420
  - GLIMMIX procedure, 418
  - GLMSELECT procedure, 418
  - HPMIXED procedure, 418
  - lag effect, 420
  - LOGISTIC procedure, 418
  - multimember effect, 423
  - ORTHOREG procedure, 418
  - PHREG procedure, 418
  - PLS procedure, 418
  - polynomial effect, 425
  - QUANTREG procedure, 418
  - ROBUSTREG procedure, 418
  - spline effect, 428
  - SURVEYLOGISTIC procedure, 418
  - SURVEYREG procedure, 418
- EFFECTPLOT statement
  - GENMOD procedure, 436
  - LOGISTIC procedure, 436
  - ORTHOREG procedure, 436
  - PLM procedure, 436
- ELSM option
  - LSMESTIMATE statement (GENMOD), 501
  - LSMESTIMATE statement (LOGISTIC), 501
  - LSMESTIMATE statement (MIXED), 501
  - LSMESTIMATE statement (ORTHOREG), 501
  - LSMESTIMATE statement (PHREG), 501
  - LSMESTIMATE statement (PLM), 501
  - LSMESTIMATE statement (SURVEYLOGISTIC), 501
  - LSMESTIMATE statement (SURVEYPHREG), 501
  - LSMESTIMATE statement (SURVEYREG), 501
- ESTIMATE statement
  - LOGISTIC procedure, 462
  - ORTHOREG procedure, 462
  - PHREG procedure, 462
  - PLM procedure, 462
  - SURVEYLOGISTIC procedure, 462
  - SURVEYPHREG procedure, 462
  - SURVEYREG procedure, 462
- EXP option
  - ESTIMATE statement (LOGISTIC), 467
  - ESTIMATE statement (PHREG), 467
  - ESTIMATE statement (PLM), 467
  - ESTIMATE statement (SURVEYLOGISTIC), 467
  - LSMEANS statement (GENMOD), 486
  - LSMEANS statement (LOGISTIC), 486
  - LSMEANS statement (PHREG), 486
  - LSMEANS statement (PLM), 486
  - LSMEANS statement (SURVEYLOGISTIC), 486
  - LSMESTIMATE statement (GENMOD), 501
  - LSMESTIMATE statement (LOGISTIC), 501
  - LSMESTIMATE statement (PHREG), 501
  - LSMESTIMATE statement (PLM), 501

- LSMESTIMATE statement
  - (SURVEYLOGISTIC), 501
- SLICE statement (GENMOD), 486
- SLICE statement (GLIMMIX), 486
- SLICE statement (LOGISTIC), 486
- SLICE statement (PHREG), 486
- SLICE statement (PLM), 486
- SLICE statement (SURVEYLOGISTIC), 486
- EXTEND= option
  - EFFECTPLOT statement, 440
- FCONV option
  - NLOPTIONS statement (GLIMMIX), 512
  - NLOPTIONS statement (HPMIXED), 512
  - NLOPTIONS statement (TCALIS), 512
- FCONV2 option
  - NLOPTIONS statement (GLIMMIX), 512
  - NLOPTIONS statement (HPMIXED), 512
  - NLOPTIONS statement (TCALIS), 512
- FSIZE option
  - NLOPTIONS statement (GLIMMIX), 513
  - NLOPTIONS statement (HPMIXED), 513
  - NLOPTIONS statement (TCALIS), 513
- FTOL option
  - NLOPTIONS statement (GLIMMIX), 512
  - NLOPTIONS statement (HPMIXED), 512
  - NLOPTIONS statement (TCALIS), 512
- FTOL2 option
  - NLOPTIONS statement (GLIMMIX), 512
  - NLOPTIONS statement (HPMIXED), 512
  - NLOPTIONS statement (TCALIS), 512
- GCONV option
  - NLOPTIONS statement (GLIMMIX), 513
  - NLOPTIONS statement (HPMIXED), 513
  - NLOPTIONS statement (TCALIS), 513
- GCONV2 option
  - NLOPTIONS statement (GLIMMIX), 513
  - NLOPTIONS statement (HPMIXED), 513
  - NLOPTIONS statement (TCALIS), 513
- GENMOD procedure, EFFECTPLOT statement
  - ALPHA= option, 438
  - AT option, 438
  - ATLEN= option, 439
  - ATORDER= option, 439
  - CLI option, 440
  - CLM option, 440
  - EXTEND= option, 440
  - GRIDSIZE= option, 440
  - ILINK option, 440
  - INDIVIDUAL option, 440
  - LIMITS option, 440
  - LINK option, 440
  - MOFF option, 441
  - NCOLS= option, 441
  - NOCLI option, 441
  - NOCLM option, 441
  - NOLIMITS option, 441
  - NOOBS option, 441
  - NROWS= option, 441
  - OBS option, 441
  - PLOTBY= option, 444
  - PLOTBYLEN= option, 445
  - POLYBAR option, 445
  - PREDLABEL= option, 445
  - SHOWCLEGEND option, 445
  - SLICEBY= option, 445
  - SMOOTH option, 445
  - UNPACK option, 445
  - X= option, 445
  - Y= option, 446
  - YRANGE= option, 446
- GENMOD procedure, LSMEANS statement
  - ADJUST= option, 482
  - ALPHA= option, 484
  - AT= option, 484
  - BYLEVEL option, 485
  - CL option, 485
  - CORR option, 485
  - COV option, 485
  - DIFF option, 485
  - E option, 486
  - EXP option, 486
  - ILINK option, 487
  - LINES option, 487
  - MEANS or NOMEANS option, 487
  - OBSMARGINS= option, 488
  - ODDSRATIO option, 487
  - ODS graph names, 494
  - ODS table names, 494
  - PDIFF option, 488
  - PLOTS= option, 488
  - SEED= option, 492
  - SINGULAR= option, 492
  - STEPDOWN option, 493
- GENMOD procedure, LSMESTIMATE statement
  - ADJUST= option, 499
  - ALPHA= option, 499
  - AT= option, 499
  - BYLEVEL option, 499
  - CATEGORY= option, 500
  - CL option, 500
  - CORR option, 500
  - COV option, 500
  - DIVISOR= option, 501
  - E option, 501
  - ELSM option, 501
  - EXP option, 501

- ILINK option, 501
- JOINT option, 502
- LOWER option, 503
- OBSMARGINS= option, 503
- ODS graph names, 507
- ODS table names, 507
- PLOTS= option, 504
- SEED= option, 505
- SINGULAR= option, 505
- STEPDOWN option, 505
- TESTVALUE= option, 506
- UPPER option, 507
- GENMOD procedure, SLICE statement
  - ADJUST= option, 482
  - ALPHA= option, 484
  - AT= option, 484
  - BYLEVEL option, 485
  - CL option, 485
  - CORR option, 485
  - COV option, 485
  - DIFF option, 485
  - E option, 486
  - EXP option, 486
  - ILINK option, 487
  - LINES option, 487
  - MEANS or NOMEANS option, 487
  - NOF option, 528
  - OBSMARGINS= option, 488
  - ODDSRATIO option, 487
  - ODS table names, 529
  - PDIFF option, 488
  - PLOTS= option, 488
  - SEED= option, 492
  - SIMPLE= option, 528
  - SINGULAR= option, 492
  - SLICEBY= option, 528
  - STEPDOWN option, 493
- GLIMMIX procedure, EFFECT statement
  - BASIS option (spline), 429
  - collection effect, 420
  - DATABOUNDARY option (spline), 429
  - DEGREE option (polynomial), 425
  - DEGREE option (spline), 429
  - DESIGNROLE option (lag), 422
  - DETAILS option (lag), 422
  - DETAILS option (multimember), 424
  - DETAILS option (polynomial), 425
  - DETAILS option (spline), 429
  - KNOTMAX option (spline), 429
  - KNOTMETHOD option (spline), 430
  - KNOTMIN option (spline), 431
  - LABELSTYLE option (polynomial), 425
  - lag effect, 420
  - MDEGREE option (polynomial), 426
  - multimember effect, 423
  - NLAG option (lag), 422
  - NOEFFECT option (multimember), 424
  - NOSEPARATE option (polynomial), 426
  - PERIOD option (lag), 422
  - polynomial effect, 425
  - SEPARATE option (spline), 431
  - spline effect, 428
  - STANDARDIZE option (polynomial), 426
  - WITHIN option (lag), 422
- GLIMMIX procedure, NLOPTIONS statement
  - ABSCONV option, 510
  - ABSFCONV option, 510
  - ABSGCONV option, 511
  - ABSGTOL option, 511
  - ABSTOL option, 510
  - ABSXCONV option, 511
  - ABSXTOL option, 511
  - ASINGULAR= option, 511
  - DAMPSTEP option, 512
  - FCONV option, 512
  - FCONV2 option, 512
  - FSIZE option, 513
  - FTOL option, 512
  - FTOL2 option, 512
  - GCONV option, 513
  - GCONV2 option, 513
  - GTOL option, 513
  - GTOL2 option, 513
  - HESCAL option, 514
  - HS option, 514
  - INHES option, 514
  - INHESSIAN option, 514
  - INSTEP option, 515
  - LCDEACT= option, 515
  - LCEPSILON= option, 515
  - LCSINGULAR= option, 516
  - LINESEARCH option, 516
  - LIS option, 516
  - LSP option, 516
  - LSPRECISION option, 516
  - MAXFU option, 517
  - MAXFUNC option, 517
  - MAXIT option, 517
  - MAXITER option, 517
  - MAXSTEP option, 517
  - MAXTIME option, 518
  - MINIT option, 518
  - MINITER option, 518
  - MSINGULAR= option, 518
  - REST option, 518
  - RESTART option, 518
  - SINGULAR= option, 519
  - SOCKET option, 519



- TECH option, 519
- TECHNIQUE option, 519
- UPD option, 520
- UPDATE option, 520
- VSINGULAR= option, 521
- XCONV option, 521
- XSIZE option, 521
- XTOL option, 521
- GLIMMIX procedure, SLICE statement
  - ADJDFE= option, 481
  - ADJUST= option, 482
  - ALPHA= option, 484
  - AT= option, 484
  - BYLEVEL option, 485
  - CL option, 485
  - CORR option, 485
  - COV option, 485
  - DF= option, 485
  - DIFF option, 485
  - E option, 486
  - EXP option, 486
  - ILINK option, 487
  - LINES option, 487
  - MEANS or NOMEANS option, 487
  - NOF option, 528
  - OBSMARGINS= option, 488
  - ODDSRATIO option, 487
  - ODS graph names, 494
  - ODS table names, 529
  - PDIFF option, 488
  - PLOTS= option, 488
  - SEED= option, 492
  - SIMPLE= option, 528
  - SINGULAR= option, 492
  - SLICEBY= option, 528
  - STEPDOWN option, 493
- GLMSELECT procedure, EFFECT statement
  - BASIS option (spline), 429
  - collection effect, 420
  - DATABOUNDARY option (spline), 429
  - DEGREE option (polynomial), 425
  - DEGREE option (spline), 429
  - DESIGNROLE option (lag), 422
  - DETAILS option (lag), 422
  - DETAILS option (multimember), 424
  - DETAILS option (polynomial), 425
  - DETAILS option (spline), 429
  - KNOTMAX option (spline), 429
  - KNOTMETHOD option (spline), 430
  - KNOTMIN option (spline), 431
  - LABELSTYLE option (polynomial), 425
  - lag effect, 420
  - MDEGREE option (polynomial), 426
  - multimember effect, 423
- NLAG option (lag), 422
- NOEFFECT option (multimember), 424
- NOSEPARATE option (polynomial), 426
- PERIOD option (lag), 422
- polynomial effect, 425
- SEPARATE option (spline), 431
- spline effect, 428
- SPLIT option (spline), 431
- STANDARDIZE option (polynomial), 426
- WITHIN option (lag), 422
- GRIDSIZE= option
  - EFFECTPLOT statement, 440
- GTOL option
  - NLOPTIONS statement (GLIMMIX), 513
  - NLOPTIONS statement (HPMIXED), 513
  - NLOPTIONS statement (TCALIS), 513
- GTOL2 option
  - NLOPTIONS statement (GLIMMIX), 513
  - NLOPTIONS statement (HPMIXED), 513
  - NLOPTIONS statement (TCALIS), 513
- HESCAL option
  - NLOPTIONS statement (GLIMMIX), 514
  - NLOPTIONS statement (HPMIXED), 514
  - NLOPTIONS statement (TCALIS), 514
- HPMIXED procedure, EFFECT statement
  - BASIS option (spline), 429
  - collection effect, 420
  - DATABOUNDARY option (spline), 429
  - DEGREE option (polynomial), 425
  - DEGREE option (spline), 429
  - DESIGNROLE option (lag), 422
  - DETAILS option (lag), 422
  - DETAILS option (multimember), 424
  - DETAILS option (polynomial), 425
  - DETAILS option (spline), 429
  - KNOTMAX option (spline), 429
  - KNOTMETHOD option (spline), 430
  - KNOTMIN option (spline), 431
  - LABELSTYLE option (polynomial), 425
  - lag effect, 420
  - MDEGREE option (polynomial), 426
  - multimember effect, 423
  - NLAG option (lag), 422
  - NOEFFECT option (multimember), 424
  - NOSEPARATE option (polynomial), 426
  - PERIOD option (lag), 422
  - polynomial effect, 425
  - SEPARATE option (spline), 431
  - spline effect, 428
  - SPLIT option (spline), 431
  - STANDARDIZE option (polynomial), 426
  - WITHIN option (lag), 422
- HPMIXED procedure, NLOPTIONS statement

- ABSCONV option, 510
- ABSFCONV option, 510
- ABSGCONV option, 511
- ABSGTOL option, 511
- ABSTOL option, 510
- ABSXCONV option, 511
- ABSXTOL option, 511
- ASINGULAR= option, 511
- FCONV option, 512
- FCONV2 option, 512
- FSIZE option, 513
- FTOL option, 512
- FTOL2 option, 512
- GCONV option, 513
- GCONV2 option, 513
- GTOL option, 513
- GTOL2 option, 513
- HESCAL option, 514
- HS option, 514
- INHESSIAN option, 514
- INSTEP option, 515
- LCDEACT= option, 515
- LCEPSILON= option, 515
- LCSINGULAR= option, 516
- LINESEARCH option, 516
- LIS option, 516
- LSP option, 516
- LSPRECISION option, 516
- MAXFU option, 517
- MAXFUNC option, 517
- MAXIT option, 517
- MAXITER option, 517
- MAXSTEP option, 517
- MAXTIME option, 518
- MINIT option, 518
- MINITER option, 518
- MSINGULAR= option, 518
- REST option, 518
- RESTART option, 518
- SINGULAR= option, 519
- SOCKET option, 519
- TECH option, 519
- TECHNIQUE option, 519
- UPD option, 520
- XSIZE option, 521
- XTOL option, 521
- HS option
  - NLOPTIONS statement (GLIMMIX), 514
  - NLOPTIONS statement (HPMIXED), 514
  - NLOPTIONS statement (TCALIS), 514
- HTYPE= option
  - TEST statement (ORTHOREG), 532
  - TEST statement (PLM), 532
  - TEST statement (SURVEYPHREG), 532
- TEST statement (SURVEYREG), 532
- ILINK option
  - EFFECTPLOT statement, 440
  - ESTIMATE statement (LOGISTIC), 467
  - ESTIMATE statement (PLM), 467
  - ESTIMATE statement
    - (SURVEYLOGISTIC), 467
  - LSMEANS statement (GENMOD), 487
  - LSMEANS statement (LOGISTIC), 487
  - LSMEANS statement (PLM), 487
  - LSMEANS statement (SURVEYLOGISTIC), 487
  - LSMESTIMATE statement (GENMOD), 501
  - LSMESTIMATE statement (LOGISTIC), 501
  - LSMESTIMATE statement (PLM), 501
  - LSMESTIMATE statement
    - (SURVEYLOGISTIC), 501
  - SLICE statement (GENMOD), 487
  - SLICE statement (GLIMMIX), 487
  - SLICE statement (LOGISTIC), 487
  - SLICE statement (PLM), 487
  - SLICE statement (SURVEYLOGISTIC), 487
- INDIVIDUAL option
  - EFFECTPLOT statement, 440
- INHESS option
  - NLOPTIONS statement (GLIMMIX), 514
  - NLOPTIONS statement (HPMIXED), 514
  - NLOPTIONS statement (TCALIS), 514
- INHESSIAN option
  - NLOPTIONS statement (GLIMMIX), 514
  - NLOPTIONS statement (HPMIXED), 514
  - NLOPTIONS statement (TCLAIS), 514
- INSTEP option
  - NLOPTIONS statement (GLIMMIX), 515
  - NLOPTIONS statement (HPMIXED), 515
  - NLOPTIONS statement (TCALIS), 515
- INTERCEPT option
  - TEST statement (ORTHOREG), 532
  - TEST statement (PLM), 532
  - TEST statement (SURVEYPHREG), 532
  - TEST statement (SURVEYREG), 532
- JOINT option
  - ESTIMATE statement (LOGISTIC), 468
  - ESTIMATE statement (ORTHOREG), 468
  - ESTIMATE statement (PHREG), 468
  - ESTIMATE statement (PLM), 468
  - ESTIMATE statement
    - (SURVEYLOGISTIC), 468
  - ESTIMATE statement (SURVEYPHREG), 468
  - ESTIMATE statement (SURVEYREG), 468

- LSMESTIMATE statement (GENMOD), 502
- LSMESTIMATE statement (LOGISTIC), 502
- LSMESTIMATE statement (MIXED), 502
- LSMESTIMATE statement (ORTHOREG), 502
- LSMESTIMATE statement (PHREG), 502
- LSMESTIMATE statement (PLM), 502
- LSMESTIMATE statement (SURVEYLOGISTIC), 502
- LSMESTIMATE statement (SURVEYPHREG), 502
- LSMESTIMATE statement (SURVEYREG), 502
- KNOTMAX option
  - EFFECT statement, spline (GLIMMIX), 429
  - EFFECT statement, spline (GLMSELECT), 429
  - EFFECT statement, spline (HPMIXED), 429
  - EFFECT statement, spline (LOGISTIC), 429
  - EFFECT statement, spline (ORTHOREG), 429
  - EFFECT statement, spline (PHREG), 429
  - EFFECT statement, spline (PLS), 429
  - EFFECT statement, spline (QUANTREG), 429
  - EFFECT statement, spline (ROBUSTREG), 429
  - EFFECT statement, spline (SURVEYLOGISTIC), 429
  - EFFECT statement, spline (SURVEYREG), 429
- KNOTMETHOD option
  - EFFECT statement, spline (GLIMMIX), 430
  - EFFECT statement, spline (GLMSELECT), 430
  - EFFECT statement, spline (HPMIXED), 430
  - EFFECT statement, spline (LOGISTIC), 430
  - EFFECT statement, spline (ORTHOREG), 430
  - EFFECT statement, spline (PHREG), 430
  - EFFECT statement, spline (PLS), 430
  - EFFECT statement, spline (QUANTREG), 430
  - EFFECT statement, spline (ROBUSTREG), 430
  - EFFECT statement, spline (SURVEYLOGISTIC), 430
  - EFFECT statement, spline (SURVEYREG), 430
- KNOTMIN option
  - EFFECT statement, spline (GLIMMIX), 431
- EFFECT statement, spline (GLMSELECT), 431
- EFFECT statement, spline (HPMIXED), 431
- EFFECT statement, spline (LOGISTIC), 431
- EFFECT statement, spline (ORTHOREG), 431
- EFFECT statement, spline (PHREG), 431
- EFFECT statement, spline (PLS), 431
- EFFECT statement, spline (QUANTREG), 431
- EFFECT statement, spline (ROBUSTREG), 431
- EFFECT statement, spline (SURVEYLOGISTIC), 431
- EFFECT statement, spline (SURVEYREG), 431
- LABELSTYLE option
  - EFFECT statement, polynomial (GLIMMIX), 425
  - EFFECT statement, polynomial (GLMSELECT), 425
  - EFFECT statement, polynomial (HPMIXED), 425
  - EFFECT statement, polynomial (LOGISTIC), 425
  - EFFECT statement, polynomial (ORTHOREG), 425
  - EFFECT statement, polynomial (PHREG), 425
  - EFFECT statement, polynomial (PLS), 425
  - EFFECT statement, polynomial (ROBUSTREG), 425
  - EFFECT statement, polynomial (SURVEYLOGISTIC), 425
  - EFFECT statement, polynomial (SURVEYREG), 425
- LCDEACT= option
  - NLOPTIONS statement (GLIMMIX), 515
  - NLOPTIONS statement (HPMIXED), 515
  - NLOPTIONS statement (TCALIS), 515
- LCEPSILON= option
  - NLOPTIONS statement (GLIMMIX), 515
  - NLOPTIONS statement (HPMIXED), 515
  - NLOPTIONS statement (TCALIS), 515
- LCSINGULAR= option
  - NLOPTIONS statement (GLIMMIX), 516
  - NLOPTIONS statement (HPMIXED), 516
  - NLOPTIONS statement (TCALIS), 516
- LIMITS option
  - EFFECTPLOT statement, 440
- LINES option
  - LSMEANS statement (GENMOD), 487
  - LSMEANS statement (LOGISTIC), 487



- LSMEANS statement (ORTHOREG), 487
- LSMEANS statement (PHREG), 487
- LSMEANS statement (PLM), 487
- LSMEANS statement (SURVEYLOGISTIC), 487
- LSMEANS statement (SURVEYPHREG), 487
- LSMEANS statement (SURVEYREG), 487
- SLICE statement (GENMOD), 487
- SLICE statement (GLIMMIX), 487
- SLICE statement (LOGISTIC), 487
- SLICE statement (MIXED), 487
- SLICE statement (ORTHOREG), 487
- SLICE statement (PHREG), 487
- SLICE statement (PLM), 487
- SLICE statement (SURVEYLOGISTIC), 487
- SLICE statement (SURVEYPHREG), 487
- SLICE statement (SURVEYREG), 487
- LINESEARCH option
  - NLOPTIONS statement (GLIMMIX), 516
  - NLOPTIONS statement (HPMIXED), 516
  - NLOPTIONS statement (TCALIS), 516
- LINK option
  - EFFECTPLOT statement, 440
- LIS option
  - NLOPTIONS statement (GLIMMIX), 516
  - NLOPTIONS statement (HPMIXED), 516
  - NLOPTIONS statement (TCALIS), 516
- LOGISTIC procedure, EFFECT statement
  - BASIS option (spline), 429
  - collection effect, 420
  - DATABOUNDARY option (spline), 429
  - DEGREE option (polynomial), 425
  - DEGREE option (spline), 429
  - DESIGNROLE option (lag), 422
  - DETAILS option (lag), 422
  - DETAILS option (multimember), 424
  - DETAILS option (polynomial), 425
  - DETAILS option (spline), 429
  - KNOTMAX option (spline), 429
  - KNOTMETHOD option (spline), 430
  - KNOTMIN option (spline), 431
  - LABELSTYLE option (polynomial), 425
  - lag effect, 420
  - MDEGREE option (polynomial), 426
  - multimember effect, 423
  - NLAG option (lag), 422
  - NOEFFECT option (multimember), 424
  - NOSEPARATE option (polynomial), 426
  - PERIOD option (lag), 422
  - polynomial effect, 425
  - SEPARATE option (spline), 431
  - spline effect, 428
  - SPLIT option (spline), 431
  - STANDARDIZE option (polynomial), 426
  - WITHIN option (lag), 422
- LOGISTIC procedure, EFFECTPLOT statement
  - ALPHA= option, 438
  - AT option, 438
  - ATLEN= option, 439
  - ATORDER= option, 439
  - CLI option, 440
  - CLM option, 440
  - EXTEND= option, 440
  - GRIDSIZE= option, 440
  - ILINK option, 440
  - INDIVIDUAL option, 440
  - LIMITS option, 440
  - LINK option, 440
  - MOFF option, 441
  - NCOLS= option, 441
  - NOCLI option, 441
  - NOCLM option, 441
  - NOLIMITS option, 441
  - NOOBS option, 441
  - NROWS= option, 441
  - OBS option, 441
  - PLOTBY= option, 444
  - PLOTBYLEN= option, 445
  - POLYBAR option, 445
  - PREDLABEL= option, 445
  - SHOWCLEGEND option, 445
  - SLICEBY= option, 445
  - SMOOTH option, 445
  - UNPACK option, 445
  - X= option, 445
  - Y= option, 446
  - YRANGE= option, 446
- LOGISTIC procedure, ESTIMATE statement
  - ADJUST= option, 465
  - ALPHA= option, 465
  - CATEGORY= option, 465
  - CL option, 466
  - CORR option, 466
  - COV option, 466
  - DIVISOR= option, 467
  - E option, 467
  - EXP option, 467
  - ILINK option, 467
  - JOINT option, 468
  - LOWER option, 469
  - NOFILL option, 469
  - ODS table names, 477
  - SEED= option, 471
  - SINGULAR= option, 471
  - STEPDOWN option, 471
  - TESTVALUE option, 472
  - UPPER option, 473

LOGISTIC procedure, LSMEANS statement

ADJUST= option, [482](#)  
ALPHA= option, [484](#)  
AT= option, [484](#)  
BYLEVEL option, [485](#)  
CL option, [485](#)  
CORR option, [485](#)  
COV option, [485](#)  
DIFF option, [485](#)  
E option, [486](#)  
EXP option, [486](#)  
ILINK option, [487](#)  
LINES option, [487](#)  
MEANS or NOMEANS option, [487](#)  
OBSMARGINS= option, [488](#)  
ODDSRATIO option, [487](#)  
ODS graph names, [494](#)  
ODS table names, [494](#)  
PDIFF option, [488](#)  
PLOTS= option, [488](#)  
SEED= option, [492](#)  
SINGULAR= option, [492](#)  
STEPDOWN option, [493](#)

LOGISTIC procedure, LSMESTIMATE statement

ADJUST= option, [499](#)  
ALPHA= option, [499](#)  
AT= option, [499](#)  
BYLEVEL option, [499](#)  
CATEGORY= option, [500](#)  
CL option, [500](#)  
CORR option, [500](#)  
COV option, [500](#)  
DIVISOR= option, [501](#)  
E option, [501](#)  
ELSM option, [501](#)  
EXP option, [501](#)  
ILINK option, [501](#)  
JOINT option, [502](#)  
LOWER option, [503](#)  
OBSMARGINS= option, [503](#)  
ODS table names, [507](#)  
PLOTS= option, [504](#)  
SEED= option, [505](#)  
SINGULAR= option, [505](#)  
STEPDOWN option, [505](#)  
TESTVALUE= option, [506](#)  
UPPER option, [507](#)

LOGISTIC procedure, SLICE statement

ADJUST= option, [482](#)  
ALPHA= option, [484](#)  
AT= option, [484](#)  
BYLEVEL option, [485](#)  
CL option, [485](#)  
CORR option, [485](#)

COV option, [485](#)  
DIFF option, [485](#)  
E option, [486](#)  
EXP option, [486](#)  
ILINK option, [487](#)  
LINES option, [487](#)  
MEANS or NOMEANS option, [487](#)  
NOF option, [528](#)  
OBSMARGINS= option, [488](#)  
ODDSRATIO option, [487](#)  
ODS table names, [529](#)  
PDIFF option, [488](#)  
PLOTS= option, [488](#)  
SEED= option, [492](#)  
SIMPLE= option, [528](#)  
SINGULAR= option, [492](#)  
SLICEBY= option, [528](#)  
STEPDOWN option, [493](#)

LOWER option

ESTIMATE statement (LOGISTIC), [469](#)  
ESTIMATE statement (ORTHOREG), [469](#)  
ESTIMATE statement (PHREG), [469](#)  
ESTIMATE statement (PLM), [469](#)  
ESTIMATE statement  
(SURVEYLOGISTIC), [469](#)  
ESTIMATE statement (SURVEYPHREG),  
[469](#)  
ESTIMATE statement (SURVEYREG), [469](#)  
LSMESTIMATE statement (GENMOD), [503](#)  
LSMESTIMATE statement (LOGISTIC),  
[503](#)  
LSMESTIMATE statement (MIXED), [503](#)  
LSMESTIMATE statement (ORTHOREG),  
[503](#)  
LSMESTIMATE statement (PHREG), [503](#)  
LSMESTIMATE statement (PLM), [503](#)  
LSMESTIMATE statement  
(SURVEYLOGISTIC), [503](#)  
LSMESTIMATE statement  
(SURVEYPHREG), [503](#)  
LSMESTIMATE statement (SURVEYREG),  
[503](#)

LSMEANS statement

GENMOD procedure, [479](#)  
LOGISTIC procedure, [479](#)  
ORTHOREG procedure, [479](#)  
PHREG procedure, [479](#)  
PLM procedure, [479](#)  
SURVEYLOGISTIC procedure, [479](#)  
SURVEYPHREG procedure, [479](#)  
SURVEYREG procedure, [479](#)

LSMESTIMATE statement

GENMOD procedure, [496](#)  
LOGISTIC procedure, [496](#)

- MIXED procedure, 496
- ORTHOREG procedure, 496
- PHREG procedure, 496
- PLM procedure, 496
- SURVEYLOGISTIC procedure, 496
- SURVEYPHREG procedure, 496
- SURVEYREG procedure, 496
- LSP option
  - NLOPTIONS statement (GLIMMIX), 516
  - NLOPTIONS statement (HPMIXED), 516
  - NLOPTIONS statement (TCALIS), 516
- LSPRECISSION option
  - NLOPTIONS statement (GLIMMIX), 516
  - NLOPTIONS statement (HPMIXED), 516
  - NLOPTIONS statement (TCALIS), 516
- MAXFU option
  - NLOPTIONS statement (GLIMMIX), 517
  - NLOPTIONS statement (HPMIXED), 517
  - NLOPTIONS statement (TCALIS), 517
- MAXFUNC option
  - NLOPTIONS statement (GLIMMIX), 517
  - NLOPTIONS statement (HPMIXED), 517
  - NLOPTIONS statement (TCALIS), 517
- MAXIT option
  - NLOPTIONS statement (GLIMMIX), 517
  - NLOPTIONS statement (HPMIXED), 517
  - NLOPTIONS statement (TCALIS), 517
- MAXITER option
  - NLOPTIONS statement (GLIMMIX), 517
  - NLOPTIONS statement (HPMIXED), 517
  - NLOPTIONS statement (TCALIS), 517
- MAXSTEP option
  - NLOPTIONS statement (GLIMMIX), 517
  - NLOPTIONS statement (HPMIXED), 517
  - NLOPTIONS statement (TCALIS), 517
- MAXTIME option
  - NLOPTIONS statement (GLIMMIX), 518
  - NLOPTIONS statement (HPMIXED), 518
  - NLOPTIONS statement (TCALIS), 518
- MDEGREE option
  - EFFECT statement, polynomial (GLIMMIX), 426
  - EFFECT statement, polynomial (GLMSELECT), 426
  - EFFECT statement, polynomial (HPMIXED), 426
  - EFFECT statement, polynomial (LOGISTIC), 426
  - EFFECT statement, polynomial (ORTHOREG), 426
  - EFFECT statement, polynomial (PHREG), 426
  - EFFECT statement, polynomial (PLS), 426
- EFFECT statement, polynomial (ROBUSTREG), 426
- EFFECT statement, polynomial (SURVEYLOGISTIC), 426
- EFFECT statement, polynomial (SURVEYREG), 426
- MEANS or NOMEANS option
  - LSMEANS statement (GENMOD), 487
  - LSMEANS statement (LOGISTIC), 487
  - LSMEANS statement (ORTHOREG), 487
  - LSMEANS statement (PHREG), 487
  - LSMEANS statement (PLM), 487
  - LSMEANS statement (SURVEYLOGISTIC), 487
  - LSMEANS statement (SURVEYPHREG), 487
  - LSMEANS statement (SURVEYREG), 487
  - SLICE statement (GENMOD), 487
  - SLICE statement (GLIMMIX), 487
  - SLICE statement (LOGISTIC), 487
  - SLICE statement (MIXED), 487
  - SLICE statement (ORTHOREG), 487
  - SLICE statement (PHREG), 487
  - SLICE statement (PLM), 487
  - SLICE statement (SURVEYLOGISTIC), 487
  - SLICE statement (SURVEYPHREG), 487
  - SLICE statement (SURVEYREG), 487
- MINIT option
  - NLOPTIONS statement (GLIMMIX), 518
  - NLOPTIONS statement (HPMIXED), 518
  - NLOPTIONS statement (TCALIS), 518
- MINITER option
  - NLOPTIONS statement (GLIMMIX), 518
  - NLOPTIONS statement (HPMIXED), 518
  - NLOPTIONS statement (TCALIS), 518
- MIXED procedure, LSMESIMATE statement
  - ADJDFE= option, 498
  - ADJUST= option, 499
  - ALPHA= option, 499
  - AT= option, 499
  - BYLEVEL option, 499
  - CHISQ option, 500
  - CL option, 500
  - CORR option, 500
  - COV option, 500
  - DF= option, 500
  - DIVISOR= option, 501
  - E option, 501
  - ELSM option, 501
  - JOINT option, 502
  - LOWER option, 503
  - OBSMARGINS= option, 503
  - ODS table names, 507
  - PLOTS= option, 504

- SEED= option, [505](#)
- SINGULAR= option, [505](#)
- STEPPDOWN option, [505](#)
- TESTVALUE= option, [506](#)
- UPPER option, [507](#)
- MIXED procedure, SLICE statement
  - ADJDFE= option, [481](#)
  - ADJUST= option, [482](#)
  - ALPHA= option, [484](#)
  - AT= option, [484](#)
  - BYLEVEL option, [485](#)
  - CL option, [485](#)
  - CORR option, [485](#)
  - COV option, [485](#)
  - DF= option, [485](#)
  - DIFF option, [485](#)
  - E option, [486](#)
  - LINES option, [487](#)
  - MEANS or NOMEANS option, [487](#)
  - NOF option, [528](#)
  - OBSMARGINS= option, [488](#)
  - ODS graph names, [494](#)
  - ODS table names, [529](#)
  - PDIFF option, [488](#)
  - PLOTS= option, [488](#)
  - SEED= option, [492](#)
  - SIMPLE= option, [528](#)
  - SINGULAR= option, [492](#)
  - SLICEBY= option, [528](#)
  - STEPPDOWN option, [493](#)
- MOFF option
  - EFFECTPLOT statement, [441](#)
- MSINGULAR= option
  - NLOPTIONS statement (GLIMMIX), [518](#)
  - NLOPTIONS statement (HPMIXED), [518](#)
  - NLOPTIONS statement (TCALIS), [518](#)
- NCOLS= option
  - EFFECTPLOT statement, [441](#)
- NLAG option
  - EFFECT statement, lag (GLIMMIX), [422](#)
  - EFFECT statement, lag (GLMSELECT), [422](#)
  - EFFECT statement, lag (HPMIXED), [422](#)
  - EFFECT statement, lag (LOGISTIC), [422](#)
  - EFFECT statement, lag (ORTHOREG), [422](#)
  - EFFECT statement, lag (PHREG), [422](#)
  - EFFECT statement, lag (PLS), [422](#)
  - EFFECT statement, lag (ROBUSTREG), [422](#)
  - EFFECT statement, lag
    - (SURVEYLOGISTIC), [422](#)
  - EFFECT statement, lag (SURVEYREG), [422](#)
- NLMIXED procedure, NLOPTIONS statement
  - VSINGULAR= option, [521](#)
- NLOPTIONS statement
  - GLIMMIX procedure, [509](#)
  - HPMIXED procedure, [509](#)
  - TCALIS procedure, [509](#)
- NOCLI option
  - EFFECTPLOT statement, [441](#)
- NOCLM option
  - EFFECTPLOT statement, [441](#)
- NOEFFECT option
  - EFFECT statement, multimember (GLIMMIX), [424](#)
  - EFFECT statement, multimember (GLMSELECT), [424](#)
  - EFFECT statement, multimember (HPMIXED), [424](#)
  - EFFECT statement, multimember (LOGISTIC), [424](#)
  - EFFECT statement, multimember (ORTHOREG), [424](#)
  - EFFECT statement, multimember (PHREG), [424](#)
  - EFFECT statement, multimember (PLS), [424](#)
  - EFFECT statement, multimember (ROBUSTREG), [424](#)
  - EFFECT statement, multimember (SURVEYLOGISTIC), [424](#)
  - EFFECT statement, multimember (SURVEYREG), [424](#)
- NOF option
  - SLICE statement (GENMOD), [528](#)
  - SLICE statement (GLIMMIX), [528](#)
  - SLICE statement (LOGISTIC), [528](#)
  - SLICE statement (MIXED), [528](#)
  - SLICE statement (ORTHOREG), [528](#)
  - SLICE statement (PHREG), [528](#)
  - SLICE statement (PLM), [528](#)
  - SLICE statement (SURVEYLOGISTIC), [528](#)
  - SLICE statement (SURVEYPHREG), [528](#)
  - SLICE statement (SURVEYREG), [528](#)
- NOFILL option
  - ESTIMATE statement (LOGISTIC), [469](#)
  - ESTIMATE statement (ORTHOREG), [469](#)
  - ESTIMATE statement (PHREG), [469](#)
  - ESTIMATE statement (PLM), [469](#)
  - ESTIMATE statement
    - (SURVEYLOGISTIC), [469](#)
  - ESTIMATE statement (SURVEYPHREG), [469](#)
  - ESTIMATE statement (SURVEYREG), [469](#)
- NOLIMITS option
  - EFFECTPLOT statement, [441](#)
- NOOBS option
  - EFFECTPLOT statement, [441](#)
- NOSEPARATE option

- EFFECT statement, polynomial (GLIMMIX), 426
- EFFECT statement, polynomial (GLMSELECT), 426
- EFFECT statement, polynomial (HPMIXED), 426
- EFFECT statement, polynomial (LOGISTIC), 426
- EFFECT statement, polynomial (ORTHOREG), 426
- EFFECT statement, polynomial (PHREG), 426
- EFFECT statement, polynomial (PLS), 426
- EFFECT statement, polynomial (ROBUSTREG), 426
- EFFECT statement, polynomial (SURVEYLOGISTIC), 426
- EFFECT statement, polynomial (SURVEYREG), 426
- NROWS= option
  - EFFECTPLOT statement, 441
- OBS option
  - EFFECTPLOT statement, 441
- OBSMARGINS= option
  - LSMEANS statement (GENMOD), 488
  - LSMEANS statement (LOGISTIC), 488
  - LSMEANS statement (ORTHOREG), 488
  - LSMEANS statement (PHREG), 488
  - LSMEANS statement (PLM), 488
  - LSMEANS statement (SURVEYLOGISTIC), 488
  - LSMEANS statement (SURVEYPHREG), 488
  - LSMEANS statement (SURVEYREG), 488
  - LSMESTIMATE statement (GENMOD), 503
  - LSMESTIMATE statement (LOGISTIC), 503
  - LSMESTIMATE statement (MIXED), 503
  - LSMESTIMATE statement (ORTHOREG), 503
  - LSMESTIMATE statement (PHREG), 503
  - LSMESTIMATE statement (PLM), 503
  - LSMESTIMATE statement (SURVEYLOGISTIC), 503
  - LSMESTIMATE statement (SURVEYPHREG), 503
  - LSMESTIMATE statement (SURVEYREG), 503
  - SLICE statement (GENMOD), 488
  - SLICE statement (GLIMMIX), 488
  - SLICE statement (LOGISTIC), 488
  - SLICE statement (MIXED), 488
  - SLICE statement (ORTHOREG), 488
  - SLICE statement (PHREG), 488
  - SLICE statement (PLM), 488
  - SLICE statement (SURVEYLOGISTIC), 488
  - SLICE statement (SURVEYPHREG), 488
  - SLICE statement (SURVEYREG), 488
- ODDSRATIO option
  - LSMEANS statement (GENMOD), 487
  - LSMEANS statement (LOGISTIC), 487
  - LSMEANS statement (PLM), 487
  - LSMEANS statement (SURVEYLOGISTIC), 487
  - SLICE statement (GENMOD), 487
  - SLICE statement (GLIMMIX), 487
  - SLICE statement (LOGISTIC), 487
  - SLICE statement (PLM), 487
  - SLICE statement (SURVEYLOGISTIC), 487
- ODS graph names
  - ESTIMATE statement (PHREG), 478
  - ESTIMATE statement (PLM), 478
  - LSMEANS statement (GENMOD), 494
  - LSMEANS statement (LOGISTIC), 494
  - LSMEANS statement (ORTHOREG), 494
  - LSMEANS statement (PHREG), 494
  - LSMEANS statement (PLM), 494
  - LSMEANS statement (SURVEYLOGISTIC), 494
  - LSMEANS statement (SURVEYPHREG), 494
  - LSMEANS statement (SURVEYREG), 494
  - LSMESTIMATE statement (GENMOD), 507
  - LSMESTIMATE statement (PHREG), 507
  - LSMESTIMATE statement (PLM), 507
  - SLICE statement (GENMOD), 494
  - SLICE statement (GLIMMIX), 494
  - SLICE statement (LOGISTIC), 494
  - SLICE statement (MIXED), 494
  - SLICE statement (ORTHOREG), 494
  - SLICE statement (PHREG), 494
  - SLICE statement (PLM), 494
  - SLICE statement (SURVEYLOGISTIC), 494
  - SLICE statement (SURVEYPHREG), 494
  - SLICE statement (SURVEYREG), 494
- ODS table names
  - ESTIMATE statement (LOGISTIC), 477
  - ESTIMATE statement (ORTHOREG), 477
  - ESTIMATE statement (PHREG), 477
  - ESTIMATE statement (PLM), 477
  - ESTIMATE statement (SURVEYLOGISTIC), 477
  - ESTIMATE statement (SURVEYPHREG), 477
  - ESTIMATE statement (SURVEYREG), 477
  - LSMEANS statement (GENMOD), 494
  - LSMEANS statement (LOGISTIC), 494



- LSMEANS statement (ORTHOREG), 494
- LSMEANS statement (PHREG), 494
- LSMEANS statement (PLM), 494
- LSMEANS statement (SURVEYLOGISTIC), 494
- LSMEANS statement (SURVEYPHREG), 494
- LSMEANS statement (SURVEYREG), 494
- LSMESTIMATE statement (GENMOD), 507
- LSMESTIMATE statement (LOGISTIC), 507
- LSMESTIMATE statement (MIXED), 507
- LSMESTIMATE statement (ORTHOREG), 507
- LSMESTIMATE statement (PHREG), 507
- LSMESTIMATE statement (PLM), 507
- LSMESTIMATE statement (SURVEYLOGISTIC), 507
- LSMESTIMATE statement (SURVEYPHREG), 507
- LSMESTIMATE statement (SURVEYREG), 507
- SLICE statement (GENMOD), 529
- SLICE statement (GLIMMIX), 529
- SLICE statement (LOGISTIC), 529
- SLICE statement (MIXED), 529
- SLICE statement (ORTHOREG), 529
- SLICE statement (PHREG), 529
- SLICE statement (PLM), 529
- SLICE statement (SURVEYLOGISTIC), 529
- SLICE statement (SURVEYPHREG), 529
- SLICE statement (SURVEYREG), 529
- TEST statement (ORTHOREG), 532
- TEST statement (PLM), 532
- TEST statement (SURVEYPHREG), 532
- TEST statement (SURVEYREG), 532
- ORTHOREG procedure, EFFECT statement
  - BASIS option (spline), 429
  - collection effect, 420
  - DATABOUNDARY option (spline), 429
  - DEGREE option (polynomial), 425
  - DEGREE option (spline), 429
  - DESIGNROLE option (lag), 422
  - DETAILS option (lag), 422
  - DETAILS option (multimember), 424
  - DETAILS option (polynomial), 425
  - DETAILS option (spline), 429
  - KNOTMAX option (spline), 429
  - KNOTMETHOD option (spline), 430
  - KNOTMIN option (spline), 431
  - LABELSTYLE option (polynomial), 425
  - lag effect, 420
  - MDEGREE option (polynomial), 426
  - multimember effect, 423
  - NLAG option (lag), 422
  - NOEFFECT option (multimember), 424
  - NOSEPARATE option (polynomial), 426
  - PERIOD option (lag), 422
  - polynomial effect, 425
  - SEPARATE option (spline), 431
  - spline effect, 428
  - SPLIT option (spline), 431
  - STANDARDIZE option (polynomial), 426
  - WITHIN option (lag), 422
- ORTHOREG procedure, EFFECTPLOT statement
  - ALPHA= option, 438
  - AT option, 438
  - ATLEN= option, 439
  - ATORDER= option, 439
  - CLI option, 440
  - CLM option, 440
  - EXTEND= option, 440
  - GRIDSIZE= option, 440
  - ILINK option, 440
  - INDIVIDUAL option, 440
  - LIMITS option, 440
  - LINK option, 440
  - MOFF option, 441
  - NCOLS= option, 441
  - NOCLI option, 441
  - NOCLM option, 441
  - NOLIMITS option, 441
  - NOOBS option, 441
  - NROWS= option, 441
  - OBS option, 441
  - PLOTBY= option, 444
  - PLOTBYLEN= option, 445
  - POLYBAR option, 445
  - PREDLABEL= option, 445
  - SHOWCLEGEND option, 445
  - SLICEBY= option, 445
  - SMOOTH option, 445
  - UNPACK option, 445
  - X= option, 445
  - Y= option, 446
  - YRANGE= option, 446
- ORTHOREG procedure, ESTIMATE statement
  - ADJDFE= option, 464
  - ADJUST= option, 465
  - ALPHA= option, 465
  - CHISQ option, 466
  - CL option, 466
  - CORR option, 466
  - COV option, 466
  - DF= option, 467
  - DIVISOR= option, 467
  - E option, 467
  - JOINT option, 468

- LOWER option, [469](#)
- NOFILL option, [469](#)
- ODS table names, [477](#)
- SEED= option, [471](#)
- SINGULAR= option, [471](#)
- STEPDOWN option, [471](#)
- TESTVALUE option, [472](#)
- UPPER option, [473](#)
- ORTHOREG procedure, LSMEANS statement
  - ADJDFE= option, [481](#)
  - ADJUST= option, [482](#)
  - ALPHA= option, [484](#)
  - AT= option, [484](#)
  - BYLEVEL option, [485](#)
  - CL option, [485](#)
  - CORR option, [485](#)
  - COV option, [485](#)
  - DF= option, [485](#)
  - DIFF option, [485](#)
  - E option, [486](#)
  - LINES option, [487](#)
  - MEANS or NOMEANS option, [487](#)
  - OBSMARGINS= option, [488](#)
  - ODS graph names, [494](#)
  - ODS table names, [494](#)
  - PDIFF option, [488](#)
  - PLOTS= option, [488](#)
  - SEED= option, [492](#)
  - SINGULAR= option, [492](#)
  - STEPDOWN option, [493](#)
- ORTHOREG procedure, LSMESIMATE statement
  - ADJDFE= option, [498](#)
  - ADJUST= option, [499](#)
  - ALPHA= option, [499](#)
  - AT= option, [499](#)
  - BYLEVEL option, [499](#)
  - CHISQ option, [500](#)
  - CL option, [500](#)
  - CORR option, [500](#)
  - COV option, [500](#)
  - DF= option, [500](#)
  - DIVISOR= option, [501](#)
  - E option, [501](#)
  - ELSM option, [501](#)
  - JOINT option, [502](#)
  - LOWER option, [503](#)
  - OBSMARGINS= option, [503](#)
  - ODS table names, [507](#)
  - PLOTS= option, [504](#)
  - SEED= option, [505](#)
  - SINGULAR= option, [505](#)
  - STEPDOWN option, [505](#)
  - TESTVALUE= option, [506](#)
  - UPPER option, [507](#)
- ORTHOREG procedure, SLICE statement
  - ADJDFE= option, [481](#)
  - ADJUST= option, [482](#)
  - ALPHA= option, [484](#)
  - AT= option, [484](#)
  - BYLEVEL option, [485](#)
  - CL option, [485](#)
  - CORR option, [485](#)
  - COV option, [485](#)
  - DF= option, [485](#)
  - DIFF option, [485](#)
  - E option, [486](#)
  - LINES option, [487](#)
  - MEANS or NOMEANS option, [487](#)
  - NOF option, [528](#)
  - OBSMARGINS= option, [488](#)
  - ODS table names, [529](#)
  - PDIFF option, [488](#)
  - PLOTS= option, [488](#)
  - SEED= option, [492](#)
  - SIMPLE= option, [528](#)
  - SINGULAR= option, [492](#)
  - SLICEBY= option, [528](#)
  - STEPDOWN option, [493](#)
- ORTHOREG procedure, TEST statement
  - CHISQ option, [531](#)
  - DDF= option, [531](#)
  - E option, [531](#)
  - E1 option, [532](#)
  - E2 option, [532](#)
  - E3 option, [532](#)
  - HTYPE= option, [532](#)
  - INTERCEPT option, [532](#)
  - ODS table names, [532](#)
- PDIFF option
  - LSMEANS statement (GENMOD), [488](#)
  - LSMEANS statement (LOGISTIC), [488](#)
  - LSMEANS statement (ORTHOREG), [488](#)
  - LSMEANS statement (PHREG), [488](#)
  - LSMEANS statement (PLM), [488](#)
  - LSMEANS statement (SURVEYLOGISTIC), [488](#)
  - LSMEANS statement (SURVEYPHREG), [488](#)
  - LSMEANS statement (SURVEYREG), [488](#)
  - SLICE statement (GENMOD), [488](#)
  - SLICE statement (GLIMMIX), [488](#)
  - SLICE statement (LOGISTIC), [488](#)
  - SLICE statement (MIXED), [488](#)
  - SLICE statement (ORTHOREG), [488](#)
  - SLICE statement (PHREG), [488](#)
  - SLICE statement (PLM), [488](#)

- SLICE statement (SURVEYLOGISTIC), 488
- SLICE statement (SURVEYPHREG), 488
- SLICE statement (SURVEYREG), 488
- PERIOD option
  - EFFECT statement, lag (GLIMMIX), 422
  - EFFECT statement, lag (GLMSELECT), 422
  - EFFECT statement, lag (HPMIXED), 422
  - EFFECT statement, lag (LOGISTIC), 422
  - EFFECT statement, lag (ORTHOREG), 422
  - EFFECT statement, lag (PHREG), 422
  - EFFECT statement, lag (PLS), 422
  - EFFECT statement, lag (ROBUSTREG), 422
  - EFFECT statement, lag
    - (SURVEYLOGISTIC), 422
  - EFFECT statement, lag (SURVEYREG), 422
- PHREG procedure, EFFECT statement
  - BASIS option (spline), 429
  - collection effect, 420
  - DATABOUNDARY option (spline), 429
  - DEGREE option (polynomial), 425
  - DEGREE option (spline), 429
  - DESIGNROLE option (lag), 422
  - DETAILS option (lag), 422
  - DETAILS option (multimember), 424
  - DETAILS option (polynomial), 425
  - DETAILS option (spline), 429
  - KNOTMAX option (spline), 429
  - KNOTMETHOD option (spline), 430
  - KNOTMIN option (spline), 431
  - LABELSTYLE option (polynomial), 425
  - lag effect, 420
  - MDEGREE option (polynomial), 426
  - multimember effect, 423
  - NLAG option (lag), 422
  - NOEFFECT option (multimember), 424
  - NOSEPARATE option (polynomial), 426
  - PERIOD option (lag), 422
  - polynomial effect, 425
  - SEPARATE option (spline), 431
  - spline effect, 428
  - SPLIT option (spline), 431
  - STANDARDIZE option (polynomial), 426
  - WITHIN option (lag), 422
- PHREG procedure, ESTIMATE statement
  - ADJUST= option, 465
  - ALPHA= option, 465
  - CL option, 466
  - CORR option, 466
  - COV option, 466
  - DIVISOR= option, 467
  - E option, 467
  - EXP option, 467
  - JOINT option, 468
  - LOWER option, 469
  - NOFILL option, 469
  - ODS graph names, 478
  - ODS table names, 477
  - PLOTS= option, 470
  - SEED= option, 471
  - SINGULAR= option, 471
  - STEPDOWN option, 471
  - TESTVALUE option, 472
  - UPPER option, 473
- PHREG procedure, LSMEANS statement
  - ADJUST= option, 482
  - ALPHA= option, 484
  - AT= option, 484
  - BYLEVEL option, 485
  - CL option, 485
  - CORR option, 485
  - COV option, 485
  - DIFF option, 485
  - E option, 486
  - EXP option, 486
  - LINES option, 487
  - MEANS or NOMEANS option, 487
  - OBSMARGINS= option, 488
  - ODS graph names, 494
  - ODS table names, 494
  - PDIFF option, 488
  - PLOTS= option, 488
  - SEED= option, 492
  - SINGULAR= option, 492
  - STEPDOWN option, 493
- PHREG procedure, LSMESTIMATE statement
  - ADJUST= option, 499
  - ALPHA= option, 499
  - AT= option, 499
  - BYLEVEL option, 499
  - CL option, 500
  - CORR option, 500
  - COV option, 500
  - DIVISOR= option, 501
  - E option, 501
  - ELSM option, 501
  - EXP option, 501
  - JOINT option, 502
  - LOWER option, 503
  - OBSMARGINS= option, 503
  - ODS graph names, 507
  - ODS table names, 507
  - PLOTS= option, 504
  - SEED= option, 505
  - SINGULAR= option, 505
  - STEPDOWN option, 505
  - TESTVALUE= option, 506
  - UPPER option, 507
- PHREG procedure, SLICE statement



- ADJUST= option, 482
- ALPHA= option, 484
- AT= option, 484
- BYLEVEL option, 485
- CL option, 485
- CORR option, 485
- COV option, 485
- DIFF option, 485
- E option, 486
- EXP option, 486
- LINES option, 487
- MEANS or NOMEANS option, 487
- NOF option, 528
- OBSMARGINS= option, 488
- ODS table names, 529
- PDIFF option, 488
- PLOTS= option, 488
- SEED= option, 492
- SIMPLE= option, 528
- SINGULAR= option, 492
- SLICEBY= option, 528
- STEPDOWN option, 493
- PLM procedure, EFFECTPLOT statement
  - ALPHA= option, 438
  - AT option, 438
  - ATLEN= option, 439
  - ATORDER= option, 439
  - CLI option, 440
  - CLM option, 440
  - EXTEND= option, 440
  - GRIDSIZE= option, 440
  - ILINK option, 440
  - INDIVIDUAL option, 440
  - LIMITS option, 440
  - LINK option, 440
  - MOFF option, 441
  - NCOLS= option, 441
  - NOCLI option, 441
  - NOCLM option, 441
  - NOLIMITS option, 441
  - NOOBS option, 441
  - NROWS= option, 441
  - OBS option, 441
  - PLOTBY= option, 444
  - PLOTBYLEN= option, 445
  - POLYBAR option, 445
  - PREDLABEL= option, 445
  - SHOWCLEGEND option, 445
  - SLICEBY= option, 445
  - SMOOTH option, 445
  - UNPACK option, 445
  - X= option, 445
  - Y= option, 446
  - YRANGE= option, 446

- PLM procedure, ESTIMATE statement
  - ADJDFE= option, 464
  - ADJUST= option, 465
  - ALPHA= option, 465
  - CATEGORY= option, 465
  - CHISQ option, 466
  - CL option, 466
  - CORR option, 466
  - COV option, 466
  - DF= option, 467
  - DIVISOR= option, 467
  - E option, 467
  - EXP option, 467
  - ILINK option, 467
  - JOINT option, 468
  - LOWER option, 469
  - NOFILL option, 469
  - ODS graph names, 478
  - ODS table names, 477
  - PLOTS= option, 470
  - SEED= option, 471
  - SINGULAR= option, 471
  - STEPDOWN option, 471
  - TESTVALUE option, 472
  - UPPER option, 473

- PLM procedure, LSMEANS statement
  - ADJDFE= option, 481
  - ADJUST= option, 482
  - ALPHA= option, 484
  - AT= option, 484
  - BYLEVEL option, 485
  - CL option, 485
  - CORR option, 485
  - COV option, 485
  - DF= option, 485
  - DIFF option, 485
  - E option, 486
  - EXP option, 486
  - ILINK option, 487
  - LINES option, 487
  - MEANS or NOMEANS option, 487
  - OBSMARGINS= option, 488
  - ODDSRATIO option, 487
  - ODS graph names, 494
  - ODS table names, 494
  - PDIFF option, 488
  - PLOTS= option, 488
  - SEED= option, 492
  - SINGULAR= option, 492
  - STEPDOWN option, 493

- PLM procedure, LSMESTIMATE statement
  - ADJDFE= option, 498
  - ADJUST= option, 499
  - ALPHA= option, 499

- AT= option, 499
- BYLEVEL option, 499
- CATEGORY= option, 500
- CHISQ option, 500
- CL option, 500
- CORR option, 500
- COV option, 500
- DF= option, 500
- DIVISOR= option, 501
- E option, 501
- ELSM option, 501
- EXP option, 501
- ILINK option, 501
- JOINT option, 502
- LOWER option, 503
- OBSMARGINS= option, 503
- ODS graph names, 507
- ODS table names, 507
- PLOTS= option, 504
- SEED= option, 505
- SINGULAR= option, 505
- STEPDOWN option, 505
- TESTVALUE= option, 506
- UPPER option, 507
- PLM procedure, SLICE statement
  - ADJDFE= option, 481
  - ADJUST= option, 482
  - ALPHA= option, 484
  - AT= option, 484
  - BYLEVEL option, 485
  - CL option, 485
  - CORR option, 485
  - COV option, 485
  - DF= option, 485
  - DIFF option, 485
  - E option, 486
  - EXP option, 486
  - ILINK option, 487
  - LINES option, 487
  - MEANS or NOMEANS option, 487
  - NOF option, 528
  - OBSMARGINS= option, 488
  - ODDSRATIO option, 487
  - ODS table names, 529
  - PDIFF option, 488
  - PLOTS= option, 488
  - SEED= option, 492
  - SIMPLE= option, 528
  - SINGULAR= option, 492
  - SLICEBY= option, 528
  - STEPDOWN option, 493
- PLM procedure, TEST statement
  - CHISQ option, 531
  - DDF= option, 531
  - E option, 531
  - E1 option, 532
  - E2 option, 532
  - E3 option, 532
  - HTYPE= option, 532
  - INTERCEPT option, 532
  - ODS table names, 532
- PLOTBY= option
  - EFFECTPLOT statement, 444
- PLOTBYLEN= option
  - EFFECTPLOT statement, 445
- PLOTS= option
  - ESTIMATE statement (PHREG), 470
  - ESTIMATE statement (PLM), 470
  - LSMEANS statement (GENMOD), 488
  - LSMEANS statement (LOGISTIC), 488
  - LSMEANS statement (ORTHOREG), 488
  - LSMEANS statement (PHREG), 488
  - LSMEANS statement (PLM), 488
  - LSMEANS statement (SURVEYLOGISTIC), 488
  - LSMEANS statement (SURVEYPHREG), 488
  - LSMEANS statement (SURVEYREG), 488
  - LSMESTIMATE statement (GENMOD), 504
  - LSMESTIMATE statement (LOGISTIC), 504
  - LSMESTIMATE statement (MIXED), 504
  - LSMESTIMATE statement (ORTHOREG), 504
  - LSMESTIMATE statement (PHREG), 504
  - LSMESTIMATE statement (PLM), 504
  - LSMESTIMATE statement (SURVEYLOGISTIC), 504
  - LSMESTIMATE statement (SURVEYPHREG), 504
  - LSMESTIMATE statement (SURVEYREG), 504
  - SLICE statement (GENMOD), 488
  - SLICE statement (GLIMMIX), 488
  - SLICE statement (LOGISTIC), 488
  - SLICE statement (MIXED), 488
  - SLICE statement (ORTHOREG), 488
  - SLICE statement (PHREG), 488
  - SLICE statement (PLM), 488
  - SLICE statement (SURVEYLOGISTIC), 488
  - SLICE statement (SURVEYPHREG), 488
  - SLICE statement (SURVEYREG), 488
- PLS procedure, EFFECT statement
  - BASIS option (spline), 429
  - collection effect, 420
  - DATABOUNDARY option (spline), 429
  - DEGREE option (polynomial), 425
  - DEGREE option (spline), 429

DESIGNROLE option (lag), 422  
 DETAILS option (lag), 422  
 DETAILS option (multimember), 424  
 DETAILS option (polynomial), 425  
 DETAILS option (spline), 429  
 KNOTMAX option (spline), 429  
 KNOTMETHOD option (spline), 430  
 KNOTMIN option (spline), 431  
 LABELSTYLE option (polynomial), 425  
 lag effect, 420  
 MDEGREE option (polynomial), 426  
 multimember effect, 423  
 NLAG option (lag), 422  
 NOEFFECT option (multimember), 424  
 NOSEPARATE option (polynomial), 426  
 PERIOD option (lag), 422  
 polynomial effect, 425  
 SEPARATE option (spline), 431  
 spline effect, 428  
 SPLIT option (spline), 431  
 STANDARDIZE option (polynomial), 426  
 WITHIN option (lag), 422

POLYBAR option  
   EFFECTPLOT statement, 445

PREDLABEL= option  
   EFFECTPLOT statement, 445

QUANTREG procedure, EFFECT statement  
   BASIS option (spline), 429  
   DATABOUNDARY option (spline), 429  
   DEGREE option (spline), 429  
   DETAILS option (spline), 429  
   KNOTMAX option (spline), 429  
   KNOTMETHOD option (spline), 430  
   KNOTMIN option (spline), 431  
   SEPARATE option (spline), 431  
   spline effect, 428  
   SPLIT option (spline), 431

REST option  
   NLOPTIONS statement (GLIMMIX), 518  
   NLOPTIONS statement (HPMIXED), 518  
   NLOPTIONS statement (TCALIS), 518

RESTART option  
   NLOPTIONS statement (GLIMMIX), 518  
   NLOPTIONS statement (HPMIXED), 518  
   NLOPTIONS statement (TCALIS), 518

ROBUSTREG procedure, EFFECT statement  
   BASIS option (spline), 429  
   collection effect, 420  
   DATABOUNDARY option (spline), 429  
   DEGREE option (polynomial), 425  
   DEGREE option (spline), 429  
   DESIGNROLE option (lag), 422  
   DETAILS option (lag), 422  
   DETAILS option (multimember), 424  
   DETAILS option (polynomial), 425  
   DETAILS option (spline), 429  
   KNOTMAX option (spline), 429  
   KNOTMETHOD option (spline), 430  
   KNOTMIN option (spline), 431  
   LABELSTYLE option (polynomial), 425  
   lag effect, 420  
   MDEGREE option (polynomial), 426  
   multimember effect, 423  
   NLAG option (lag), 422  
   NOEFFECT option (multimember), 424  
   NOSEPARATE option (polynomial), 426  
   PERIOD option (lag), 422  
   polynomial effect, 425  
   SEPARATE option (spline), 431  
   spline effect, 428  
   SPLIT option (spline), 431  
   STANDARDIZE option (polynomial), 426  
   WITHIN option (lag), 422

SEED= option  
   ESTIMATE statement (LOGISTIC), 471  
   ESTIMATE statement (ORTHOREG), 471  
   ESTIMATE statement (PHREG), 471  
   ESTIMATE statement (PLM), 471  
   ESTIMATE statement  
     (SURVEYLOGISTIC), 471  
   ESTIMATE statement (SURVEYPHREG),  
     471  
   ESTIMATE statement (SURVEYREG), 471  
   LSMEANS statement (GENMOD), 492  
   LSMEANS statement (LOGISTIC), 492  
   LSMEANS statement (ORTHOREG), 492  
   LSMEANS statement (PHREG), 492  
   LSMEANS statement (PLM), 492  
   LSMEANS statement (SURVEYLOGISTIC),  
     492  
   LSMEANS statement (SURVEYPHREG),  
     492  
   LSMEANS statement (SURVEYREG), 492  
   LSMESTIMATE statement (GENMOD), 505  
   LSMESTIMATE statement (LOGISTIC),  
     505  
   LSMESTIMATE statement (MIXED), 505  
   LSMESTIMATE statement (ORTHOREG),  
     505  
   LSMESTIMATE statement (PHREG), 505  
   LSMESTIMATE statement (PLM), 505  
   LSMESTIMATE statement  
     (SURVEYLOGISTIC), 505  
   LSMESTIMATE statement  
     (SURVEYPHREG), 505

- LSMESTIMATE statement (SURVEYREG), 505
- SLICE statement (GENMOD), 492
- SLICE statement (GLIMMIX), 492
- SLICE statement (LOGISTIC), 492
- SLICE statement (MIXED), 492
- SLICE statement (ORTHOREG), 492
- SLICE statement (PHREG), 492
- SLICE statement (PLM), 492
- SLICE statement (SURVEYLOGISTIC), 492
- SLICE statement (SURVEYPHREG), 492
- SLICE statement (SURVEYREG), 492
- SEPARATE option
  - EFFECT statement, spline (GLIMMIX), 431
  - EFFECT statement, spline (GLMSELECT), 431
  - EFFECT statement, spline (HPMIXED), 431
  - EFFECT statement, spline (LOGISTIC), 431
  - EFFECT statement, spline (ORTHOREG), 431
  - EFFECT statement, spline (PHREG), 431
  - EFFECT statement, spline (PLS), 431
  - EFFECT statement, spline (QUANTREG), 431
  - EFFECT statement, spline (ROBUSTREG), 431
  - EFFECT statement, spline (SURVEYLOGISTIC), 431
  - EFFECT statement, spline (SURVEYREG), 431
- SHOWCLEGEND option
  - EFFECTPLOT statement, 445
- SIMPLE= option
  - SLICE statement (GENMOD), 528
  - SLICE statement (GLIMMIX), 528
  - SLICE statement (LOGISTIC), 528
  - SLICE statement (MIXED), 528
  - SLICE statement (ORTHOREG), 528
  - SLICE statement (PHREG), 528
  - SLICE statement (PLM), 528
  - SLICE statement (SURVEYLOGISTIC), 528
  - SLICE statement (SURVEYPHREG), 528
  - SLICE statement (SURVEYREG), 528
- SINGULAR= option
  - ESTIMATE statement (LOGISTIC), 471
  - ESTIMATE statement (ORTHOREG), 471
  - ESTIMATE statement (PHREG), 471
  - ESTIMATE statement (PLM), 471
  - ESTIMATE statement (SURVEYLOGISTIC), 471
  - ESTIMATE statement (SURVEYPHREG), 471
  - ESTIMATE statement (SURVEYREG), 471
  - LSMEANS statement (GENMOD), 492
  - LSMEANS statement (LOGISTIC), 492
  - LSMEANS statement (ORTHOREG), 492
  - LSMEANS statement (PHREG), 492
  - LSMEANS statement (PLM), 492
  - LSMEANS statement (SURVEYLOGISTIC), 492
  - LSMEANS statement (SURVEYPHREG), 492
  - LSMEANS statement (SURVEYREG), 492
  - LSMESTIMATE statement (GENMOD), 505
  - LSMESTIMATE statement (LOGISTIC), 505
  - LSMESTIMATE statement (MIXED), 505
  - LSMESTIMATE statement (ORTHOREG), 505
  - LSMESTIMATE statement (PHREG), 505
  - LSMESTIMATE statement (PLM), 505
  - LSMESTIMATE statement (SURVEYLOGISTIC), 505
  - LSMESTIMATE statement (SURVEYPHREG), 505
  - LSMESTIMATE statement (SURVEYREG), 505
  - NLOPTIONS statement (GLIMMIX), 519
  - NLOPTIONS statement (HPMIXED), 519
  - NLOPTIONS statement (TCALIS), 519
  - SLICE statement (GENMOD), 492
  - SLICE statement (GLIMMIX), 492
  - SLICE statement (LOGISTIC), 492
  - SLICE statement (MIXED), 492
  - SLICE statement (ORTHOREG), 492
  - SLICE statement (PHREG), 492
  - SLICE statement (PLM), 492
  - SLICE statement (SURVEYLOGISTIC), 492
  - SLICE statement (SURVEYPHREG), 492
  - SLICE statement (SURVEYREG), 492
  - SLICE statement
    - GENMOD procedure, 526
    - GLIMMIX procedure, 526
    - LOGISTIC procedure, 526
    - MIXED procedure, 526
    - ORTHOREG procedure, 526
    - PHREG procedure, 526
    - PLM procedure, 526
    - SURVEYLOGISTIC procedure, 526
    - SURVEYPHREG procedure, 526
    - SURVEYREG procedure, 526
  - SLICEBY= option
    - EFFECTPLOT statement, 445
    - SLICE statement (GENMOD), 528
    - SLICE statement (GLIMMIX), 528
    - SLICE statement (LOGISTIC), 528
    - SLICE statement (MIXED), 528
    - SLICE statement (ORTHOREG), 528

- SLICE statement (PHREG), 528
- SLICE statement (PLM), 528
- SLICE statement (SURVEYLOGISTIC), 528
- SLICE statement (SURVEYPHREG), 528
- SLICE statement (SURVEYREG), 528
- SMOOTH option
  - EFFECTPLOT statement, 445
- SOCKET option
  - NLOPTIONS statement (GLIMMIX), 519
  - NLOPTIONS statement (HPMIXED), 519
  - NLOPTIONS statement (TCALIS), 519
- SPLIT option
  - EFFECT statement, spline (GLMSELECT), 431
  - EFFECT statement, spline (HPMIXED), 431
  - EFFECT statement, spline (LOGISTIC), 431
  - EFFECT statement, spline (ORTHOREG), 431
  - EFFECT statement, spline (PHREG), 431
  - EFFECT statement, spline (PLS), 431
  - EFFECT statement, spline (QUANTREG), 431
  - EFFECT statement, spline (ROBUSTREG), 431
  - EFFECT statement, spline (SURVEYLOGISTIC), 431
  - EFFECT statement, spline (SURVEYREG), 431
- STANDARDIZE option
  - EFFECT statement, polynomial (GLIMMIX), 426
  - EFFECT statement, polynomial (GLMSELECT), 426
  - EFFECT statement, polynomial (HPMIXED), 426
  - EFFECT statement, polynomial (LOGISTIC), 426
  - EFFECT statement, polynomial (ORTHOREG), 426
  - EFFECT statement, polynomial (PHREG), 426
  - EFFECT statement, polynomial (PLS), 426
  - EFFECT statement, polynomial (ROBUSTREG), 426
  - EFFECT statement, polynomial (SURVEYLOGISTIC), 426
  - EFFECT statement, polynomial (SURVEYREG), 426
- STEPDOWN option
  - ESTIMATE statement (LOGISTIC), 471
  - ESTIMATE statement (ORTHOREG), 471
  - ESTIMATE statement (PHREG), 471
  - ESTIMATE statement (PLM), 471
- ESTIMATE statement (SURVEYLOGISTIC), 471
- ESTIMATE statement (SURVEYPHREG), 471
- ESTIMATE statement (SURVEYREG), 471
- LSMEANS statement (GENMOD), 493
- LSMEANS statement (LOGISTIC), 493
- LSMEANS statement (ORTHOREG), 493
- LSMEANS statement (PHREG), 493
- LSMEANS statement (PLM), 493
- LSMEANS statement (SURVEYLOGISTIC), 493
- LSMEANS statement (SURVEYPHREG), 493
- LSMEANS statement (SURVEYREG), 493
- LSMESTIMATE statement (GENMOD), 505
- LSMESTIMATE statement (LOGISTIC), 505
- LSMESTIMATE statement (MIXED), 505
- LSMESTIMATE statement (ORTHOREG), 505
- LSMESTIMATE statement (PHREG), 505
- LSMESTIMATE statement (PLM), 505
- LSMESTIMATE statement (SURVEYLOGISTIC), 505
- LSMESTIMATE statement (SURVEYPHREG), 505
- LSMESTIMATE statement (SURVEYREG), 505
- SLICE statement (GENMOD), 493
- SLICE statement (GLIMMIX), 493
- SLICE statement (LOGISTIC), 493
- SLICE statement (MIXED), 493
- SLICE statement (ORTHOREG), 493
- SLICE statement (PHREG), 493
- SLICE statement (PLM), 493
- SLICE statement (SURVEYLOGISTIC), 493
- SLICE statement (SURVEYPHREG), 493
- SLICE statement (SURVEYREG), 493
- STORE statement
  - GENMOD procedure, 529
  - GLIMMIX procedure, 529
  - GLM procedure, 529
  - LOGISTIC procedure, 529
  - MIXED procedure, 529
  - ORTHOREG procedure, 529
  - PHREG procedure, 529
  - SURVEYLOGISTIC procedure, 529
  - SURVEYPHREG procedure, 529
  - SURVEYREG procedure, 529
- SUREYREG procedure, EFFECT statement
  - DESIGNROLE option (lag), 422
- SURVEYLOGISTIC procedure, EFFECT statement

- BASIS option (spline), 429
- collection effect, 420
- DATABOUNDARY option (spline), 429
- DEGREE option (polynomial), 425
- DEGREE option (spline), 429
- DESIGNROLE option (lag), 422
- DETAILS option (lag), 422
- DETAILS option (multimember), 424
- DETAILS option (polynomial), 425
- DETAILS option (spline), 429
- KNOTMAX option (spline), 429
- KNOTMETHOD option (spline), 430
- KNOTMIN option (spline), 431
- LABELSTYLE option (polynomial), 425
- lag effect, 420
- MDEGREE option (polynomial), 426
- multimember effect, 423
- NLAG option (lag), 422
- NOEFFECT option (multimember), 424
- NOSEPARATE option (polynomial), 426
- PERIOD option (lag), 422
- polynomial effect, 425
- SEPARATE option (spline), 431
- spline effect, 428
- SPLIT option (spline), 431
- STANDARDIZE option (polynomial), 426
- WITHIN option (lag), 422
- SURVEYLOGISTIC procedure, ESTIMATE
  - statement
    - ADJUST= option, 465
    - ALPHA= option, 465
    - CATEGORY= option, 465
    - CL option, 466
    - CORR option, 466
    - COV option, 466
    - DIVISOR= option, 467
    - E option, 467
    - EXP option, 467
    - ILINK option, 467
    - JOINT option, 468
    - LOWER option, 469
    - NOFILL option, 469
    - ODS table names, 477
    - SEED= option, 471
    - SINGULAR= option, 471
    - STEPDOWN option, 471
    - TESTVALUE option, 472
    - UPPER option, 473
- SURVEYLOGISTIC procedure, LSMEANS
  - statement
    - ADJUST= option, 482
    - ALPHA= option, 484
    - AT= option, 484
    - BYLEVEL option, 485
    - CL option, 485
    - CORR option, 485
    - COV option, 485
    - DIFF option, 485
    - E option, 486
    - EXP option, 486
    - CL option, 485
    - CORR option, 485
    - COV option, 485
    - DIFF option, 485
    - E option, 486
    - EXP option, 486
- CL option, 485
- CORR option, 485
- COV option, 485
- DIFF option, 485
- E option, 486
- EXP option, 486
- ILINK option, 487
- LINES option, 487
- MEANS or NOMEANS option, 487
- OBSMARGINS= option, 488
- ODDSRATIO option, 487
- ODS graph names, 494
- ODS table names, 494
- PDIFF option, 488
- PLOTS= option, 488
- SEED= option, 492
- SINGULAR= option, 492
- STEPDOWN option, 493
- SURVEYLOGISTIC procedure, LSMESTIMATE
  - statement
    - ADJUST= option, 499
    - ALPHA= option, 499
    - AT= option, 499
    - BYLEVEL option, 499
    - CATEGORY= option, 500
    - CL option, 500
    - CORR option, 500
    - COV option, 500
    - DIVISOR= option, 501
    - E option, 501
    - ELSM option, 501
    - EXP option, 501
    - ILINK option, 501
    - JOINT option, 502
    - LOWER option, 503
    - OBSMARGINS= option, 503
    - ODS table names, 507
    - PLOTS= option, 504
    - SEED= option, 505
    - SINGULAR= option, 505
    - STEPDOWN option, 505
    - TESTVALUE= option, 506
    - UPPER option, 507
- SURVEYLOGISTIC procedure, SLICE statement
  - ADJUST= option, 482
  - ALPHA= option, 484
  - AT= option, 484
  - BYLEVEL option, 485
  - CL option, 485
  - CORR option, 485
  - COV option, 485
  - DIFF option, 485
  - E option, 486
  - EXP option, 486



- ILINK option, 487
- LINES option, 487
- MEANS or NOMEANS option, 487
- NOF option, 528
- OBSMARGINS= option, 488
- ODDSRATIO option, 487
- ODS table names, 529
- PDIFF option, 488
- PLOTS= option, 488
- SEED= option, 492
- SIMPLE= option, 528
- SINGULAR= option, 492
- SLICEBY= option, 528
- STEPPDOWN option, 493
- SURVEYPHREG procedure, ESTIMATE statement
  - ADJDFE= option, 464
  - ADJUST= option, 465
  - ALPHA= option, 465
  - CHISQ option, 466
  - CL option, 466
  - CORR option, 466
  - COV option, 466
  - DF= option, 467
  - DIVISOR= option, 467
  - E option, 467
  - JOINT option, 468
  - LOWER option, 469
  - NOFILL option, 469
  - ODS table names, 477
  - SEED= option, 471
  - SINGULAR= option, 471
  - STEPPDOWN option, 471
  - TESTVALUE option, 472
  - UPPER option, 473
- SURVEYPHREG procedure, LSMEANS statement
  - ADJDFE= option, 481
  - ADJUST= option, 482
  - ALPHA= option, 484
  - AT= option, 484
  - BYLEVEL option, 485
  - CL option, 485
  - CORR option, 485
  - COV option, 485
  - DF= option, 485
  - DIFF option, 485
  - E option, 486
  - LINES option, 487
  - MEANS or NOMEANS option, 487
  - NOF option, 528
  - OBSMARGINS= option, 488
  - ODS graph names, 494
  - ODS table names, 494
  - PDIFF option, 488
- PLOTS= option, 488
- SEED= option, 492
- SINGULAR= option, 492
- STEPPDOWN option, 493
- SURVEYPHREG procedure, LSMESIMATE statement
  - ADJDFE= option, 498
  - ADJUST= option, 499
  - ALPHA= option, 499
  - AT= option, 499
  - BYLEVEL option, 499
  - CHISQ option, 500
  - CL option, 500
  - CORR option, 500
  - COV option, 500
  - DF= option, 500
  - DIVISOR= option, 501
  - E option, 501
  - ELSM option, 501
  - JOINT option, 502
  - LOWER option, 503
  - OBSMARGINS= option, 503
  - ODS table names, 507
  - PLOTS= option, 504
  - SEED= option, 505
  - SINGULAR= option, 505
  - STEPPDOWN option, 505
  - TESTVALUE= option, 506
  - UPPER option, 507
- SURVEYPHREG procedure, SLICE statement
  - ADJUST= option, 482
  - ALPHA= option, 484
  - AT= option, 484
  - BYLEVEL option, 485
  - CL option, 485
  - CORR option, 485
  - COV option, 485
  - DF= option, 485
  - DIFF option, 485
  - E option, 486
  - LINES option, 487
  - MEANS or NOMEANS option, 487
  - NOF option, 528
  - OBSMARGINS= option, 488
  - ODS table names, 529
  - PDIFF option, 488
  - PLOTS= option, 488
  - SEED= option, 492
  - SIMPLE= option, 528
  - SINGULAR= option, 492
  - SLICEBY= option, 528
  - STEPPDOWN option, 493
- SURVEYPHREG procedure, TEST statement
  - CHISQ option, 531

- DDF= option, 531
- E option, 531
- E1 option, 532
- E2 option, 532
- E3 option, 532
- HTYPE= option, 532
- INTERCEPT option, 532
- ODS table names, 532
- SURVEYREG procedure, EFFECT statement
  - BASIS option (spline), 429
  - collection effect, 420
  - DATABOUNDARY option (spline), 429
  - DEGREE option (polynomial), 425
  - DEGREE option (spline), 429
  - DETAILS option (lag), 422
  - DETAILS option (multimember), 424
  - DETAILS option (polynomial), 425
  - DETAILS option (spline), 429
  - KNOTMAX option (spline), 429
  - KNOTMETHOD option (spline), 430
  - KNOTMIN option (spline), 431
  - LABELSTYLE option (polynomial), 425
  - lag effect, 420
  - MDEGREE option (polynomial), 426
  - multimember effect, 423
  - NLAG option (lag), 422
  - NOEFFECT option (multimember), 424
  - NOSEPARATE option (polynomial), 426
  - PERIOD option (lag), 422
  - polynomial effect, 425
  - SEPARATE option (spline), 431
  - spline effect, 428
  - SPLIT option (spline), 431
  - STANDARDIZE option (polynomial), 426
  - WITHIN option (lag), 422
- SURVEYREG procedure, ESTIMATE statement
  - ADJDFE= option, 464
  - ADJUST= option, 465
  - ALPHA= option, 465
  - CHISQ option, 466
  - CL option, 466
  - CORR option, 466
  - COV option, 466
  - DF= option, 467
  - DIVISOR= option, 467
  - E option, 467
  - JOINT option, 468
  - LOWER option, 469
  - NOFILL option, 469
  - ODS table names, 477
  - SEED= option, 471
  - SINGULAR= option, 471
  - STEPDOWN option, 471
  - TESTVALUE option, 472
  - UPPER option, 473
- SURVEYREG procedure, LSMEANS statement
  - ADJDFE= option, 481
  - ADJUST= option, 482
  - ALPHA= option, 484
  - AT= option, 484
  - BYLEVEL option, 485
  - CL option, 485
  - CORR option, 485
  - COV option, 485
  - DF= option, 485
  - DIFF option, 485
  - E option, 486
  - LINES option, 487
  - MEANS or NOMEANS option, 487
  - OBSMARGINS= option, 488
  - ODS graph names, 494
  - ODS table names, 494
  - PDIFF option, 488
  - PLOTS= option, 488
  - SEED= option, 492
  - SINGULAR= option, 492
  - STEPDOWN option, 493
- SURVEYREG procedure, LSMESTIMATE statement
  - ADJDFE= option, 498
  - ADJUST= option, 499
  - ALPHA= option, 499
  - AT= option, 499
  - BYLEVEL option, 499
  - CHISQ option, 500
  - CL option, 500
  - CORR option, 500
  - COV option, 500
  - DF= option, 500
  - DIVISOR= option, 501
  - E option, 501
  - ELSM option, 501
  - JOINT option, 502
  - LOWER option, 503
  - OBSMARGINS= option, 503
  - ODS table names, 507
  - PLOTS= option, 504
  - SEED= option, 505
  - SINGULAR= option, 505
  - STEPDOWN option, 505
  - TESTVALUE= option, 506
  - UPPER option, 507
- SURVEYREG procedure, SLICE statement
  - ADJUST= option, 482
  - ALPHA= option, 484
  - AT= option, 484
  - BYLEVEL option, 485
  - CL option, 485



- CORR option, 485
- COV option, 485
- DF= option, 485
- DIFF option, 485
- E option, 486
- LINES option, 487
- MEANS or NOMEANS option, 487
- NOF option, 528
- OBSMARGINS= option, 488
- ODS table names, 529
- PDIFF option, 488
- PLOTS= option, 488
- SEED= option, 492
- SIMPLE= option, 528
- SINGULAR= option, 492
- SLICEBY= option, 528
- STEPDOWN option, 493
- SURVEYREG procedure, TEST statement
  - CHISQ option, 531
  - DDF= option, 531
  - E option, 531
  - E1 option, 532
  - E2 option, 532
  - E3 option, 532
  - HTYPE= option, 532
  - INTERCEPT option, 532
  - ODS table names, 532
- TCALIS procedure, NLOPTIONS statement
  - ABSCONV option, 510
  - ABSFCNV option, 510
  - ABSGCONV option, 511
  - ABSGTOL option, 511
  - ABSTOL option, 510
  - ABSXCONV option, 511
  - ABSXTOL option, 511
  - ASINGULAR= option, 511
  - FCONV option, 512
  - FCONV2 option, 512
  - FSIZE option, 513
  - FTOL option, 512
  - FTOL2 option, 512
  - GCONV option, 513
  - GCONV2 option, 513
  - GTOL option, 513
  - GTOL2 option, 513
  - HESCAL option, 514
  - HS option, 514
  - INHESSIAN option, 514
  - INSTEP option, 515
  - LCDEACT= option, 515
  - LCEPSILON= option, 515
  - LCSINGULAR= option, 516
  - LINESEARCH option, 516
  - LIS option, 516
  - LSP option, 516
  - LSPRECISION option, 516
  - MAXFU option, 517
  - MAXFUNC option, 517
  - MAXIT option, 517
  - MAXITER option, 517
  - MAXSTEP option, 517
  - MAXTIME option, 518
  - MINIT option, 518
  - MINITER option, 518
  - MSINGULAR= option, 518
  - REST option, 518
  - RESTART option, 518
  - SINGULAR= option, 519
  - SOCKET option, 519
  - TECH option, 519
  - TECHNIQUE option, 519
  - UPD option, 520
  - VSINGULAR= option, 521
  - XSIZE option, 521
  - XTOL option, 521
- TECH option
  - NLOPTIONS statement (GLIMMIX), 519
  - NLOPTIONS statement (HPMIXED), 519
  - NLOPTIONS statement (TCALIS), 519
- TECHNIQUE option
  - NLOPTIONS statement (GLIMMIX), 519
  - NLOPTIONS statement (HPMIXED), 519
  - NLOPTIONS statement (TCALIS), 519
- TEST statement
  - ORTHOREG procedure, 530
  - PLM procedure, 530
  - SURVEYPHREG procedure, 530
  - SURVEYREG procedure, 530
- TESTVALUE= option
  - ESTIMATE statement (LOGISTIC), 472
  - ESTIMATE statement (ORTHOREG), 472
  - ESTIMATE statement (PHREG), 472
  - ESTIMATE statement (PLM), 472
  - ESTIMATE statement (SURVEYLOGISTIC), 472
  - ESTIMATE statement (SURVEYPHREG), 472
  - ESTIMATE statement (SURVEYREG), 472
  - LSMESTIMATE statement (GENMOD), 506
  - LSMESTIMATE statement (LOGISTIC), 506
  - LSMESTIMATE statement (MIXED), 506
  - LSMESTIMATE statement (ORTHOREG), 506
  - LSMESTIMATE statement (PHREG), 506
  - LSMESTIMATE statement (PLM), 506

- LSMESTIMATE statement
  - (SURVEYLOGISTIC), 506
- LSMESTIMATE statement
  - (SURVEYPHREG), 506
- LSMESTIMATE statement (SURVEYREG), 506
- UNPACK option
  - EFFECTPLOT statement, 445
- UPD option
  - NLOPTIONS statement (GLIMMIX), 520
  - NLOPTIONS statement (HPMIXED), 520
  - NLOPTIONS statement (TCALIS), 520
- UPDATE option
  - NLOPTIONS statement (GLIMMIX), 520
  - NLOPTIONS statement (HPMIXED), 520
  - NLOPTIONS statement (TCALIS), 520
- UPPER option
  - ESTIMATE statement (LOGISTIC), 473
  - ESTIMATE statement (ORTHOREG), 473
  - ESTIMATE statement (PHREG), 473
  - ESTIMATE statement (PLM), 473
  - ESTIMATE statement
    - (SURVEYLOGISTIC), 473
  - ESTIMATE statement (SURVEYPHREG), 473
  - ESTIMATE statement (SURVEYREG), 473
  - LSMESTIMATE statement (GENMOD), 507
  - LSMESTIMATE statement (LOGISTIC), 507
  - LSMESTIMATE statement (MIXED), 507
  - LSMESTIMATE statement (ORTHOREG), 507
  - LSMESTIMATE statement (PHREG), 507
  - LSMESTIMATE statement (PLM), 507
  - LSMESTIMATE statement
    - (SURVEYLOGISTIC), 507
  - LSMESTIMATE statement
    - (SURVEYPHREG), 507
  - LSMESTIMATE statement (SURVEYREG), 507
- VSINGULAR= option
  - NLOPTIONS statement (GLIMMIX), 521
  - NLOPTIONS statement (HPMIXED), 521
  - NLOPTIONS statement (TCALIS), 521
- WITHIN option
  - EFFECT statement, lag (GLIMMIX), 422
  - EFFECT statement, lag (GLMSELECT), 422
  - EFFECT statement, lag (HPMIXED), 422
  - EFFECT statement, lag (LOGISTIC), 422
  - EFFECT statement, lag (ORTHOREG), 422
  - EFFECT statement, lag (PHREG), 422
  - EFFECT statement, lag (PLS), 422
- EFFECT statement, lag (ROBUSTREG), 422
- EFFECT statement, lag
  - (SURVEYLOGISTIC), 422
- EFFECT statement, lag (SURVEYREG), 422
- X= option
  - EFFECTPLOT statement, 445
- XCONV option
  - NLOPTIONS statement (GLIMMIX), 521
  - NLOPTIONS statement (HPMIXED), 521
  - NLOPTIONS statement (TCALIS), 521
- XSIZE option
  - NLOPTIONS statement (GLIMMIX), 521
  - NLOPTIONS statement (HPMIXED), 521
  - NLOPTIONS statement (TCALIS), 521
- XTOL option
  - NLOPTIONS statement (GLIMMIX), 521
  - NLOPTIONS statement (HPMIXED), 521
  - NLOPTIONS statement (TCALIS), 521
- Y= option
  - EFFECTPLOT statement, 446
- YRANGE= option
  - EFFECTPLOT statement, 446

## Your Turn

---

We welcome your feedback.

- If you have comments about this book, please send them to **`yourturn@sas.com`**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **`suggest@sas.com`**.



# SAS® Publishing Delivers!

Whether you are new to the work force or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart. Visit us online at [support.sas.com/bookstore](http://support.sas.com/bookstore).

## SAS® Press

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from SAS Press. Written by experienced SAS professionals from around the world, SAS Press books deliver real-world insights on a broad range of topics for all skill levels.

**[support.sas.com/saspress](http://support.sas.com/saspress)**

## SAS® Documentation

To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information: SAS documentation. We currently produce the following types of reference documentation to improve your work experience:

- Online help that is built into the software.
- Tutorials that are integrated into the product.
- Reference documentation delivered in HTML and PDF – **free** on the Web.
- Hard-copy books.

**[support.sas.com/publishing](http://support.sas.com/publishing)**

## SAS® Publishing News

Subscribe to SAS Publishing News to receive up-to-date information about all new SAS titles, author podcasts, and new Web site features via e-mail. Complete instructions on how to subscribe, as well as access to past issues, are available at our Web site.

**[support.sas.com/spn](http://support.sas.com/spn)**



**THE  
POWER  
TO KNOW®**

