

# The GLIMMIX Procedure

## Contents

---

<b>OVERVIEW</b> . . . . .	5
Basic Features . . . . .	6
Assumptions . . . . .	6
Notation for the Generalized Linear Mixed Model . . . . .	7
The Basic Model . . . . .	8
G-side and R-side Random Effects . . . . .	8
Relationship with Generalized Linear Models . . . . .	9
PROC GLIMMIX Contrasted with Other SAS Procedures . . . . .	10
<b>GETTING STARTED</b> . . . . .	11
Logistic Regressions with Random Intercepts . . . . .	11
<b>SYNTAX</b> . . . . .	18
PROC GLIMMIX Statement . . . . .	18
BY Statement . . . . .	33
CLASS Statement . . . . .	33
CONTRAST Statement . . . . .	34
ESTIMATE Statement . . . . .	37
FREQ Statement . . . . .	42
ID Statement . . . . .	42
LSMEANS Statement . . . . .	43
LSMESTIMATE Statement . . . . .	56
MODEL Statement . . . . .	62
Response Variable Options . . . . .	63
Model Options . . . . .	64
NLOPTIONS Statement . . . . .	76
OUTPUT Statement . . . . .	85
PARMS Statement . . . . .	89
RANDOM Statement . . . . .	94
WEIGHT Statement . . . . .	108
Programming Statements . . . . .	108
User-Defined Link or Variance Function . . . . .	110
Implied Variance Functions . . . . .	110
Automatic Variables . . . . .	111
<b>DETAILS</b> . . . . .	113
Generalized Linear Models Theory . . . . .	113

Maximum Likelihood . . . . .	114
Quasi-Likelihood for Independent Data . . . . .	117
Effects of Adding Overdispersion . . . . .	118
Generalized Linear Mixed Models Theory . . . . .	119
Model or Integral Approximation . . . . .	119
Pseudo-Likelihood Estimation Based on Linearization . . . . .	120
GLM-Mode or GLMM-Mode . . . . .	125
Satterthwaite Degrees of Freedom Approximation . . . . .	126
Empirical Covariance (“Sandwich”) Estimators . . . . .	128
Processing by Subjects . . . . .	129
Radial Smoothing Based on Mixed Models . . . . .	131
From Penalized Splines to Mixed Models . . . . .	132
Knot Selection . . . . .	133
Odds and Odds Ratio Estimation . . . . .	138
The Odds Ratio Estimates Table . . . . .	140
Odds or Odds Ratio . . . . .	142
Odds Ratios in Multinomial Models . . . . .	142
Parameterization of Generalized Linear Mixed Models . . . . .	143
Intercept . . . . .	143
Regression Effects . . . . .	144
Main Effects . . . . .	144
Interaction Effects . . . . .	144
Nested Effects . . . . .	145
Continuous-Nesting-Class Effects . . . . .	146
Continuous-by-Class Effects . . . . .	146
General Effects . . . . .	146
Implications of the Non-Full-Rank Parameterization . . . . .	147
Missing Level Combinations . . . . .	148
Response Level Ordering and Referencing . . . . .	148
Comparing PROC GLIMMIX with PROC MIXED . . . . .	149
Singly or Doubly Iterative Fitting . . . . .	151
Default Estimation Techniques . . . . .	153
Choosing an Optimization Algorithm . . . . .	153
First- or Second-Order Algorithms . . . . .	153
Algorithm Descriptions . . . . .	155
Remote Monitoring . . . . .	158
Default Output . . . . .	160
Model Information . . . . .	160
Class Level Information . . . . .	160
Number of Observations . . . . .	160
Response Profile . . . . .	161
Dimensions . . . . .	161
Optimization Information . . . . .	161
Iteration History . . . . .	162
Convergence Status . . . . .	163
Fit Statistics . . . . .	163
Covariance Parameter Estimates . . . . .	164
Type III Tests of Fixed Effects . . . . .	164

Notes on Output Statistics . . . . .	165
ODS Table Names . . . . .	166
ODS Graphics (Experimental) . . . . .	168
Statistical Graphics for LS-Mean Comparisons . . . . .	169
ODS Graph Names . . . . .	174
<b>EXAMPLES</b> . . . . .	176
Example 1. Binomial Counts in Randomized Blocks . . . . .	176
Example 2. Mating Experiment with Crossed Random Effects . . . . .	189
Example 3. Smoothing Disease Rates; Standardized Mortality Ratios . . . . .	198
Example 4. Quasi-Likelihood Estimation for Proportions with Unknown Distribution . . . . .	207
Example 5. Joint Modeling of Binary and Count Data . . . . .	216
Example 6. Radial Smoothing of Repeated Measures Data . . . . .	224
Example 7. Isotonic Contrasts for Ordered Alternatives . . . . .	235
<b>REFERENCES</b> . . . . .	236
<b>SUBJECT INDEX</b> . . . . .	243
<b>SYNTAX INDEX</b> . . . . .	249



# The GLIMMIX Procedure

---

## Overview

The GLIMMIX procedure fits statistical models to data with correlations or nonconstant variability and where the response is not necessarily normally distributed. These models are known as generalized linear mixed models (GLMM).

The GLMMs, like linear mixed models, assume normal (Gaussian) random effects. Conditional on these random effects, data can have any distribution in the exponential family. The exponential family comprises many of the elementary discrete and continuous distributions. The binary, binomial, Poisson, and negative binomial distributions, for example, are discrete members of this family. The normal, beta, gamma, and chi-square distributions are representatives of the continuous distributions in this family. In the absence of random effects, the GLIMMIX procedure fits generalized linear models (fit by the GENMOD procedure).

GLMMs are useful for

- estimating trends in disease rates
- modeling CD4 counts in a clinical trial over time
- modeling the proportion of infected plants on experimental units in a design with randomly selected treatments or randomly selected blocks
- predicting the probability of high ozone levels in counties
- modeling skewed data over time
- analyzing customer preference
- joint modeling of multivariate outcomes

Such data often display correlations among some or all observations as well as non-normality. The correlations can arise from repeated observation of the same sampling units, shared random effects in an experimental design, spatial (temporal) proximity, multivariate observations, and so on.

The GLIMMIX procedure does not fit hierarchical models with non-normal random effects. With the GLIMMIX procedure you select the distribution of the response variable conditional on normally distributed random effects.

For more information on the differences between the GLIMMIX procedure and SAS procedures that specialize in certain subsets of the GLMM models, see the [“PROC GLIMMIX Contrasted with Other SAS Procedures”](#) section on page 10.

---

## Basic Features

The GLIMMIX procedure enables you to specify a generalized linear mixed model and to perform confirmatory inference in such models. The syntax is similar to that of the MIXED procedure and includes [CLASS](#), [MODEL](#), and [RANDOM](#) statements. For instructions on how to specify PROC MIXED REPEATED effects with PROC GLIMMIX, see “[Comparing PROC GLIMMIX with PROC MIXED](#)”. The following are some of the basic features of PROC GLIMMIX.

- [SUBJECT=](#) and [GROUP=](#) options, which enable blocking of variance matrices and parameter heterogeneity
- choice of linearization about expected values or expansion about current solutions of best linear unbiased predictors
- flexible covariance structures for random and residual random effects, including variance components, unstructured, autoregressive, and spatial structures
- [CONTRAST](#), [ESTIMATE](#), [LSMEANS](#), and [LSMESTIMATE](#) statements, which produce hypothesis tests and estimable linear combinations of effects
- [NLOPTIONS](#) statement, which enables you to exercise control over the numerical optimization. You can choose techniques, update methods, line search algorithms, convergence criteria, and more. Or, you can choose the default optimization strategies selected for the particular class of model you are fitting
- computed variables with SAS programming statements inside of PROC GLIMMIX (except for variables listed in the [CLASS](#) statement). These computed variables can appear in the [MODEL](#), [RANDOM](#), [WEIGHT](#), or [FREQ](#) statements.
- grouped data analysis
- user-specified link and variance functions
- choice of model-based variance-covariance estimators for the fixed effects or empirical (sandwich) estimators to make analysis robust against misspecification of the covariance structure and to adjust for small-sample bias
- joint modeling for multivariate data. For example, you can model binary and normal responses from a subject jointly and use random effects to relate (fuse) the two outcomes.
- multinomial models for ordinal and nominal outcomes
- univariate and multivariate low-rank mixed model smoothing

---

## Assumptions

The primary assumptions underlying the analyses performed by PROC GLIMMIX are as follows:

- If the model contains random effects, the distribution of the data conditional on the random effects is known. This distribution is either a member of the

exponential family of distributions or one of the supplementary distributions provided by the GLIMMIX procedure. In models without random effects, the unconditional (marginal) distribution is assumed to be known for maximum likelihood estimation, or the first two moments are known in the case of quasi-likelihood estimation.

- The conditional expected value of the data takes the form of a linear mixed model after a monotonic transformation is applied.
- The problem of fitting the GLMM can be cast as a singly or doubly iterative optimization problem. The objective function for the optimization is a function of either the actual log likelihood, an approximation to the log likelihood, or the log likelihood of an approximated model.

For a model containing random effects, the GLIMMIX procedure, by default, estimates the parameters by applying pseudo-likelihood techniques as in Wolfinger and O'Connell (1993) and Breslow and Clayton (1993). In a model without random effects (GLM models), PROC GLIMMIX estimates the parameters by maximum likelihood, restricted maximum likelihood, or quasi-likelihood. See the [“Singly or Doubly Iterative Fitting”](#) section on page 151 on when the GLIMMIX procedure applies noniterative, singly and doubly iterative algorithms, and the [“Default Estimation Techniques”](#) section on page 153 on the default estimation methods.

Once the parameters have been estimated, you can perform statistical inferences for the fixed effects and covariance parameters of the model. Tests of hypotheses for the fixed effects are based on Wald-type tests and the estimated variance-covariance matrix.

PROC GLIMMIX uses the Output Delivery System (ODS) for displaying and controlling the output from SAS procedures. ODS enables you to convert any of the output from PROC GLIMMIX into a SAS data set. See the [“ODS Table Names”](#) section on page 166.

Experimental ODS statistical graphics are available with the GLIMMIX procedure. For more information, see the [PLOTS](#) options in the PROC GLIMMIX and [LSMEANS](#) statements. For general information about ODS graphics and the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS” (*SAS/STAT User's Guide*). For specific information about the graphics available in the GLIMMIX procedure, see the [“ODS Graphics”](#) section on page 168.

---

## Notation for the Generalized Linear Mixed Model

This section introduces the mathematical notation used throughout the chapter to describe the generalized linear mixed model (GLMM). See the [“Details”](#) section on page 113 for a description of the fitting algorithms and the mathematical-statistical details.

### The Basic Model

Suppose  $\mathbf{Y}$  represents the  $(n \times 1)$  vector of observed data and  $\boldsymbol{\gamma}$  is a  $(r \times 1)$  vector of random effects. Models fit by the GLIMMIX procedure assume that

$$E[\mathbf{Y}|\boldsymbol{\gamma}] = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma})$$

where  $g(\cdot)$  is a differentiable monotonic link function and  $g^{-1}(\cdot)$  is its inverse. The matrix  $\mathbf{X}$  is a  $(n \times p)$  matrix of rank  $k$ , and  $\mathbf{Z}$  is a  $(n \times r)$  design matrix for the random effects. The random effects are assumed to be normally distributed with mean  $\mathbf{0}$  and variance matrix  $\mathbf{G}$ .

The GLMM contains a linear mixed model inside the inverse link function. This model component is referred to as the linear predictor,

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$$

The variance of the observations, conditional on the random effects, is

$$\text{var}[\mathbf{Y}|\boldsymbol{\gamma}] = \mathbf{A}^{1/2}\mathbf{R}\mathbf{A}^{1/2}$$

The matrix  $\mathbf{A}$  is a diagonal matrix and contains the variance functions of the model. The variance function expresses the variance of a response as a function of the mean. The GLIMMIX procedure determines the variance function from the `DIST=` option in the `MODEL` statement or from the user-supplied variance function (see the “[Implied Variance Functions](#)” section on page 110). The matrix  $\mathbf{R}$  is a variance matrix specified by the user through the `RANDOM` statement. If the conditional distribution of the data contains an additional scale parameter, it is either part of the variance functions or part of the  $\mathbf{R}$  matrix. For example, the gamma distribution with mean  $\mu$  has variance function  $a(\mu) = \mu^2$  and  $\text{var}[Y|\boldsymbol{\gamma}] = \mu^2\phi$ . If your model calls for G-side random effects only (see below), the procedure models  $\mathbf{R} = \phi\mathbf{I}$ , where  $\mathbf{I}$  is the identity matrix. [Table 10](#) on page 110 identifies the distributions for which  $\phi \equiv 1$ .

### G-side and R-side Random Effects

The GLIMMIX procedure distinguishes two types of random effects. Depending on whether the variance of the random effect is contained in  $\mathbf{G}$  or in  $\mathbf{R}$ , these are referred to as “G-side” and “R-side” random effects. R-side effects are also called “residual” effects. Simply put, if a random effect is an element of  $\boldsymbol{\gamma}$ , it is a G-side effect; otherwise, it is an R-side effect. Models without G-side effects are also known as marginal (or population-averaged) models. Models fit with the GLIMMIX procedure can have none, one, or more of each type of effect.

Note that an R-side effect in the GLIMMIX procedure is equivalent to a `REPEATED` effect in the `MIXED` procedure. In the GLIMMIX procedure all random effects are specified through the `RANDOM` statement.

The columns of  $\mathbf{X}$  are constructed from effects listed on the right-hand side in the `MODEL` statement. Columns of  $\mathbf{Z}$  and the variance matrices  $\mathbf{G}$  and  $\mathbf{R}$  are constructed from the `RANDOM` statement.



The  $\mathbf{R}$  matrix is by default the scaled identity matrix,  $\mathbf{R} = \phi\mathbf{I}$ . The scale parameter  $\phi$  is set to one if the distribution does not have a scale parameter, for example, in the case of the the binary, binomial, Poisson, and exponential distribution (see [Table 10](#) on page 110). To specify a different  $\mathbf{R}$  matrix, use the `RANDOM` statement with the `_RESIDUAL_` keyword or the `RESIDUAL` option. For example, to specify that the `TIME` effect for each patient is an R-side effect with a first-order autoregressive covariance structure, use the `RESIDUAL` option:

```
random time / type=ar(1) subject=patient residual;
```

To add a multiplicative overdispersion parameter, use the `_RESIDUAL_` keyword

```
random _residual_;
```

You specify the link function  $g(\cdot)$  with the `LINK=` option of the `MODEL` statement or with programming statements. You specify the variance function that controls the matrix  $\mathbf{A}$  with the `DIST=` option of the `MODEL` statement or with programming statements.

Unknown quantities subject to estimation are the fixed-effects parameter vector  $\beta$  and the covariance parameter vector  $\theta$  that comprises all unknowns in  $\mathbf{G}$  and  $\mathbf{R}$ . The random effects  $\gamma$  are not parameters of the model in the sense that they are not estimated. The vector  $\gamma$  is a vector of random variables. The solutions for  $\gamma$  are predictors of these random variables.

Some fitting algorithms require that the best linear unbiased predictors (BLUPs) of  $\gamma$  be computed at every iteration.

### **Relationship with Generalized Linear Models**

Generalized linear models (Nelder and Wedderburn 1972; McCullagh and Nelder 1989) are a special case of GLMMs. If  $\gamma = \mathbf{0}$  and  $\mathbf{R} = \phi\mathbf{I}$ , the GLMM reduces to either a generalized linear model (GLM) or a GLM with overdispersion. For example, if  $\mathbf{Y}$  is a vector of Poisson variables so that  $\mathbf{A}$  is a diagonal matrix containing  $E[\mathbf{Y}] = \boldsymbol{\mu}$  on the diagonal, then the model is a Poisson regression model for  $\phi = 1$  and overdispersed relative to a Poisson distribution for  $\phi > 1$ . Since the Poisson distribution does not have an extra scale parameter, you can model overdispersion by adding the statement

```
random _residual_;
```

to your `GLIMMIX` statements. If the only random effect is an overdispersion effect, `PROC GLIMMIX` fits the model by (restricted) maximum likelihood and not one of the methods specific to GLMMs.

---

## PROC GLIMMIX Contrasted with Other SAS Procedures

The GLIMMIX procedure generalizes the MIXED and GENMOD procedures in two important ways. First, the response can have a nonnormal distribution. The MIXED procedure assumes that the response is normally (Gaussian) distributed. Second, the GLIMMIX procedure incorporates random effects in the model and so allows for subject-specific (conditional) and population-averaged (marginal) inference. The GENMOD procedure only allows for marginal inference.

The GLIMMIX and MIXED procedure are closely related; see the syntax and feature comparison in the section “[Comparing PROC GLIMMIX with PROC MIXED](#)” on page 149. The remainder of this section compares PROC GLIMMIX with the GENMOD, NLMIXED, LOGISTIC, and CATMOD procedures.

The GENMOD procedure fits generalized linear models for independent data by maximum likelihood. It can also handle correlated data through the marginal GEE approach of Liang and Zeger (1986) and Zeger and Liang (1986). The GEE implementation in the GENMOD procedure is a marginal method that does not incorporate random effects. The GEE estimation in the GENMOD procedure relies on R-side covariances only, and the unknown parameters in  $\mathbf{R}$  are estimated by the method of moments. The GLIMMIX procedure allows G-side random effects and R-side covariances. PROC GLIMMIX can fit marginal (GEE-type) models, but the covariance parameters are not estimated by method of moments. The parameters are estimated by likelihood-based techniques.

Many of the fit statistics and tests in the GENMOD procedure are based on the likelihood. In a GLMM it is not always possible to derive the log likelihood of the data. Even if the log likelihood is tractable, it may be computationally infeasible. In some cases, the objective function must be constructed based on a substitute model. In other cases, only the first two moments of the marginal distribution can be approximated. Consequently, obtaining likelihood-based tests and statistics is difficult in the majority of generalized linear mixed models. The GLIMMIX procedure relies heavily on linearization and Taylor-series techniques to construct Wald-type test statistics and confidence intervals. Likelihood ratio tests and confidence intervals are not available in the GLIMMIX procedure.

The NLMIXED procedure also fits generalized linear mixed models but the class of models it can accommodate is more narrow. The NLMIXED procedure relies on approximating the marginal log likelihood by integral approximation through Gaussian quadrature. Like the GLIMMIX procedure, the NLMIXED procedure defines the problem of obtaining solutions for the parameter estimates as an optimization problem. The objective function for the NLMIXED procedure is the marginal log likelihood obtained by integrating out the random effects from the joint distribution of responses and random effects using quadrature techniques. Although these are very accurate, the number of random effects that can be practically managed is limited. Also, R-side random effects cannot be accommodated with the NLMIXED procedure. The GLIMMIX procedure, on the other hand, determines by default the marginal log likelihood as that of an approximate linear mixed model. This allows multiple random effects, nested and crossed random effects, multiple cluster types, and R-side random

components. The disadvantage is a doubly iterative fitting algorithm and the absence of a true log likelihood.

The LOGISTIC and CATMOD procedures also fit generalized linear models; PROC LOGISTIC accommodates the independence case only. Binary, binomial, multinomial models for ordered data, and generalized logit models that can be fit with PROC LOGISTIC can also be fit with the GLIMMIX procedure. The diagnostic tools and capabilities specific to such data implemented in the LOGISTIC procedure go beyond the capabilities of the GLIMMIX procedure.

---

## Getting Started

---

### Logistic Regressions with Random Intercepts

Researchers investigated the performance of two medical procedures in a multicenter study. They randomly selected 15 centers for inclusion. One of the study goals was to compare the occurrence of side effects for the procedures. In each center  $n_A$  patients were randomly selected and assigned to procedure “A,” and  $n_B$  patients were randomly assigned to procedure “B”. The following DATA step creates the data set for the analysis.

```

data multicenter;
  input center group$ n sideeffect;
  datalines;
1  A  32  14
1  B  33  18
2  A  30   4
2  B  28   8
3  A  23  14
3  B  24   9
4  A  22   7
4  B  22  10
5  A  20   6
5  B  21  12
6  A  19   1
6  B  20   3
7  A  17   2
7  B  17   6
8  A  16   7
8  B  15   9
9  A  13   1
9  B  14   5
10 A  13   3
10 B  13   1
11 A  11   1
11 B  12   2
12 A  10   1
12 B   9   0
13 A   9   2
13 B   9   6
14 A   8   1

```

```

14 B 8 1
15 A 7 1
15 B 8 0
;

```

The variable `group` identifies the two procedures, `n` is the number of patients who received a given procedure in a particular center, and `sideeffect` is the number of patients who reported side effects.

If  $Y_{iA}$  and  $Y_{iB}$  denote the number of patients in center  $i$  who report side effects for procedures  $A$  and  $B$ , respectively, then—for a given center—these are independent binomial random variables. To model the probability of side effects for the two drugs,  $\pi_{iA}$  and  $\pi_{iB}$ , you need to account for the fixed group effect and the random selection of centers. One possibility is to assume a model that relates group and center effects linearly to the logit of the probabilities:

$$\log \left\{ \frac{\pi_{iA}}{1 - \pi_{iA}} \right\} = \beta_0 + \beta_A + \gamma_i$$

$$\log \left\{ \frac{\pi_{iB}}{1 - \pi_{iB}} \right\} = \beta_0 + \beta_B + \gamma_i$$

In this model,  $\beta_A - \beta_B$  measures the difference in the logits of experiencing side effects, and the  $\gamma_i$  are independent random variables due to the random selection of centers. If you think of  $\beta_0$  as the overall intercept in the model, then the  $\gamma_i$  are random intercept adjustments. Observations from the same center receive the same adjustment, and these vary randomly from center to center with variance  $\text{var}[\gamma_i] = \sigma_c^2$ .

Since  $\pi_{iA}$  is the conditional mean of the sample proportion,  $E[Y_{iA}/n_{iA}|\gamma_i] = \pi_{iA}$ , you can model the sample proportions as binomial ratios in a generalized linear mixed model. The following statements request this analysis under the assumption of normally distributed center effects with equal variance and a logit link function.

```

proc glimmix data=multicenter;
  class center group;
  model sideeffect/n = group / solution;
  random intercept / subject=center;
run;

```

The `PROC GLIMMIX` statement invokes the procedure. The `CLASS` statement instructs the procedure to treat the variables `center` and `group` as classification variables. The `MODEL` statement specifies the response variable as a sample proportion using the `events/trials` syntax. In terms of the previous formulas, `sideeffect/n` corresponds to  $Y_{iA}/n_{iA}$  for observations from Group A and to  $Y_{iB}/n_{iB}$  for observations from Group B. The `SOLUTION` option in the `MODEL` statement requests a listing of the solutions for the fixed-effects parameter estimates. Note that because of the `events/trials` syntax, the GLIMMIX procedure defaults to the binomial distribution,

and that distribution’s default link is the logit link. The **RANDOM** statement specifies that the linear predictor contains an intercept term that randomly varies at the level of the **center** effect. In other words, a random intercept is drawn separately and independently for each center in the study.

The results of this analysis are shown in the following tables.

The “Model Information Table” in [Figure 1](#) summarizes important information about the model you fit and about aspects of the estimation technique.

The GLIMMIX Procedure	
Model Information	
Data Set	WORK.MULTICENTER
Response Variable (Events)	sideeffect
Response Variable (Trials)	n
Response Distribution	Binomial
Link Function	Logit
Variance Function	Default
Variance Matrix Blocked By	center
Estimation Technique	Residual PL
Degrees of Freedom Method	Containment

**Figure 1.** Model Information

PROC GLIMMIX recognizes the variables `sideeffect` and `n` as the numerator and denominator in the *events/trials* syntax, respectively. The distribution—conditional on the random center effects—is binomial. The marginal variance matrix is block-diagonal, and observations from the same center form the blocks. The default estimation technique in generalized linear mixed models is residual pseudo-likelihood with a subject-specific expansion (METHOD=RSPL).

In [Figure 2](#), the “Class Level Information” table lists the levels of the variables specified in the **CLASS** statement and the ordering of the levels. The “Number of Observations” table displays the number of observations read and used in the analysis.

Class Level Information		
Class	Levels	Values
center	15	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
group	2	A B
Number of Observations Read		30
Number of Observations Used		30
Number of Events		155
Number of Trials		503

**Figure 2.** Class Level Information and Number of Observations

There are two variables listed in the `CLASS` statement. The `center` variable has fifteen levels, and the `group` variable has two levels. Since the response is specified through the `events/trial` syntax, the “Number of Observations” table also contains the total number of events and trials used in the analysis.

The “Dimensions” table in [Figure 3](#) lists the size of relevant matrices.

Dimensions	
G-side Cov. Parameters	1
Columns in X	3
Columns in Z per Subject	1
Subjects (Blocks in V)	15
Max Obs per Subject	2

**Figure 3.** Dimensions

There are three columns in the **X** matrix, corresponding to an intercept and the two levels of the `group` variable. For each subject (`center`), the **Z** matrix contains only an intercept column.

The “Optimization Information” table in [Figure 4](#) provides information about the methods and size of the optimization problem.

Optimization Information	
Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	1
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Profiled
Starting From	Data

**Figure 4.** Optimization Information

The default optimization technique for generalized linear mixed models is the Quasi-Newton method. Because a residual likelihood technique is used to compute the objective function, only the covariance parameters are participating in the optimization. A lower boundary constraint is placed on the variance component for the random `center` effect. The solution for this variance cannot be less than zero.

The “Iteration History” table in [Figure 5](#) displays information about the progress of the optimization process.

Iteration History						
Iteration	Restarts	Subiterations	Objective Function	Change	Max Gradient	
0	0	5	79.688580269	0.11807224	7.851E-7	
1	0	3	81.294622554	0.02558021	8.209E-7	
2	0	2	81.438701534	0.00166079	4.061E-8	
3	0	1	81.444083567	0.00006263	2.311E-8	
4	0	1	81.444265216	0.00000421	0.000025	
5	0	1	81.444277364	0.00000383	0.000023	
6	0	1	81.444266322	0.00000348	0.000021	
7	0	1	81.44427636	0.00000316	0.000019	
8	0	1	81.444267235	0.00000287	0.000017	
9	0	1	81.444275529	0.00000261	0.000016	
10	0	1	81.44426799	0.00000237	0.000014	
11	0	1	81.444274843	0.00000216	0.000013	
12	0	1	81.444268614	0.00000196	0.000012	
13	0	1	81.444274277	0.00000178	0.000011	
14	0	1	81.444269129	0.00000162	9.772E-6	
15	0	0	81.444273808	0.00000000	9.102E-6	

Convergence criterion (PCONV=1.11022E-8) satisfied.

**Figure 5.** Iteration History and Convergence Status

After the initial optimization, the GLIMMIX procedure performed 15 updates before the convergence criterion was met. At convergence, the largest absolute value of the gradient was near zero. This indicates that the process stopped at an extremum of the objective function.

The “Fit Statistics” table in [Figure 6](#) lists information about the fitted model.

Fit Statistics	
-2 Res Log Pseudo-Likelihood	81.44
Generalized Chi-Square	30.69
Gener. Chi-Square / DF	1.10

**Figure 6.** Fit Statistics

Twice the negative of the residual log likelihood in the final pseudo-model equaled 81.44. The ratio of the generalized chi-square statistic and its degrees of freedom is close to 1. This is a measure of the residual variability in the marginal distribution of the data.

The “Covariance Parameter Estimates” table in [Figure 7](#) displays estimates and asymptotic estimated standard errors for all covariance parameters.

Covariance Parameter Estimates			
Cov Parm	Subject	Estimate	Standard Error
Intercept	center	0.6176	0.3181

**Figure 7.** Covariance Parameter Estimates

The variance of the random center intercepts on the logit scale is estimated as  $\hat{\sigma}_c^2 = 0.6176$ .

The “Parameter Estimates” table in [Figure 8](#) displays the solutions for the fixed effects in the model.

Solutions for Fixed Effects						
Effect	group	Estimate	Standard Error	DF	t Value	Pr >  t
Intercept		-0.8071	0.2514	14	-3.21	0.0063
group	A	-0.4896	0.2034	14	-2.41	0.0305
group	B	0	.	.	.	.

**Figure 8.** Parameter Estimates

Because of the fixed-effects parameterization used in the GLIMMIX procedure, the “Intercept” effect is an estimate of  $\beta_0 + \beta_B$ , and the “A” group effect is an estimate of  $\beta_A - \beta_B$ , the log-odds ratio. The associated estimated probabilities of side effects in the two groups are

$$\hat{\pi}_A = \frac{1}{1 + \exp\{0.8071 + 0.4896\}} = 0.2147$$

$$\hat{\pi}_B = \frac{1}{1 + \exp\{0.8071\}} = 0.3085$$

There is a significant difference between the two groups ( $p=0.0305$ ).

The “Type III Tests of Fixed Effect” table in [Figure 9](#) displays significance tests for the fixed effects in the model.

Type III Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
group	1	14	5.79	0.0305

**Figure 9.** Type III Tests of Fixed Effects



Because the `group` effect has only two levels, the  $p$ -value for the effect is the same as in the “Parameter Estimates” table, and the “ $F$  Value” is the square of the “ $t$  Value” shown there.

You can produce the estimates of the average logits in the two groups and their predictions on the scale of the data with the `LSMEANS` statement in PROC GLIMMIX.

```
ods select lsmeans;
proc glimmix data=multicenter;
  class center group;
  model sideeffect/n = group / solution;
  random intercept / subject=center;
  lsmeans group / cl ilink;
run;
```

The `LSMEANS` statement requests the least-squares means of the `group` effect on the logit scale. The `CL` option requests their confidence limits. The `ILINK` option adds estimates, standard errors, and confidence limits on the mean (probability) scale. The table in Figure 10 displays the results.

The GLIMMIX Procedure								
group Least Squares Means								
group	Estimate	Standard Error	DF	t Value	Pr >  t	Alpha	Lower	Upper
A	-1.2966	0.2601	14	-4.99	0.0002	0.05	-1.8544	-0.7388
B	-0.8071	0.2514	14	-3.21	0.0063	0.05	-1.3462	-0.2679
group Least Squares Means								
group	Mean	Standard Error Mean	Lower Mean	Upper Mean				
A	0.2147	0.04385	0.1354	0.3233				
B	0.3085	0.05363	0.2065	0.4334				

**Figure 10.** Least-squares Means

The “Estimate” column displays the least-squares mean estimate on the logit scale, and the “Mean” column represents its mapping onto the probability scale. The “Lower” and “Upper” columns are 95% confidence limits for the logits in the two groups. The “Lower Mean” and “Upper Mean” columns are the corresponding confidence limits for the probabilities of side effects. These limits are obtained by inversely linking the confidence bounds on the linear scale, and thus are not symmetric about the estimate of the probabilities.

---

## Syntax

You can specify the following statements in the GLIMMIX procedure.

```

PROC GLIMMIX < options > ;
  BY variables ;
  CLASS variables ;
  CONTRAST 'label' contrast-specification
            <, contrast-specification > <, ... >
            < / options > ;
  ESTIMATE 'label' contrast-specification <(divisor=n)>
            <, 'label' contrast-specification <(divisor=n)> >><, ... >
            < / options > ;
  FREQ variable ;
  ID variables ;
  LSMEANS fixed-effects < / options > ;
  LSMESTIMATE fixed-effect <'label'> values <divisor=n >
                <, <'label'> values <divisor=n >> <, ... >
                < / options > ;
  MODEL response<(response options)> = < fixed-effects >
            < / options > ;
  MODEL events/trials = < fixed-effects >< / options > ;
  NLOPTIONS < options > ;
  OUTPUT < OUT=SAS-data-set >
            <keyword<(keyword-options)><=name>> ...
            <keyword<(keyword-options)><=name>>< / options > ;
  PARMS (value-list) ... < / options > ;
  RANDOM random-effects < / options > ;
  WEIGHT variable ;
  Programming statements

```

The **CONTRAST**, **ESTIMATE**, **LSMEANS**, **LSMESTIMATE**, and **RANDOM** statements can appear multiple times; all other statements can appear only once with the exception of programming statements. The **PROC GLIMMIX** and **MODEL** statements are required, and the **MODEL** statement must appear after the **CLASS** statement if a **CLASS** statement is included.

---

## PROC GLIMMIX Statement

```
PROC GLIMMIX < options >;
```

The PROC GLIMMIX statement invokes the procedure. You can specify the following options.

### **ABSPCONV**=*r*

specifies an absolute parameter estimate convergence criterion for doubly iterative estimation methods. For such methods, the GLIMMIX procedure by default examines the *relative* change in parameter estimates between optimizations (see **PCONV**=).

The purpose of the `ABSPCONV=` criterion is to stop the process when the *absolute* change in parameter estimates is less than the tolerance criterion  $r$ . The criterion is based on fixed effects and covariance parameters.

Note that this convergence criterion does not affect the convergence criteria applied within any individual optimization. In order to change the convergence behavior within an optimization, you can change the `ABSCONV=`, `ABSFCONV=`, `ABSGCONV=`, `ABSXCONV=`, `FCONV=`, or `GCONV=` options of the `NLOPTIONS` statement.

### **ASYCORR**

produces the asymptotic correlation matrix of the covariance parameter estimates. It is computed from the corresponding asymptotic covariance matrix (see the description of the `ASYCOV` option, which follows).

### **ASYCOV**

requests that the asymptotic covariance matrix of the covariance parameter estimates be displayed. By default, this matrix is the observed inverse Fisher information matrix, which equals  $m\mathbf{H}^{-1}$ , where  $\mathbf{H}$  is the Hessian (second derivative) matrix of the objective function. The factor  $m$  equals 1 in a GLM and equals 2 in a GLMM.

When you use the `SCORING=` option and PROC GLIMMIX converges without stopping the scoring algorithm, the procedure uses the expected Hessian matrix to compute the covariance matrix instead of the observed Hessian. Regardless of whether a scoring algorithm is used or the number of scoring iterations has been exceeded, you can request that the asymptotic covariance matrix be based on the expected Hessian with the `EXPHESSIAN` option of the PROC GLIMMIX statement. If a residual scale parameter is profiled from the likelihood equation, the asymptotic covariance matrix is adjusted for the presence of this parameter; details of this adjustment process are found in Wolfinger, Tobias, and Sall (1994) and in the “[Estimated Precision of Estimates](#)” section on page 122.

### **CHOLESKY | CHOL**

requests that the mixed model equations are constructed and solved using the Cholesky root of the  $\mathbf{G}$  matrix. This option applies only to estimation methods that involve mixed model equations. The Cholesky root algorithm has greater numerical stability but also requires more computing resources. When the estimated  $\mathbf{G}$  matrix is not positive definite during a particular function evaluation, PROC GLIMMIX switches to the Cholesky algorithm for that evaluation and returns to the regular algorithm if  $\hat{\mathbf{G}}$  becomes positive definite again. When the CHOLESKY option is in effect, the procedure applies the algorithm all the time.

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

names the SAS data set to be used by PROC GLIMMIX. The default is the most recently created data set.

### **EMPIRICAL<=CLASSICAL | HC0>**

### **EMPIRICAL<=DF | HC1>**

### **EMPIRICAL<=ROOT | HC2>**

### **EMPIRICAL<=FIRORES | HC3>**

### **EMPIRICAL<=FIROEEQ<(r)>>**

requests that the covariance matrix of the fixed-effects parameter estimates is computed by using one of the asymptotically consistent estimators, known as *sandwich* or *empirical* estimators. The name stems from the layering of the estimator. An empirically based estimate of the inverse variance of the fixed-effects parameter estimates (the “meat”) is wrapped by the model-based variance estimate (the “bread”).

Empirical estimators are useful for obtaining inferences for the fixed effects that are not sensitive to the choice of the covariance model. In nonmixed models, they are useful, for example, to allay the effects of variance heterogeneity on the tests of fixed effects.

For a general model, let  $\mathbf{Y}$  denote the response with mean  $\boldsymbol{\mu}$  and variance  $\boldsymbol{\Sigma}$ , and let  $\mathbf{D}$  be the matrix of first derivatives of  $\boldsymbol{\mu}$  with respect to the fixed effects  $\boldsymbol{\beta}$ . The classical sandwich estimator (Huber 1967; White 1980; Liang and Zeger 1986) is

$$\hat{\boldsymbol{\Omega}} \left( \sum_{i=1}^m \hat{\mathbf{D}}_i' \hat{\boldsymbol{\Sigma}}_i^{-1} \mathbf{e}_i \mathbf{e}_i' \hat{\boldsymbol{\Sigma}}_i^{-1} \hat{\mathbf{D}}_i \right) \hat{\boldsymbol{\Omega}}$$

where  $\boldsymbol{\Omega} = (\mathbf{D}'\boldsymbol{\Sigma}^{-1}\mathbf{D})^{-1}$ ,  $\mathbf{e}_i = \mathbf{y}_i - \hat{\boldsymbol{\mu}}_i$ , and  $m$  denotes the number of independent sampling units. If you specify the EMPIRICAL option without further qualifiers, PROC GLIMMIX computes this classical sandwich estimator.

Since the expected value of  $\mathbf{e}_i \mathbf{e}_i'$  does not equal  $\boldsymbol{\Sigma}_i$ , the classical sandwich estimator is biased, particularly if  $m$  is small. The estimator tends to underestimate the variance of  $\hat{\boldsymbol{\beta}}$ . The EMPIRICAL=DF, ROOT, FIRORES, and FIROEEQ estimators are bias-corrected sandwich estimators. Except for the DF estimator, they are based on Taylor series approximations applied to residuals and estimating equations. For uncorrelated data, the EMPIRICAL=FIRORES estimator can be motivated as a jackknife estimator.

In the case of a linear regression model, the various estimators reduce to the *heteroscedasticity-consistent covariance matrix* estimators (HCMM) of White (1980) and MacKinnon and White (1985). The classical estimator, HC0, was found to perform poorly in small samples. Based on simulations in regression models, MacKinnon and White (1985) and Long and Ervin (2000) strongly recommend the HC3 estimator. The sandwich estimators computed by the GLIMMIX procedure can be viewed as an extension of the HC0—HC3 estimators of MacKinnon and White (1985) to accommodate nonnormal data and correlated observations.

For details on the general expression for these estimators and their relationship, see the section “[Empirical Covariance \(“Sandwich”\) Estimators](#)” on page 128.

The EMPIRICAL=DF estimator applies a simple, multiplicative correction factor to the classical estimator (Hinkley 1977). This correction factor is

$$c = \begin{cases} m/(m-k) & m > k \\ 1 & \text{otherwise} \end{cases}$$

where  $k$  is the rank of  $\mathbf{X}$ , and  $m$  equals the sum of all frequencies when PROC GLIMMIX is in GLM-Mode and equals the number of subjects in GLMM-Mode. For example, the statements

```
proc glimmix empirical;
  model y = x;
  random _residual_;
run;
```

fit an overdispersed GLM. PROC GLIMMIX is in GLM-Mode and the individual observations are the independent sampling units from which the sandwich estimator is constructed. If you use a **SUBJECT=** effect on the **RANDOM** statement, however, the procedure fits the model in GLMM-Mode and the subjects represent the sampling units in the construction of the sandwich estimator. In other words, the statements

```
proc glimmix empirical;
  class id;
  model y = x;
  random _residual_ / subject=id type=vc;
run;
```

fit a GEE-type model with independence working covariance structure and subjects (clusters) defined by the levels of ID. See the “GLM-Mode or GLMM-Mode” section on page 125 on how the GLIMMIX procedure determines the estimation mode.

The **EMPIRICAL=ROOT** estimator is based on the residual approximation in Kauermann and Carroll (2001), and the **EMPIRICAL=FIRORES** estimator is based on the approximation in Mancl and DeRouen (2001). The Kauermann and Carroll estimator requires the inverse square root of a nonsymmetric matrix. This square root matrix is obtained from the singular value decomposition in PROC GLIMMIX, thus this sandwich estimator is computationally more demanding than others. In the linear regression case, the Mancl-DeRouen estimator can be motivated as a jackknife estimator, based on the “leave-one-out” estimates of  $\hat{\beta}$ ; refer to MacKinnon and White (1985) for details.

The **EMPIRICAL=FIROEEQ** estimator is based on approximating an unbiased estimating equation (Fay and Graubard 2001). It is computationally less demanding than the estimator of Kauermann and Carroll (2001), and, in certain balanced cases, gives identical results. The optional number  $0 \leq r < 1$  is chosen to provide an upper bound on the correction factor. The default value for  $r$  is 0.75. The diagonal entries of  $\mathbf{A}_i$  are then no greater than 2.

Computation of an empirical variance estimator requires that the data can be processed by independent sampling units. This is always the case in GLMs. In this case,  $m$ , the number of independent units, equals the sum of the frequencies used in the analysis (see “Number of Observations” table). In GLMMs, empirical estimators can only be computed if the data comprise more than one subject as per the “Dimensions” table. See the section “Processing by Subjects” on page 129 on how the GLIMMIX procedure determines whether the data can be processed by subjects. If a GLMM comprises only a single subject for a particular BY group, the model-based variance estimator is used instead of the empirical estimator, and a message is written to the SAS log.

When you specify the **EMPIRICAL** option, PROC GLIMMIX adjusts all standard errors and test statistics involving the fixed-effects parameters.

**EXPHESSIAN**

requests that the expected Hessian matrix be used in computing the covariance matrix of the nonprofiled parameters. By default, the GLIMMIX procedure uses the observed Hessian matrix in computing the asymptotic covariance matrix of covariance parameters in mixed models and the covariance matrix of fixed effects in models without random effects. The EXPHESSIAN option is ignored if the (conditional) distribution is not a member of the exponential family or is unknown. It is also ignored in models for nominal data.

**FDIGITS=*r***

specifies the number of accurate digits in evaluations of the objective function. Fractional values are allowed. The default value is  $r = -\log_{10} \epsilon$ , where  $\epsilon$  is the machine precision. The value of  $r$  is used to compute the interval size for the computation of finite-difference approximations of the derivatives of the objective function. It is also used in computing the default value of the **FCONV=** option in the **NLOPTIONS** statement.

**GRADIENT**

displays the gradient of the parameter estimates in the “Covariance Parameter Estimates” or the “Parameter Estimates” table.

**HESSIAN | HESS | H**

displays the Hessian matrix of the optimization.

**IC=NONE****IC=PQ****IC=Q**

determines the computation of information criteria in the “Fit Statistics” table. The GLIMMIX procedure computes various information criteria which typically apply a penalty to the (possibly restricted) log likelihood, log pseudo-likelihood, or log quasi-likelihood, that depends on the number of parameters and/or the sample size. If **IC=NONE**, these criteria are suppressed in the “Fit Statistics” table. This is the default for models based on pseudo-likelihoods.

The AIC, AICC, BIC, CAIC, and HQIC fit statistics are various information criteria. AIC and AICC represent Akaike’s information criteria (Akaike 1974) and a small sample bias corrected version thereof (for AICC, see Hurvich and Tsai 1989; Burnham and Anderson 1998). BIC represents Schwarz’ Bayesian criterion (Schwarz 1978). [Table 1](#) gives formulas for the criteria.

**Table 1.** Information Criteria

Criteria	Formula	Reference
AIC	$-2\ell + 2d$	Akaike (1974)
AICC	$-2\ell + 2dn^*/(n^* - d - 1)$	Hurvich and Tsai (1989) Burnham and Anderson (1998)
HQIC	$-2\ell + 2d \log \log n$	Hannan and Quinn (1979)
BIC	$-2\ell + d \log n$	Schwarz (1978)
CAIC	$-2\ell + d(\log n + 1)$	Bozdogan (1987)

Here,  $\ell$  denotes the maximum value of the (possibly restricted) log likelihood, log pseudo-likelihood, or log quasi-likelihood,  $d$  is the dimension of the model, and  $n$ ,  $n^*$  reflect the size of the data.

The IC=PQ option requests that the penalties include the number of fixed effects parameters, when estimation in models with random effects is based on a residual (restricted) likelihood. For `METHOD=MSPL` and `METHOD=MRPL`, IC=Q and IC=PQ produce the same results. IC=Q is the default for linear mixed models with normal errors and the resulting information criteria are identical to the IC option in the MIXED procedure.

The quantities  $d$ ,  $n$ , and  $n^*$  depend on the model and IC= option.

**GLM:** IC=Q and IC=PQ options have no effect on the computation.

- $d$  equals the number of parameters in the optimization whose solutions do not fall on the boundary or are otherwise constrained. The scale parameter is included if it is part of the optimization. If you use the `PARMS` statement to place a hold on a scale parameter, that parameter does not count toward  $d$ .
- $n$  equals the sum of the frequencies ( $f$ ) for maximum likelihood and quasi-likelihood estimation and  $f - \text{rank}(\mathbf{X})$  for restricted maximum likelihood estimation.
- $n^*$  equals  $n$ , unless  $n < d + 2$ , in which case  $n^* = d + 2$ .

**GLMM, IC=Q:**

- $d$  equals the number of effective covariance parameters, that is, covariance parameters whose solution does not fall on the boundary. For estimation of an unrestricted objective function (`METHOD=MMPL`, `METHOD=MSPL`), this value is incremented by  $\text{rank}(\mathbf{X})$ .
- $n$  equals the effective number of subjects as displayed in the “Dimensions” table, unless this value equals 1, in which case  $n$  equals the number of levels of the first G-side `RANDOM` effect specified. If the number of effective subjects equals 1 and there are no G-side random effects,  $n$  is determined as

$$n = \begin{cases} f - \text{rank}(\mathbf{X}) & \text{METHOD=RMPL, METHOD=RSPL} \\ f & \text{otherwise} \end{cases}$$

where  $f$  is the sum of frequencies used.

- $n^*$  equals  $f$  or  $f - \text{rank}(\mathbf{X})$  (for `METHOD=RMPL` and `METHOD=RSPL`), unless this value is less than  $d + 2$ , in which case  $n^* = d + 2$ .

**GLMM, IC=PQ:** For `METHOD=MSPL` and `METHOD=MMPL`, the results are the same as for IC=Q. For `METHOD=RSPL` and `METHOD=RMPL`,  $d$  equals the number of effective covariance parameters plus  $\text{rank}(\mathbf{X})$ , and  $n = n^*$  equals  $f - \text{rank}(\mathbf{X})$ . The formulas for the information criteria thus agree with Verbeke and Molenberghs (2000, Table 6.7, p. 74) and Vonesh and Chinchilli (1997, p. 263).

**INITGLM**

requests that the estimates from a generalized linear model fit (a model without random effects) be used as the starting values for the generalized linear mixed model.

**INITITER**<=*number*>

specifies the maximum number of iterations used when a generalized linear model is fit initially to derive starting values for the fixed effects; see the [INITGLM](#) option. By default, the initial fit involves at most four iteratively reweighted least squares updates. You can change the upper limit of initial iterations with *number*. If the model does not contain random effects, this option has no effect.

**ITDETAILS**

adds parameter estimates and gradients to the “Iteration History” table.

**LIST**

requests that the model program and variable lists be displayed. This is a debugging feature and is not normally needed. When you use programming statements to define your statistical model, this option enables you to examine the complete set of statements submitted for processing. See the “[Programming Statements](#)” section for more details on how to use SAS statements with the GLIMMIX procedure.

**MAXLMMUPDATE**<=*number*>**MAXOPT**<=*number*>

specifies the maximum number of optimizations for doubly iterative estimation methods based on linearizations. After each optimization, a new pseudo-model is constructed through a Taylor series expansion. This step is known as the linear mixed model update. The MAXLMMUPDATE option limits the number of updates and thereby limits the number of optimizations. If this option is not specified, *number* is set equal to the value specified in the [MAXITER](#) option of the [NLOPTIONS](#) statement. If no MAXITER= value is given, *number* defaults to 20.

**METHOD=RSPL****METHOD=MSPL****METHOD=RMPL****METHOD=MMPL**

specifies the estimation method in a generalized linear mixed model (GLMM). The default is METHOD=RSPL.

Estimation methods ending in “PL” are pseudo-likelihood techniques. The first letter of the METHOD= identifier determines whether estimation is based on a residual likelihood (“R”) or a maximum likelihood (“M”). The second letter identifies the expansion locus for the underlying approximation. Pseudo-likelihood methods for generalized linear mixed models can be cast in terms of Taylor series expansions (linearizations) of the GLMM. The expansion locus of the expansion is either the vector of random effects solutions (“S”) or the mean of the random effects (“M”). The expansions are also referred to as the “S”-subject-specific and “M”-arginal expansions. The abbreviation “PL” identifies the method as a pseudo-likelihood technique.

Residual methods account for the fixed effects in the construction of the objective function, which reduces the bias in covariance parameter estimates. Estimation methods involving Taylor series create pseudo data for each optimization. Those data are



transformed to have zero mean in a residual method. While the covariance parameter estimates in a residual method are the maximum likelihood estimates for the transformed problem, the fixed effects estimates are (estimated) generalized least squares estimates. In a likelihood method that is not residual based, both the covariance parameters and the fixed effects estimates are maximum likelihood estimates, but the former are known to have greater bias. In some problems, residual likelihood estimates of covariance parameters are unbiased.

For more information about linearization methods for generalized linear mixed models, see the [“Pseudo-Likelihood Estimation Based on Linearization”](#) section (beginning on page 120).

If the model does not contain random effects or only a single overdispersion component, the model belongs to the family of (overdispersed) generalized linear models if the distribution is known or the quasi-likelihood models for independent data if the distribution is not known. The GLIMMIX procedure then estimates model parameters by the following techniques:

- normally distributed data: residual maximum likelihood
- nonnormal data: maximum likelihood
- data with unknown distribution: quasi-likelihood

The METHOD= specification then has only an effect with respect to the divisor used in estimating the overdispersion component. With a residual method, the divisor is  $f - k$ , where  $f$  denotes the sum of the frequencies and  $k$  is the rank of  $\mathbf{X}$ . Otherwise, the divisor is  $f$ .

**NAMELEN**<=number>

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

**NOCLPRINT**<=number>

suppresses the display of the “Class Level Information” table, if you do not specify *number*. If you specify *number*, only levels with totals that are less than *number* are listed in the table.

**NOFIT**

suppresses fitting of the model. When the NOFIT option is in effect, PROC GLIMMIX produces the “Model Information,” “Class Level Information,” “Number of Observations,” and “Dimensions” tables. These can be helpful to gauge the computational effort required to fit the model. For example, the “Dimensions” table informs you as to whether the GLIMMIX procedure processes the data by subjects, which is typically more computationally efficient than processing the data as a single subject. See the [“Processing by Subjects”](#) section (beginning on page 129) for more information.

If you request a radial smooth with knot selection by  $k$ - $d$  tree methods, PROC GLIMMIX also computes the knot locations of the smoother. You can then examine

the knots without fitting the model. This enables you to try out different knot construction methods and bucket sizes. See the [KNOTMETHOD=KDTREE](#) option (and its suboptions) of the [RANDOM](#) statement.

**NOITPRINT**

suppresses the display of the “Iteration History” table.

**NOPROFILE**

includes the scale parameter  $\phi$  into the optimization for models that have such a parameter (see [Table 10](#) on page 110). By default, the GLIMMIX procedure profiles scale parameters from the optimization in mixed models. In generalized linear models, scale parameters are not profiled.

**NOREML**

determines the denominator for the computation of the scale parameter in a GLM for normal data and for overdispersion parameters. By default, the GLIMMIX procedure computes the scale parameter for the normal distribution as

$$\hat{\phi} = \sum_{i=1}^n \frac{f_i (y_i - \hat{y}_i)^2}{f - k}$$

where  $k$  is the rank of  $\mathbf{X}$ ,  $f_i$  is the frequency associated with the  $i$ th observation, and  $f = \sum f_i$ . Similarly, the overdispersion parameter in an overdispersed GLM is estimated by the ratio of the Pearson statistic and  $(f - k)$ . If the NOREML option is in effect, the denominators are replaced by  $f$ , the sum of the frequencies. In a GLM for normal data, this yields the maximum likelihood estimate of the error variance.

In GLMM models fit by pseudo-likelihood methods, the NOREML option changes the estimation method to the nonresidual form. See the [METHOD=](#) option for the distinction between residual and nonresidual estimation methods.

**ODDSRATIO | OR**

requests that odds ratios be added to the output when applicable. Odds ratios and their confidence limits are only reported for models with logit, cumulative logit, or generalized logit link. Specifying the ODDSRATIO option in the PROC GLIMMIX statement has the same effect as specifying the ODDSRATIO option in the [MODEL](#) statement and in all [LSMEANS](#) statements. Note that the ODDSRATIO option in the [MODEL](#) statement has several suboptions that enable you to construct customized odds ratios. These suboptions are only available through the [MODEL](#) statement. For details on the interpretation and computation of odds and odds ratios with the GLIMMIX procedure, see the “[Odds and Odds Ratio Estimation](#)” section (beginning on page 138).

**ORDER=DATA****ORDER=FORMATTED****ORDER=FREQ****ORDER=INTERNAL**

specifies the sorting order for the levels of all [CLASS](#) variables. This ordering determines which parameters in the model correspond to each level in the data, so the

ORDER= option may be useful when you use **CONTRAST** or **ESTIMATE** statements.

When the default ORDER=FORMATTED 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 this format explicitly for the **CLASS** variables.

The following table shows how PROC GLIMMIX interprets values of the ORDER= option.

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

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

#### PCONV=*r*

specifies the parameter estimate convergence criterion for doubly iterative estimation methods. The GLIMMIX procedure applies this criterion to fixed effects estimates and covariance parameter estimates. Suppose  $\widehat{\psi}_i^{(u)}$  denotes the estimate of the *i*th parameter at the *u*th optimization. The procedure terminates the doubly iterative process if the largest value

$$2 \times \frac{|\widehat{\psi}_i^{(u)} - \widehat{\psi}_i^{(u-1)}|}{|\widehat{\psi}_i^{(u)}| + |\widehat{\psi}_i^{(u-1)}|}$$

is less than *r*. To check an absolute convergence criteria as well, you can set the **ABSPCONV=** option of the PROC GLIMMIX statement. The default value for *r* is 1E8 times the machine epsilon, a product that equals about 1E−8 on most machines.

Note that this convergence criterion does not affect the convergence criteria applied within any individual optimization. In order to change the convergence behavior within an optimization, you can use the **ABSCONV=**, **ABSFCONV=**, **ABSGCONV=**, **ABSXCONV=**, **FCONV=**, or **GCONV=** options of the **NLOPTIONS** statement.

**Experimental**

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

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

requests that the GLIMMIX procedure produces statistical graphics via the Output Delivery System, provided that the ODS GRAPHICS statement has been specified. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS” (*SAS/STAT User’s Guide*). For specific information about the graphics available in the GLIMMIX procedure, see the “[ODS Graphics](#)” section on page 168. For examples of the basic statistical graphics produced by the GLIMMIX procedure and aspects of their computation and interpretation, see the section “[Statistical Graphics for LS-Mean Comparisons](#)” on page 169 in this chapter. You can also request statistical graphics for least-squares means through the **PLOTS** option in the **LSMEANS** statement, which gives you more control over the display compared to the **PLOTS** option in the PROC GLIMMIX statement.

The *global-plot-options* apply to all plots generated by the GLIMMIX procedure, unless it is altered by a *specific-plot-option*. Currently, the global plot options supported by the GLIMMIX procedure are

- |        |   |
|--------|---|
| OBSNO  | uses the data set observation number to identify observations in tool tips, provided that the observation number can be determined. Otherwise, the number displayed in tool tips is the index of the observation as it is used in the analysis within the BY group. |
| UNPACK | breaks a graphic that is otherwise paneled into individual component plots.   |

The following listing describes the specific plots and their options.

- |     |  |
|-----|--|
| ALL | requests that all default plots are produced. The default for each residual plot is based on using the BLUPs of random effects and representing the residual on the linearized (linked) scale. Plots of least-squares means differences are created only for <b>LSMEANS</b> statements without options that would contradict such a display. |
|-----|--|

- |                 |   |
|-----------------|---|
| ANOMPLOT   ANOM | requests an analysis of means display in which least-squares means are compared against an average least-squares mean (Ott, 1967; Nelson, 1982, 1991, 1993). See the <b>DIFF=</b> option of the <b>LSMEANS</b> statement for the computation of this average. Least-squares mean ANOM plots are only produced for those model effects that are listed in <b>LSMEANS</b> statements that have options that do not contradict with the display. For example, if you request ANOM plots with the <b>PLOTS=</b> option in the PROC GLIMMIX statement, the <b>LSMEANS</b> statements |
|-----------------|---|

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

produce analysis of mean plots for effects A and C. The DIFF option in the second LSMEANS statement implies all pairwise differences.

When differences against the average LS-mean are adjusted for multiplicity with the ADJUST=NELSON option of the LSMEANS statement, the ANOMPLOT display is adjusted accordingly.

CONTROLPLOT | CONTROL requests a display in which least-squares means are visually compared against a reference level. Least-squares mean control plots are only produced for those model effects that are listed in LSMEANS statements that have options that do not contradict with the display. For example, the statements

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

produce control plots for effects A and C if you specify PLOTS=CONTROL in the PROC GLIMMIX statement. The DIFF option in the second LSMEANS statement implies all pairwise differences.

When differences against a control level are adjusted for multiplicity with the ADJUST= option of the LSMEANS statement, the control plot display is adjusted accordingly.

DIFFPLOT | DIFF | DIFFOGRAM <(ABS | NOABS)> requests a display of all pairwise least-squares mean differences and their significance. When constructed from arithmetic means, the display is also known as a “mean-mean scatter plot” (Hsu 1996; 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. The ABS and NOABS suboptions determine the positioning of the line segments in the plot. When the ABS option is in effect, and this is the default, all line segments are shown on the same side of the reference line. The NOABS option separates comparisons according to the sign of the difference.

If you specify the ADJUST= option in the LSMEANS statement, the lengths of the line segments are adjusted for multiplicity.

LS-mean difference plots are only produced for those model effects that are listed in LSMEANS statements that have options that do not conflict with the display. For example, the statements

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

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. The **DIFF=** type in the first statement contradicts a display of all pairwise differences. Diffograms are produced for the **B** and **C** effects if you specify **PLOTS=DIFF** in the PROC GLIMMIX statement.

**MEANPLOT** *<(meanplot-options)>* requests a display of the least-squares means of effects specified in **LSMEANS** statements. The following *meanplot-options* affect the display. Upper and lower confidence limits are plotted when the **CL** option is used. When the **CLBAND** option is in effect, confidence limits are shown as bands, and the means are connected. By default, least-squares means are not joined by lines. You can achieve that effect with the **JOIN** or **CONNECT** options. Least-squares means are displayed in the same order as they appear in the “Least Squares Means” table. You can change that order for plotting with the **ASCENDING** and **DESCENDING** options. The **ILINK** option requests that results be displayed on the inverse linked scale.

Note that there is also a **MEANPLOT** suboption of the **PLOTS=** option in the **LSMEANS** statement. In addition to the *meanplot-options* just described, you can also specify classification effects that give you more control over the display of interaction means through the **PLOTBY=** and **SLICEBY=** options. To display interaction means you typically want to use the **MEANPLOT** option in the **LSMEANS** statement. For example,

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

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.

**NONE** requests that no plots are produced.

**RESIDUALPANEL** *<(residualplot-options)>* requests a paneled display constructed from raw residuals. The panel consists of a plot of the residuals against the linear predictor or predicted mean, a histogram with normal density overlaid, a *Q-Q* plot, and a box plot of the residuals. The *residualplot-options* enable you to specify which type of residual is being graphed. These are further discussed below.

**STUDENTPANEL** *<(residualplot-options)>* requests a paneled display constructed from studentized residuals. The same panel organization is applied as for the **RESIDUALPANEL** plot type.

**PEARSONPANEL** *<(residualplot-options)>* requests a paneled display constructed from Pearson residuals. The same panel organization is applied as for the **RESIDUALPANEL** plot type.

The *residualplot-options* apply to the RESIDUALPANEL, STUDENTPANEL, and PEARSONPANEL displays. The primary function of these options is to control which type of a residual to display. The four types correspond to the same *keyword-options* as for output statistics in the OUTPUT statement. The *residualplot-options* take on the following values.

TYPE=	<i>BLUP</i>	uses the predictors of the random effects in computing the residual.
	<i>ILINK</i>	computes the residual on the scale of the data.
	<i>NOBLUP</i>	does not use the predictors of the random effects in computing the residual.
	<i>NOILINK</i>	computes the residual on the linked scale
UNPACK		displays the component plots of a paneled residual display separately.

If you specify the UNPACK option for one of the residual panels it applies to all panels of this type. For example, the statements

```
proc glimmix plots=(residualpanel(type=noblup unpack)
                   residualpanel(type=blup)
                   studentpanel(type=noblup) );
```

produce two unpacked RESIDUALPANELs and one packed panel of studentized residuals. The TYPE= specification is specific to each *plot-request*.

The default is to compute residuals on the linearized scale using BLUPs if the model contains G-side random effects. Not all combinations of BLUP/NOBLUP and ILINK/NOILINK suboptions are possible for all residual types and models. For details, see the description of output statistics for the [OUTPUT](#) statement. Pearson residuals are always displayed against the linear predictor, all other residuals are graphed versus the linear predictor if the NOILINK suboption is in effect (default), and against the corresponding prediction on the mean scale if the ILINK option is in effect. See [Table 8](#) (page 87) for a definition of the residual quantities and exclusions.

## PROFILE

requests that scale parameters are profiled from the optimization, if possible. This is the default for generalized linear mixed models. In generalized linear models with normally distributed data, you can use the PROFILE option to request profiling of the residual variance.

## SCOREMOD

requests that the Hessian matrix in GLMMs be based on a modified scoring algorithm, provided that PROC GLIMMIX is in scoring mode when the Hessian is evaluated. The procedure is in scoring mode during iteration, if the optimization technique requires second derivatives, the [SCORING=\*n\*](#) option is specified, and the iteration count has not exceeded *n*. The procedure also computes the expected (scoring) Hessian matrix when you use the [EXPHESIAN](#) option of the PROC GLIMMIX statement.

The SCOREMOD option has no effect if the SCORING= or EXPHESSIAN options are not specified. The nature of the SCOREMOD modification to the expected Hessian computation is shown in Table 11 on page 124 in the section “Pseudo-Likelihood Estimation Based on Linearization” on page 120. The modification can improve the convergence behavior of the GLMM compared to standard Fisher scoring and can provide a better approximation of the variability of the covariance parameters. For more details, see the “Estimated Precision of Estimates” section.

**SCORING=number**

requests that Fisher scoring be used in association with the estimation method up to iteration *number*. By default, no scoring is applied. When you use the SCORING= option and PROC GLIMMIX converges without stopping the scoring algorithm, the procedure uses the expected Hessian matrix to compute approximate standard errors for the covariance parameters instead of the observed Hessian. If necessary, the standard errors of the covariance parameters as well as the output from the ASYCOV and ASYCORR options are adjusted.

If scoring stopped prior to convergence and you want to use the expected Hessian matrix in the computation of standard errors, use the EXPHESSIAN option of the PROC GLIMMIX statement.

Scoring is not possible in models for nominal data. It is also not possible for GLMs with unknown distribution or those outside the exponential family. If you perform quasi-likelihood estimation, the GLIMMIX procedure is always in scoring mode and the SCORING= option has no effect. See the section “Quasi-Likelihood for Independent Data” on page 117 for a description of the types of models where GLIMMIX applies quasi-likelihood estimation.

The SCORING= option has no effect for optimization methods that do not involve second derivatives. See the TECHNIQUE= option of the NLOPTIONS statement and the section “Choosing an Optimization Algorithm” on page 153 for details on first- and second-order algorithms.

**SINGCHOL=number**

tunes the singularity criterion in Cholesky decompositions. The default is the square root of the SINGULAR criterion.

**SINGULAR=number**

tunes the general singularity criterion applied by the GLIMMIX procedure in divisions and inversions. The default is 1E4 times the machine epsilon; this product is approximately  $1E - 12$  on most computers.

**STARTGLM**

is an alias of the INITGLM option.



---

## BY Statement

**BY** *variables* ;

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

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

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

Since sorting the data changes the order in which PROC GLIMMIX reads observations, the sorting order for the levels of the CLASS variable may be affected if you have specified ORDER=DATA in the PROC GLIMMIX statement. This, in turn, affects specifications in the CONTRAST, ESTIMATE, or LSMESTIMATE statements.

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

---

## CLASS Statement

**CLASS** *variables* ;

The CLASS statement names the classification variables to be used in the analysis. If the CLASS statement is used, it must appear before the MODEL statement.

Classification variables can be either character or numeric. By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. In releases prior to SAS 9, class levels were determined using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*. You can adjust the order of CLASS variable levels with the ORDER= option in the PROC GLIMMIX statement.

You can specify the following option in the CLASS statement after a slash (/).

**TRUNCATE**

specifies that class levels should be determined using no more than the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to SAS 9.

---

**CONTRAST Statement**

```

CONTRAST 'label' contrast-specification
           <, contrast-specification > <, ... >
           < / options > ;

```

The CONTRAST statement provides a mechanism for obtaining custom hypothesis tests. It is patterned after the CONTRAST statement in PROC MIXED and enables you to select an appropriate inference space (McLean, Sanders, and Stroup 1991).

You can test the hypothesis  $\mathbf{L}'\phi = 0$ , where  $\mathbf{L}' = [\mathbf{K}' \mathbf{M}']$  and  $\phi' = [\beta' \gamma']$ , in several inference spaces. The inference space corresponds to the choice of  $\mathbf{M}$ . When  $\mathbf{M} = \mathbf{0}$ , your inferences apply to the entire population from which the random effects are sampled; this is known as the *broad* inference space. When all elements of  $\mathbf{M}$  are nonzero, your inferences apply only to the observed levels of the random effects. This is known as the *narrow* inference space, and you can also choose it by specifying all of the random effects as fixed. The GLM procedure uses the narrow inference space. Finally, by zeroing portions of  $\mathbf{M}$  corresponding to selected main effects and interactions, you can choose *intermediate* inference spaces. The broad inference space is usually the most appropriate; it is used when you do not specify random effects in the CONTRAST statement.

In the CONTRAST statement,

<i>label</i>	identifies the contrast in the table. A label is required for every contrast specified. Labels can be up to 20 characters and must be enclosed in single quotes.
<i>contrast-specification</i>	identifies the fixed-effects and random-effects and their coefficients from which the $\mathbf{L}$ matrix is formed. The syntax representation of a <i>contrast-specification</i> is $\langle \textit{fixed-effect values} \dots \rangle \langle   \textit{random-effect values} \dots \rangle$
<i>fixed-effect</i>	identifies an effect that appears in the <b>MODEL</b> 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 <b>MODEL</b> statement.
<i>random-effect</i>	identifies an effect that appears in the <b>RANDOM</b> statement. The first random effect must follow a vertical bar ( ); however, random effects do not have to be specified.
<i>values</i>	are constants that are elements of the $\mathbf{L}$ matrix associated with the fixed and random effects.

The rows of  $\mathbf{L}'$  are specified in order and are separated by commas. The rows of the  $\mathbf{K}'$  component of  $\mathbf{L}'$  are specified on the left side of the vertical bars (|). These rows

test the fixed effects and are, therefore, checked for estimability. The rows of the  $M'$  component of  $L'$  are specified on the right side of the vertical bars. They test the random effects, and no estimability checking is necessary.

If PROC GLIMMIX finds the fixed-effects portion of the specified contrast to be nonestimable (see the `SINGULAR=` option on page 36), then it displays missing values for the test statistics.

If the elements of  $L$  are not specified for an effect that contains a specified effect, then the elements of the unspecified effect are automatically “filled in” over the levels of the higher-order effect. This feature is designed to preserve estimability for cases when there are complex higher-order effects. The coefficients for the higher-order effect are determined by equitably distributing the coefficients of the lower-level effect as in the construction of least-squares means. In addition, if the intercept is specified, it is distributed over all classification effects that are not contained by any other specified effect. If an effect is not specified and does not contain any specified effects, then all of its coefficients in  $L$  are set to 0. You can override this behavior by specifying coefficients for the higher-order effect.

If too many values are specified for an effect, the extra ones are ignored; if too few are specified, the remaining ones are set to 0. If no random effects are specified, the vertical bar can be omitted; otherwise, it must be present. If a SUBJECT effect is used in the `RANDOM` statement, then the coefficients specified for the effects in the `RANDOM` statement are equitably distributed across the levels of the SUBJECT effect. You can use the `E` option to see exactly what  $L$  matrix is used.

PROC GLIMMIX handles missing level combinations of classification variables similarly to PROC GLM and PROC MIXED. These procedures delete fixed-effects parameters corresponding to missing levels in order to preserve estimability. However, PROC MIXED and PROC GLIMMIX do not delete missing level combinations for random-effects parameters, because linear combinations of the random-effects parameters are always estimable. These conventions can affect the way you specify your CONTRAST coefficients.

The CONTRAST statement computes the statistic

$$F = \frac{\begin{bmatrix} \hat{\beta} \\ \hat{\gamma} \end{bmatrix}' \mathbf{L}(\mathbf{L}'\hat{\mathbf{C}}\mathbf{L})^{-1}\mathbf{L}' \begin{bmatrix} \hat{\beta} \\ \hat{\gamma} \end{bmatrix}}{\text{rank}(\mathbf{L})}$$

and approximates its distribution with an  $F$  distribution unless `DDFM=NONE`. If you select `DDFM=NONE` as the degrees-of-freedom method in the `MODEL` statement, and if you do not assign degrees of freedom to the contrast with the `DF=` option, PROC GLIMMIX computes the test statistic  $\text{rank}(\mathbf{L})F$  and approximates its distribution with a chi-square distribution. In the expression for  $F$ ,  $\hat{\mathbf{C}}$  is an estimate of  $\text{var}[\hat{\beta}, \hat{\gamma} - \gamma]$ .

The numerator degrees of freedom in the  $F$  approximation and the degrees of freedom in the chi-square approximation are equal to  $\text{rank}(\mathbf{L})$ . The denominator degrees of freedom is taken from the “Tests of Fixed Effects” table and corresponds to the

final effect you list in the CONTRAST statement. You can change the denominator degrees of freedom by using the `DF=` option.

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

### BYCATEGORY

#### BYCAT

requests that in models for nominal data (generalized logit models) the contrasts are not combined across response categories but reported separately for each category. For example, assume that the response variable `Style` is multinomial with three (un-ordered) categories. The GLIMMIX statements

```
proc glimmix data=school;
  class School Program;
  model Style(order=data) = School Program / s ddfm=none
                                dist=multinomial link=glogit;

  freq Count;
  contrast 'School 1 vs. 2' school 1 -1;
  contrast 'School 1 vs. 2' school 1 -1 / bycat;
run;
```

fit a generalized logit model relating the preferred style of instruction to school and educational program effects. The first contrast compares school effects in all categories. This is a two-degree-of-freedom contrast because there are two nonredundant categories. The second CONTRAST statement produces two single-degree-of-freedom contrasts, one for each non-reference `Style` category.

The BYCATEGORY option has no effect unless your model is a generalized (mixed) logit model.

#### CHISQ

requests that chi-square tests be performed for all contrasts in addition to any  $F$  tests. A chi-square statistic equals its corresponding  $F$  statistic times the numerator degrees of freedom, and this same degrees of freedom is used to compute the  $p$ -value for the chi-square test. This  $p$ -value will always be less than that for the  $F$  test, as it effectively corresponds to an  $F$  test with infinite denominator degrees of freedom.

#### DF=*number*

specifies the denominator degrees of freedom for the  $F$  test. For the degrees of freedom methods `DDFM=BETWITHIN`, `DDFM=CONTAIN`, and `DDFM=RESIDUAL`, the default is the denominator degrees of freedom taken from the “Tests of Fixed Effects” table and corresponds to the final effect you list in the CONTRAST statement. For `DDFM=NONE`, infinite denominator degrees of freedom are assumed by default and for `DDFM=SATTERTHWAITE` and `DDFM=KENWARDROGER`, the denominator degrees of freedom are computed separately for each contrast.

#### E

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

#### SINGULAR=*number*

tunes the estimability checking. If  $\mathbf{v}$  is a vector, define  $ABS(\mathbf{v})$  to be the absolute

value of the element of  $\mathbf{v}$  with the largest absolute value. 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  $\mathbf{K}$  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.

---

## ESTIMATE Statement

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

The ESTIMATE statement provides a mechanism for obtaining custom hypothesis tests. As in the CONTRAST statement, the basic element of the ESTIMATE statement is the *contrast-specification*, which consists of MODEL and G-side RANDOM effects and their coefficients. Specifically, a *contrast-specification* takes the form

```
< fixed-effect values ... > < | random-effect values ... >
```

Based on the *contrast-specifications* in your ESTIMATE statement, PROC GLIMMIX constructs the matrix  $\mathbf{L}' = [\mathbf{K}' \ \mathbf{M}']$ , as in the CONTRAST statement, where  $\mathbf{K}$  is associated with the fixed-effects and  $\mathbf{M}$  is associated with the G-side random effects.

PROC GLIMMIX then produces for each row  $\mathbf{l}$  of  $\mathbf{L}'$  an approximate  $t$  test of the hypothesis  $H: \mathbf{l}\phi = 0$ , where  $\phi = [\beta' \ \gamma']'$ . You can also obtain multiplicity adjusted  $p$ -values and confidence limits for multi-row estimates with the ADJUST= option. The output from multiple ESTIMATE statements is organized as follows. Results from unadjusted estimates are reported first in a single table, followed by separate tables for each of the adjusted estimates. Results from all ESTIMATE statement are combined in the “Estimates” ODS table.

Note that multi-row estimates are permitted. Unlike the CONTRAST statement, you need to specify a *'label'* for every row of the multi-row estimate, since PROC GLIMMIX produces one test per row.

PROC GLIMMIX selects the degrees of freedom to match those displayed in the “Type III Tests of Fixed Effects” table for the final effect you list in the ESTIMATE statement. You can modify the degrees of freedom using the DF= option. If you select DDFM=NONE and do not modify the degrees of freedom using the DF= option, PROC GLIMMIX uses infinite degrees of freedom, essentially computing approximate  $z$  tests. If PROC GLIMMIX finds the fixed-effects portion of the specified estimate to be nonestimable, then it displays “Non-est” for the estimate entry.

### 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 listed in the ESTIMATE statement from 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. This can be the case, for example, when the DDFM=SATTERTHWAITE or DDFM=KENWARDROGER degrees-of-freedom methods are in effect.

**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= options are in effect. For a description of the adjustments, see Chapter 32, “The GLM Procedure,” and Chapter 48, “The MULTTEST Procedure,” in the *SAS/STAT User’s Guide* and the documentation for the ADJUST= option of the LSMEANS statement. The ADJUST option is ignored for generalized logit models.

If the STEPDOWN option is in effect and you choose ADJUST=BON or ADJUST=SIMULATE, 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  $1 - \text{number}$ . The value of *number* must be between 0 and 1; the default is 0.05. If DDFM=NONE and you do not specify degrees of freedom with the DF= option, PROC GLIMMIX uses infinite degrees of freedom, essentially computing a  $z$  interval.

**BYCATEGORY****BYCAT**

requests that in models for nominal data (generalized logit models) estimates are reported separately for each category. In contrast to the BYCATEGORY option in the CONTRAST statement, an ESTIMATE statement in a generalized logit model does not distribute coefficients by response category because ESTIMATES always correspond to single rows of the **L** matrix.

For example, assume that the response variable **Style** is multinomial with three (un-ordered) categories. The GLIMMIX statements

```
proc glimmix data=school;
  class School Program;
  model Style(order=data) = School Program / s ddfm=none
                                dist=multinomial link=logit;

  freq Count;
  estimate 'School 1 vs. 2' school 1 -1 / bycat;
  estimate 'School 1 vs. 2' school 1 -1;
run;
```

fit a generalized logit model relating the preferred style of instruction to school and educational program effects. The first ESTIMATE statement compares school effects separately for each nonredundant category. The second ESTIMATE statement compares the school effects for the first non-reference category.

The BYCATEGORY option has no effect unless your model is a generalized (mixed) logit model.

### CL

requests that  $t$  type confidence limits be constructed. If `DDFM=NONE` and you do not specify degrees of freedom with the `DF=` option, PROC GLIMMIX uses infinite degrees of freedom, essentially computing a  $z$  interval. The confidence level is 0.95 by default. These intervals are adjusted for multiplicity when you specify the `ADJUST=` option.

### 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 “Tests of Fixed Effects” table and corresponds to the final effect you list in the ESTIMATE statement.

### 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 implied by the *value-list*. For example, the statement

```
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;
```

divides the coefficients in the first row by 8, and the coefficients in the third and fourth row by 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. See the “[Odds and Odds Ratio Estimation](#)” section (beginning on page 138) for important details concerning the computation and interpretation of odds and odds ratio results with the GLIMMIX procedure. If you specify the `CL` or the `ALPHA=` option, the (adjusted) confidence bounds are also exponentiated.

**ILINK**

requests that the estimate and its standard error are also reported on the scale of the mean (the inverse linked scale). PROC GLIMMIX computes the value on the mean scale by applying the inverse link to the estimate. The interpretation of this quantity depends on the *fixed-effect values* and *random-effect values* specified in your ESTIMATE statement and on the link function. In a model for binary data with logit link, for example,

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

computes

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

where  $\alpha_1$  and  $\alpha_2$  are the fixed-effects solutions associated with the first two levels of the classification effect **A**. This quantity is not the difference of the probabilities 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 GLIMMIX 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.

**LOWER****LOWERTAILED**

requests that the *p*-value for the *t* test be based only on values 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 the **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, since this adjustment is based on only the right tail of the F distribution.

**SINGULAR=number**

tunes the estimability checking as documented for the **CONTRAST** statement.

Experimental

**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 will never be declared significant unless all smaller  $p$ -values are also declared significant. Note that the STEPDOWN adjustment is available only for

- ADJUST=BON, where it corresponds to the methods of Holm (1979) and Schaffer's "Method 2" (1986), and
- ADJUST=SIMULATE, where it corresponds to the method of Westfall (1997).

If the degrees-of-freedom method is `DDFM==KENWARDROGER` or `DDFM=SATTERTHWAITE`, then stepdown adjusted  $p$ -values are only produced if the `ADJDFE=ROW` option is in effect.

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

You can specify the following options in parentheses after the STEPDOWN option.

`MAXTIME = n` specifies the time (in seconds) to spend 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 ten groups. In such cases, try using `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 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 should 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` If you specify `TYPE=LOGICAL`, the step-down adjustments are computed 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 ten 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.

### UPPER UPPERTAILED

requests that the  $p$ -value for the  $t$ -test be based only on values 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 the `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, since this adjustment is based on only the right tail of the F distribution.

---

## FREQ Statement

**FREQ** *variable* ;

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

The analysis produced using a `FREQ` statement reflects the expanded number of observations.

---

## ID Statement

**ID** *variables* ;

The `ID` statement specifies which quantities to include in the `OUT=` data set from the `OUTPUT` statement in addition to any statistics requested in the `OUTPUT` statement. If no `ID` statement is given, the `GLIMMIX` procedure includes all variables from the input data set in the `OUT=` data set. Otherwise, only the variables you list in the `ID` statement are included. Specifying an `ID` statement with no variables prevents any variables from being included in these data sets. Automatic variables such as `_LINF_`, `_MU_`, `_VARIANCE_`, etc. are not transferred to the `OUT=` data set, unless explicitly listed in the `ID` statement.

The `ID` statement can be used to transfer computed quantities that depend on the model to an output data set. In the following example, two sets of Hessian weights are

computed in a Gamma regression with a noncanonical link. The covariance matrix for the fixed effects can be constructed as the inverse of  $\mathbf{X}'\mathbf{W}\mathbf{X}$ .  $\mathbf{W}$  is a diagonal matrix of the  $w_{ei}$  or  $w_{oi}$ , depending on whether the expected or observed Hessian matrix is desired, respectively.

```
proc glimmix;
  class group age;
  model cost = group age / s error=gamma link=pow(0.5);
  output out=gmxout pred=pred;
  id _variance_ wei woi;
  vpmu = 2*_mu_;
  if (_mu_ > 1.0e-8) then do;
    gpmu = 0.5 * (_mu_**(-0.5));
    gppmu = -0.25 * (_mu_**(-1.5));
    wei = 1/(_phi*_variance_*gpmu*gpmu);
    woi = wei + (cost-_mu_) *
          (_variance_*gppmu + vpmu*gpmu) /
          (_variance*_variance_*gpmu*gpmu*_phi_);
  end;
run;
```

The variables `_VARIANCE_`, `_MU_`, and other symbols are predefined by PROC GLIMMIX and can be used in programming statements. For rules and restrictions, see the “[Programming Statements](#)” section on page 108.

---

## LSMEANS Statement

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

The LSMEANS statement computes least-squares means (LS-means) of fixed effects. As in the GLM and the MIXED 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. The  $\mathbf{L}$  matrix constructed to compute them is the same as the  $\mathbf{L}$  matrix formed in PROC GLM; however, the standard errors are adjusted for the covariance parameters in the model. Least-squares means computations are currently not supported for multinomial models.

Each LS-mean is computed as  $\mathbf{L}\hat{\boldsymbol{\beta}}$  where  $\mathbf{L}$  is the coefficient matrix 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.

LS-means are constructed on the linked scale, that is, the scale on which the model effects are additive. For example, in a binomial model with logit link, the least-squares means are predicted population margins of the logits.

LS-means can be computed for any effect in the **MODEL** statement 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. As in the **ESTIMATE** statement, the **L** matrix is tested for estimability, and if this test fails, PROC GLIMMIX displays “Non-est” for the LS-means entries.

Assuming the LS-mean is estimable, PROC GLIMMIX constructs an approximate *t* test to test the null hypothesis that the associated population quantity equals zero. By default, the denominator degrees of freedom for this test are the same as those displayed for the effect in the “Tests of Fixed Effects” table. If the **DDFM=SATTERTHWAITE** or **DDFM=KENWARDROGER** option is specified in the **MODEL** statement, PROC GLIMMIX determines degrees of freedom separately for each test, unless the **DDF=** option overrides it for a particular effect. See the **DDFM=** option on page 66 for more information.

You can specify the following options in the LSMEANS 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. When you specify **ADJDFE=ROW**, the denominator degrees of freedom for multiplicity-adjusted results correspond to the degrees of freedom 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. This can be the case, for example, when the **DDFM=SATTERTHWAITE** or **DDFM=KENWARDROGER** degrees-of-freedom methods are in effect.

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 statements

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

fit a heteroscedastic one-way model and perform Dunnett’s T3 method (Dunnett 1980), which is based on the Studentized maximum modulus (**ADJUST=SMM**). If you combine the **ADJDFE=ROW** option with **ADJUST=SIDAK**, the multiplicity adjustment corresponds to the T2 method of Tamhane (1979), while **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, may protect the overall error rate better.

Unless the **ADJUST=** option of the LSMEANS statement is specified, the **ADJDFE=** option has no effect.

**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, PROC GLIMMIX performs all pairwise differences unless you specify ADJUST=DUNNETT, in which case the procedure analyzes all differences with a control level. The ADJUST= option implies the **DIFF option** (see page 48), unless the **SLICEDIFF=** option is specified.

The BON (Bonferroni) and SIDAK adjustments involve correction factors described in Chapter 32, “The GLM Procedure,” and Chapter 48, “The MULTTEST Procedure,” of the *SAS/STAT User’s Guide*; also refer to Westfall and Young (1993) and Westfall et al. (1999). When you specify ADJUST=TUKEY and your data are unbalanced, PROC GLIMMIX 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 GLIMMIX 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. Note that 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, may 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, or to slice differences that you specify with the **SLICEDIFF=** option. See the **DIFF=ANOM** option for more details regarding the analysis of means with the GLIMMIX 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,

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

where  $\hat{q}$  is the simulated  $q$  and  $F$  is the true distribution function of the maximum; refer to 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. The **ACC=** and **EPS=** *simoptions* reset  $\gamma$  and  $\epsilon$ , respectively, the **NSAMP=** *simoption* sets the sample size directly, and the **SEED=** *simoption* specifies an integer 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 generated from reading the time of day from the computer clock. For additional descriptions of these and other simulation options, see the “LSMEANS Statement” section on page 1753 in Chapter 32, “The GLM Procedure” (*SAS/STAT User’s Guide*).

If the **STEPDOWN** option is in effect and you choose **ADJUST=BON** or **ADJUST=SIMULATE**, 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**

enables you to modify the values of the covariates 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 containing 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 invocation of PROC GLIMMIX:

```
proc glimmix;
  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);
run;
```

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  $\bar{x}_1 \cdot \bar{x}_2$ . 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. You should use the **E option** in conjunction with the AT option to check that the modified LS-means coefficients are the ones you desire.

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

#### **BYLEVEL**

requests PROC GLIMMIX to 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, PROC GLIMMIX 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. If **DDFM=NONE**, then PROC GLIMMIX uses infinite degrees of freedom for this test, essentially computing a  $z$  interval. The confidence level is 0.95 by default; this can be changed with the **ALPHA=** option.

#### **CORR**

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

#### **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 of Fixed Effects” table corresponding to the LS-means effect.

**DIFF**<=*difftype*>

**PDIF**F<=*difftype*>

requests that differences of the LS-means be displayed. The optional *difftype* specifies which differences to produce, with possible values ALL, ANOM, CONTROL, CONTROLL, and CONTROLU. The ALL value requests all pairwise differences, and it is the default. The CONTROL *difftype* requests differences with a control, which, by default, is the first level of each of the specified LSMEANS effects.

The ANOM value 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

$$(\mathbf{X}'\mathbf{S}\mathbf{X})^{-1}$$

matrix (see the “Pseudo-Likelihood Estimation Based on Linearization” section). 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; refer to Nelson (1982, 1991, 1993) and Guirguis and Tobias (2004) as well as the documentation for the ADJUST=NELSON option.

To specify which levels of the effects are the controls, list the quoted formatted values in parentheses after the keyword CONTROL. For example, if the effects A, B, and C are class 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 *difftype*. For one-tailed results, use either the CONTROLL or CONTROLU *difftype*. The CONTROLL *difftype* 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. Conversely, the CONTROLU *difftype* 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.

## ILINK

requests that estimates and their standard errors in the “Least Squares Means” table are also 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 GLIMMIX 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.

## 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 based on the corresponding tests, made by comparing the respective *p*-values to the value of the **ALPHA=** option (0.05 by default). However, equal variances are rarely the case for differences between LS-means. If the variances are not all the same, then the comparison lines may 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, PROC GLIMMIX lists the pairs of means that are inferred to be significantly different by the tests but not by the comparison lines. Note, however, that in many cases, even though the variances are unequal, they are near enough so that the comparison lines accurately reflect the test inferences.

## ODDS

requests that in models with logit, cumulative logit, and generalized logit link function the odds of the levels of the *fixed-effects* are reported. If you specify the **CL** or **ALPHA=** options, confidence intervals for the odds are also computed. See the “[Odds and Odds Ratio Estimation](#)” section (beginning on page 138) for further details about computation and interpretation of odds and odds ratios with the GLIMMIX procedure.

## ODDSRATIO | OR

requests that LS-mean differences (**DIFF**, **ADJUST=** options) and simple effect comparisons (**SLICEDIFF** option) are also reported in terms of odds ratios. The **ODDSRATIO** option is ignored unless you are using either the logit, cumulative logit, or generalized logit link function. If you specify the **CL** or **ALPHA=** options, confidence intervals for the odds ratios are also computed. These intervals are adjusted for multiplicity when you specify the **ADJUST=** option. See the “[Odds and Odds Ratio Estimation](#)” section (beginning on page 138) for further details about computation and interpretation of odds and odds ratios with the GLIMMIX procedure.

**OBSMARGINS | OM**

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 input 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 your data.

In computing the observed margins, PROC GLIMMIX 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, PROC GLIMMIX 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 input data set, PROC GLIMMIX 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 check that the modified LS-means coefficients are the ones you desire. 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. See the description of the DIFF option on page 48.

**Experimental**

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

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

requests that least-squares means related graphics are produced via ODS Graphics, provided that the ODS GRAPHICS statement has been specified and that the plot request does not conflict with other options in the LSMEANS statement. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS” (*SAS/STAT User’s Guide*). For specific information about the graphics available in the GLIMMIX procedure and for examples of the basic statistical graphics produced by the GLIMMIX procedure and aspects of their computation and interpretation, see the “**ODS Graphics**” section on page 168 in this chapter.

The *options* for a specific plot request (and their sub-options) of the LSMEANS statement include those for the **PLOTS=** option in the PROC GLIMMIX statement. You can specify classification effects in the MEANPLOT request of the LSMEANS statement to control the display of interaction means with the PLOTBY= and SLICEBY= sub-options; these are not available in the PLOTS= option of the PROC GLIMMIX statement. Options specified in the LSMEANS statement override those on the PLOTS= option in the PROC GLIMMIX statement.

The available options and sub-options are as follows.

**ALL** requests that the default plots corresponding to this LSMEANS statement are 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 against an average least-squares mean. Least-squares mean ANOM plots are only produced for those model effects that are listed in LSMEANS statements that have options that do not contradict with the display. For example, the statements

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

produce analysis of mean plots for effects A and C. The DIFF option in the second LSMEANS statement implies all pairwise differences.

**CONTROLPLOT | CONTROL** requests a display in which least-squares means are visually compared against a reference level. These plots are only produced for statements with options that are compatible with control differences. For example, the statements

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

produce control plots for effects A and C. The DIFF option in the second LSMEANS statement implies all pairwise differences.

**DIFFPLOT | DIFF | DIFFOGRAM <(ABS | NOABS)>** 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; 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. The ABS and NOABS suboptions of the diffogram determine the positioning of the line segments in the plot. When the ABS option is in effect, and this is the default, all line segments are shown on the same side of the reference line. The NOABS option separates comparisons according to the sign of the difference.

Least-squares mean difference plots are only produced for statements with options that are compatible with the display. For example, the statements

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

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. The **DIFF=** type in the first statement is incompatible with a display of all pairwise differences. Diffograms are produced only for the **B** and **C** effects.

**MEANPLOT** <(meanplot-options)> requests displays of the least-squares means. See below for a description of the *meanplot-options*.

**NONE** requests that no plots are produced.

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

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 sliced or 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 sliced or displayed in separate plots.

**ILINK** requests that means (and confidence limits) are 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 statement

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

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.

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 statement

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

requests for each level of **C** one plot of the **A\*B** cell means that are associated with that level of **C**. 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.

**SINGULAR=***number*

tunes the estimability checking as documented in the “**CONTRAST Statement**” section on page 34.

**SLICE=** *fixed-effect*

**SLICE=** (*fixed-effects*)

specifies effects by which to partition interaction LSMEANS effects. This can produce what are known as tests of simple effects (Winer 1971). For example, suppose that **A\*B** is significant, and you want to test the effect of **A** for each level of **B**. The appropriate LSMEANS statement is

```
lsmeans A*B / slice=B;
```

This statement tests for the simple main effects of **A** for **B**, which are calculated by extracting the appropriate rows from the coefficient matrix for the **A\*B** LS-means and using them to form an *F* test.

The **SLICE** option produces *F* tests that test the simultaneous equality of cell means at a fixed level of the slice-effect (Schabenberger, Gregoire, and Kong 2000). You can request differences of the least-squares means while holding one or more factors at a fixed level with the **SLICEDIFF=** option.

The **SLICE** option produces a table titled “Tests of Effect Slices.”

**SLICEDIFF=** *fixed-effect*

**SLICEDIFF=** (*fixed-effects*)

**SIMPLEDIFF=** *fixed-effect*

**SIMPLEDIFF=** (*fixed-effects*)

requests that differences of simple effects be constructed and tested against zero. Whereas the **SLICE** option extracts multiple rows of the coefficient matrix and forms an *F*-test, the **SLICEDIFF** option tests pairwise differences of these rows. This allows you to perform multiple comparisons among the levels of one factor at a fixed level of the other factor. For example, assume that, in a balanced design, factors **A** and **B** have  $a = 4$  and  $b = 3$  levels, respectively. Consider the statements

```
proc glimmix;
  class a b;
```

```

model y = a b a*b;
lsmeans a*b / slice=a;
lsmeans a*b / slicediff=a;
run;

```

The first LSMEANS statement produces four  $F$ -tests, one per level of **A**. The first of these tests is constructed by extracting the three rows corresponding to the first level of **A** from the coefficient matrix for the **A\*B** interaction. Call this matrix  $\mathbf{L}_{a1}$  and its rows  $\mathbf{1}_{a1}^{(1)}$ ,  $\mathbf{1}_{a1}^{(2)}$ , and  $\mathbf{1}_{a1}^{(3)}$ . The SLICE tests the two degree-of-freedom hypothesis

$$H: \begin{cases} \left( \mathbf{1}_{a1}^{(1)} - \mathbf{1}_{a1}^{(2)} \right) \boldsymbol{\beta} = 0 \\ \left( \mathbf{1}_{a1}^{(1)} - \mathbf{1}_{a1}^{(3)} \right) \boldsymbol{\beta} = 0 \end{cases}$$

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 SLICEDIFF option considers the three rows of  $\mathbf{L}_{a1}$  in turn and performs tests of the difference between pairs of rows. How these differences are constructed depends on the SLICEDIFFTYPE= option. By default, all pairwise differences within the subset of **L** are considered; in the example this corresponds to tests of the form

$$\begin{aligned} H: \left( \mathbf{1}_{a1}^{(1)} - \mathbf{1}_{a1}^{(2)} \right) \boldsymbol{\beta} &= 0 \\ H: \left( \mathbf{1}_{a1}^{(1)} - \mathbf{1}_{a1}^{(3)} \right) \boldsymbol{\beta} &= 0 \\ H: \left( \mathbf{1}_{a1}^{(2)} - \mathbf{1}_{a1}^{(3)} \right) \boldsymbol{\beta} &= 0 \end{aligned}$$

In the example, with  $a = 4$  and  $b = 3$ , the second LSMEANS statement 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.

When the ADJUST= option is specified, the GLIMMIX procedure also adjusts the tests for multiplicity. The adjustment is based on the number of comparisons within each level of the SLICEDIFF= effect; see the SLICEDIFFTYPE= option. The Nelson adjustment is not available for slice differences.

**SLICEDIFFTYPE**<=*difftype*>

**SIMPLEDIFFTYPE**=<=*difftype*>

determines the type of simple effect differences produced with the SLICEDIFF= option.

The possible values for the *difftype* are ALL, CONTROL, CONTROLL, and CONTROLU. The *difftype* ALL requests all simple effects differences, and it is the default. The *difftype* CONTROL requests the 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 keyword CONTROL. For example, if the effects **A**, **B**, and **C** are class variables, each having three levels, 1, 2 and 3, the following LSMEANS statement specifies the (1,3) level of **A\*B** as the control.

```
lsmeans A*B / slicediff=(A B)
              slicedifftype=control('1' '3');
```

This LSMEANS statement first produces simple effects differences holding the levels of **A** fixed, and then it produces simple effects differences holding the levels of **B** fixed. In the former case, level '3' of **B** serves as the control level. In the latter case, level '1' of **A** serves as the control.

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. The CONTROLL *diff*type 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. Conversely, the CONTROLU *diff*type 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.

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

*Experimental*

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 will never be declared significant unless all smaller *p*-values are also declared significant. Note that the STEPDOWN adjustment is available only for

- ADJUST=BON, where it corresponds to the methods of Holm (1979) and Schaffer's "Method 2" (1986), and
- ADJUST=SIMULATE, where it corresponds to the method of Westfall (1997).

If the degrees-of-freedom method is DDFM==KENWARDROGER or DDFM=SATTERTHWAITE, then stepdown adjusted *p*-values are only produced if the ADJDFE=ROW option is in effect.

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

You can specify the following options in parentheses after the STEPDOWN option.

MAXTIME = *n* specifies the time (in seconds) to spend 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

ten groups. In such cases, try using TYPE=FREE (the default) or TYPE=LOGICAL(*n*) for small *n*.

**REPORT** specifies that a report on the step-down adjustment should 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** If you specify TYPE=LOGICAL, the step-down adjustments are computed 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 ten 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.

---

## LSMESTIMATE Statement

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

The LSMESTIMATE statement provides a mechanism for obtaining custom hypothesis tests among the least-squares means. In contrast to the hypotheses tested with the **ESTIMATE** or **CONTRAST** statements, the LSMESTIMATE statement allows you to form linear combinations of the least-squares means, rather than linear combination of fixed-effects parameter estimates and/or random effects BLUPs. Multiple-row sets of coefficients are permitted.

The computation of an LSMESTIMATE involves two coefficient matrices. Suppose that the *fixed-effect* has  $n_l$  levels. 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

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

PROC GLIMMIX produces a *t* test for each row of coefficients specified in the LSMESTIMATE statement. You can adjust *p*-values and confidence intervals for



multiplicity with the **ADJUST=** option. You can obtain an  $F$  test of single-row or multi-row LSMESTIMATEs with the **FTEST** option.

Note that in contrast to a multi-row estimate in the **ESTIMATE** statement, you specify only a single fixed-effect in the LSMESTIMATE statement. The row labels are optional and follow the effects specification. For example, the statements

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

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.

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 allows you to assign a separate divisor to each row of the LSMESTIMATE. You can also assign divisor values through the **DIVISOR=** option. See the documentation for this option on the interaction between the two ways of specifying divisors.

Many options of the LSMESTIMATE statement affect the computation of least-squares means. For example, the **AT=**, **BYLEVEL**, and **OM** options. See the documentation for the **LSMEANS** statement for details.

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. This can be the case, for example, when **DDFM=SATTERTHWAITE** or **DDFM=KENWARDROGER** are specified in the **MODEL** statement.

#### **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=** options are in effect. For a description of the adjustments, see Chapter 32, “The GLM Procedure,” and Chapter 48, “The MULTTEST Procedure,” of the *SAS/STAT User’s Guide* as well as the documentation for the **ADJUST=** option of the **LSMEANS** statement.

Note that 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 and you choose **ADJUST=BON** or **ADJUST=SIMULATE**, 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**

enables you to modify the values of the covariates used in computing LS-means. See the **AT=** option of the **LSMEANS** statement for details.

**BYLEVEL**

requests PROC GLIMMIX to 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, PROC GLIMMIX uses weighted margins to construct the LS-means coefficients.

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

**CHISQ**

requests that chi-square tests be performed in addition to  $F$  tests, when you request an  $F$  test with the **FTEST** option.

**CL**

requests that  $t$  type confidence limits be constructed for each of the LS-means. If **DDFM=NONE**, then PROC GLIMMIX uses infinite degrees of freedom for this test,

essentially computing a  $z$  interval. The confidence level is 0.95 by default; this can be changed with the `ALPHA=` option.

**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  $t$  test and confidence limits. The default is the denominator degrees of freedom taken from the “Type III Tests of Fixed Effects” table corresponding to the LS-means effect.

**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 statement

```
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;
```

divides the coefficients in the first row by 8, and the coefficients in the third and fourth row by 3. Coefficients in the second row are not altered.

**E**

requests that the **L** coefficients of the estimable function are 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 are 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. See the “[Odds and Odds Ratio Estimation](#)” section

(beginning on page 138) for important details concerning the computation and interpretation of odds and odds ratio results with the GLIMMIX procedure. If you specify the `CL` or the `ALPHA=` option, the (adjusted) confidence limits for the estimate are also exponentiated.

**FTEST** <(LABEL='label')>

produces an  $F$  test that jointly tests the rows of the LSMESTIMATE against zero. You can specify the optional label to identify the results from that test in the “LSMFtest” table.

**ILINK**

requests that the estimate and its standard error are also reported on the scale of the mean (the inverse linked scale). PROC GLIMMIX computes the value on the mean scale by applying the inverse link to the estimate. The interpretation of this quantity depends on the coefficients specified in your LSMESTIMATE statement and the link function. In a model for binary data with logit link, for example,

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

computes

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

where  $\tau_1$  and  $\tau_2$  are the least-squares means associated with the first two levels of the classification effect **A**. 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 the `ALPHA=` option, the GLIMMIX 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.

**LOWER**

**LOWERTAILED**

requests that the  $p$ -value for the  $t$  test be based only on values 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 the `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, since this adjustment is based on only the right tail of the  $F$  distribution.

**OBSMARGINS | OM**

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 input data set. See the **OBSMARGINS** option of the **LSMEANS** statement for further details.

**SINGULAR=number**

tunes the estimability checking as documented in the “**CONTRAST Statement**” section on page 34.

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

requests that multiplicity adjustments for the  $p$ -values of LS-mean 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 will never be declared significant unless all smaller  $p$ -values are also declared significant. Note that the STEPDOWN adjustment is available only for

- **ADJUST=BON**, where it corresponds to the methods of Holm (1979) and Schaffer’s “Method 2” (1986), and
- **ADJUST=SIMULATE**, where it corresponds to the method of Westfall (1997).

If the degrees-of-freedom method is **DDFM==KENWARDROGER** or **DDFM=SATTERTHWAITE**, then stepdown adjusted  $p$ -values are only produced if the **ADJDFE=ROW** option is in effect.

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

You can specify the following options in parentheses after the STEPDOWN option.

**MAXTIME =  $n$**  specifies the time (in seconds) to spend 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 ten groups. In such cases, try using **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 should be displayed, including a listing of the sequential subsets (Westfall 1997) and, for **ADJUST=SIMULATE**, the step-down simulation results.

**TYPE=LOGICAL**  $\langle(n)\rangle$

**TYPE=FREE** If you specify **TYPE=LOGICAL**, the step-down adjustments are computed using maximal logically consistent sequential subsets of equality hypotheses (Schaffer 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 ten 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.

#### **UPPER UPPERTAILED**

requests that the  $p$ -value for the  $t$ -test be based only on values 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 the **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, since this adjustment is based on only the right tail of the  $F$  distribution.

---

## **MODEL Statement**

**MODEL** *response*  $\langle(\text{response options})\rangle$  =  $\langle$  *fixed-effects*  $\rangle$   $\langle$  / *options*  $\rangle$  ;

**MODEL** *events/trials* =  $\langle$  *fixed-effects*  $\rangle$   $\langle$  / *options*  $\rangle$  ;

The **MODEL** statement is required and names the dependent variable and the fixed-effects. The fixed-effects determine the  $\mathbf{X}$  matrix of the model (see the “[Notation for the Generalized Linear Mixed Model](#)” section on page 7 for details). The specification of effects (Chapter 34, *SAS/STAT User's Guide*) is the same as in the **GLM** or **MIXED** procedures. In contrast to **PROC GLM**, you do not specify random effects in the **MODEL** statement. However, in contrast to **PROC GLM** and **PROC MIXED**, continuous variables on the left-hand and right-hand side of the **MODEL** statement can be computed through **PROC GLIMMIX** programming statements.

An intercept is included in the fixed-effects model by default. It can be removed with the **NOINT** option.

The dependent variable can be specified using either the *response* syntax or the *events/trials* syntax. The *events/trials* syntax is specific to models for binomial data. A binomial( $n, \pi$ ) variable is the sum of  $n$  independent Bernoulli trials with event probability  $\pi$ . Each Bernoulli trial results either in an event, or a non-event (with probability  $1 - \pi$ ). You use the *events/trials* syntax to indicate to the GLIMMIX procedure that the Bernoulli outcomes are grouped. The value of the second variable, *trials*, gives the number  $n$  of Bernoulli trials. The value of the first variable, *events*, is the number of events out of  $n$ . The values of both *events* and (*trials-events*) must be nonnegative and the value of trials must be positive. Observations for which these conditions are not met are excluded from the analysis. If the *events/trials* syntax is used, the GLIMMIX procedure defaults to the binomial distribution. The response is then the *events* variable. The *trials* variable is accounted in model fitting as an additional weight. If you use the *response* syntax, the procedure defaults to the normal distribution.

### Response Variable Options

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

You can specify the following options by enclosing them in a pair of parentheses after the response variable. See the “[Response Level Ordering and Referencing](#)” section on page 148 for more detail and examples.

#### **DESCENDING | DESC**

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

#### **EVENT='category' | keyword**

specifies the event category for the binary response model. PROC GLIMMIX models the probability of the event category. The **EVENT=** option has no effect when there are more than two response categories. You can specify the value (formatted, if a format is applied) of the event category in quotes, or you can specify one of the following keywords. The default is **EVENT=FIRST**.

**FIRST**                    designates the first ordered category as the event.

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

#### **ORDER= DATA | FORMATTED | FREQ | INTERNAL**

specifies the sort order for the levels of the response variable. When **ORDER=FORMATTED** (the default) for numeric variables for which you have supplied no explicit format (that is, for which there is no corresponding **FORMAT** statement in the current PROC GLIMMIX run or in the **DATA** step that created the data set), the levels are ordered by their internal (numeric) value. The following table shows the interpretation of the **ORDER=** values.

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

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

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

**REFERENCE=**'category' | keyword

**REF=**'category' | keyword

specifies the reference category for the generalized logit model and the binary response model. For the generalized logit model, each nonreference category is contrasted with the reference category. For the binary response model, specifying one response category as the reference is the same as specifying the other response category as the event category. You can specify the value (formatted if a format is applied) of the reference category in quotes or you can specify one of the following keywords. The default is REF=LAST.

FIRST            designates the first ordered category as the reference category.

LAST             designates the last ordered category as the reference category.

### Model Options

**ALPHA=**number

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

**CHISQ**

requests that chi-square tests be performed for all specified 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.

**CL**

requests that *t* type confidence limits be constructed for each of the fixed-effects parameter estimates. The confidence level is 0.95 by default; this can be changed with the **ALPHA=** option.



**CORRB**

produces the correlation matrix of the approximate variance-covariance matrix of the fixed-effects parameter estimates.

**COVB**

produces the approximate variance-covariance matrix of the fixed-effects parameter estimates  $\hat{\beta}$ . In a generalized linear mixed model this matrix typically takes the form  $(\mathbf{X}'\hat{\mathbf{V}}^{-1}\mathbf{X})^{-1}$  and is obtained by sweeping the mixed model equations; see the “Estimated Precision of Estimates” section. In a model without random effects, it is obtained from the inverse of the observed or expected Hessian matrix. Which Hessian is used in the computation depends on whether the procedure is in scoring mode (see the **SCORING=** option of the PROC GLIMMIX statement) and whether the **EXPHESSIAN** option is in effect.

**COVBI**

produces the inverse of the approximate variance-covariance matrix of the fixed-effects parameter estimates.

**DDF=***value-list*

enables you to specify your own denominator degrees of freedom for the fixed effects. The *value-list* specification is a list of numbers or missing values (.) separated by commas. The degrees of freedom should be listed in the order in which the effects appear in the “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. For example,

```
model Y = A B A*B / ddf=3, ., 4.7;
```

assigns 3 denominator degrees of freedom to A and 4.7 to A\*B, while those for B remain the same. If you select a degrees-of-freedom method with the **DDFM=** option, nonmissing, positive values in *value-list* override the degrees of freedom for the particular effect. For example,

```
model Y = A B A*B / ddf=3, ., 6 ddfm=Satterth;
```

assigns 3 and 6 denominator degrees of freedom in the test of the A main effect and the A\*B interaction, respectively. The denominator degrees of freedom for the test for the B effect are determined from a Satterthwaite approximation.

Note that the **DDF=** and **DDFM=** options determine the degrees of freedom in the “Type I Tests of Fixed Effects,” “Type I Tests of Fixed Effects,” and “Type III Tests of Fixed Effects” tables. These degrees of freedom are also used in determining the degrees of freedom in tests and confidence intervals from the **CONTRAST**, **ESTIMATE**, and **LSMEANS** statements. Exceptions from this rule are noted in the documentation for the respective statements.

**DDFM=BETWITHIN**  
**DDFM=CONTAIN**  
**DDFM=KENWARDROGER**  
**DDFM=NONE**  
**DDFM=RESIDUAL**  
**DDFM=SATTERTHWAITE**

specifies the method for computing the denominator degrees of freedom for the tests of fixed effects resulting from the MODEL, CONTRAST, ESTIMATE, LSMEANS, and LSMESTIMATE statements.

The following table lists syntax aliases for the degrees-of-freedom methods.

**Table 2.** Aliases for DDFM= Option

DDFM= OPTION	ALIAS
BETWITHIN	BW
CONTAIN	CON
KENWARDROGER	KENROG, KR
RESIDUAL	RES
SATTERTHWAITE	SATTERTH, SAT

The DDFM=BETWITHIN option divides the residual degrees of freedom into between-subject and within-subject portions. PROC GLIMMIX then determines whether a fixed effect changes within any subject. If the GLIMMIX procedure does not process the data by subjects, the DDFM=BETWITHIN option has no effect. See the section “Processing by Subjects” on page 129 for details. If so, it assigns within-subject degrees of freedom to the effect; otherwise, it assigns the between-subject degrees of freedom to the effect (refer to Schluchter and Elashoff 1990). If there are multiple within-subject effects containing classification variables, the within-subject degrees of freedom are partitioned into components corresponding to the subject-by-effect interactions.

One exception to the preceding method is the case when you model only R-side covariation with an unstructured covariance matrix (TYPE=UN option). In this case, all fixed effects are assigned the between-subject degrees of freedom to provide for better small-sample approximations to the relevant sampling distributions. The DDFM=BETWITHIN method is the default for models with only R-side random effects and a SUBJECT= option.

The DDFM=CONTAIN option invokes the *containment method* to compute denominator degrees of freedom, and this method is the default when the model contains G-side random effects. The containment method is carried out as follows: Denote the fixed effect in question **A** and search the RANDOM effect list for the effects that *syntactically* contain **A**. For example, the RANDOM effect **B(A)** contains **A**, but the RANDOM effect **C** does not, even if it has the same levels as **B(A)**.

Among the RANDOM effects that contain **A**, compute their rank contributions to the  $[\mathbf{X} \ \mathbf{Z}]$  matrix (in order). The denominator degrees of freedom assigned to **A** is the smallest of these rank contributions. If no effects are found, the denominator df for **A** is set equal to the residual degrees of freedom,  $n - \text{rank}[\mathbf{X} \ \mathbf{Z}]$ . This choice of degrees of freedom is the same as for the tests performed for balanced split-plot designs and should be adequate for moderately unbalanced designs.

**CAUTION:** If you have a **Z** matrix with a large number of columns, the overall memory requirements and the computing time after convergence can be substantial for the containment method. If it is too large, you may want to use a different degrees-of-freedom method; for example, DDFM=RESIDUAL, DDFM=NONE, or DDFM=BETWITHIN.

The DDFM=NONE specifies that no denominator degrees of freedom be applied. PROC GLIMMIX then essentially assumes that infinite degrees of freedom are available in the calculation of  $p$ -values. The  $p$ -values for  $t$  tests are then identical to  $p$ -values derived from the standard normal distribution. In the case of  $F$  tests, the  $p$ -values equal those of chi-square tests determined as follows: if  $F_{obs}$  is the observed value of the  $F$  test with  $l$  numerator degrees of freedom, then

$$p = \Pr\{F_{l,\infty} > F_{obs}\} = \Pr\{\chi_l^2 > lF_{obs}\}$$

Regardless of the DDFM= method, you can obtain these chi-square  $p$ -values with the **CHISQ** option of the MODEL statement.

The DDFM=RESIDUAL option performs all tests using the residual degrees of freedom,  $n - \text{rank}(\mathbf{X})$ , where  $n$  is the sum of the frequencies used. It is the default degrees of freedom method for GLMs and overdispersed GLMs.

The DDFM=KENWARDROGER option applies the (prediction) standard error and degrees-of-freedom correction detailed by Kenward and Roger (1997). This approximation involves inflating the estimated variance-covariance matrix of the fixed and random effects by the method proposed by Prasad and Rao (1990) and Harville and Jeske (1992); refer also to Kackar and Harville (1984). Satterthwaite-type degrees of freedom are then computed based on this adjustment. By default, the observed information matrix of the covariance parameter estimates is used in the calculations.

The DDFM=SATTERTHWAITE option performs a general Satterthwaite approximation for the denominator degrees of freedom in a generalized linear mixed model. This method is a generalization of the techniques described in Giesbrecht and Burns (1985), McLean and Sanders (1988), and Fai and Cornelius (1996). The method can also include estimated random effects. The calculations require extra memory to hold  $q$  matrices that are the size of the mixed model equations, where  $q$  is the number of covariance parameters. Extra computing time is also required to process these matrices. The Satterthwaite method implemented is intended to produce an accurate  $F$  approximation; however, the results may differ from those produced by PROC GLM. Also, the small sample properties of this approximation have not been extensively investigated for the various models available with PROC GLIMMIX. Computational details can be found in the section “[Satterthwaite Degrees of Freedom Approximation](#)” on page 126.

When the asymptotic variance matrix of the covariance parameters is found to be singular, a generalized inverse is used. Covariance parameters with zero variance then do not contribute to the degrees of freedom adjustment for DDFM=SATTERTH and DDFM=KENWARDROGER, and a message is written to the LOG.

**DISTRIBUTION | DIST | D | ERROR | ERR = keyword**

specifies the built-in (conditional) probability distribution of the data. If you specify the DIST= option and you do not specify a user-defined link function, a default link function is chosen according to the following table. If you do not specify a distribution, the GLIMMIX procedure defaults to the normal distribution for continuous response variables and to the multinomial distribution for classification or character variables, unless the *events/trial* syntax is used in the MODEL statement. If you choose the *events/trial* syntax, the GLIMMIX procedure defaults to the binomial distribution.

Table 3 lists the values of the DIST= option and the corresponding default link functions.

**Table 3.** Values of the DIST= Option

DIST=	Distribution	Default Link Function	Numeric Value
BETA	beta	logit	12
BINARY	binary	logit	4
BINOMIAL   BIN   B	binomial	logit	3
EXPONENTIAL   EXPO	exponential	log	9
GAMMA   GAM	gamma	log	5
GAUSSIAN   G   NORMAL   N	normal	identity	1
GEOMETRIC   GEOM	geometric	log	8
INVGAUSS   IGAUSSIAN   IG	inverse Gaussian	inverse squared (power(-2) )	6
LOGNORMAL   LOGN	log-normal	identity	11
MULTINOMIAL   MULTI   MULT	multinomial	cumulative logit	NA
NEGBINOMIAL   NEGBIN   NB	negative binomial	log	7
POISSON   POI   P	Poisson	log	2
TCENTRAL   TDIST   T	<i>t</i>	identity	10
BYOBS( <i>variable</i> )	multivariate	varied	NA

Note that the PROC GLIMMIX default link for the gamma or exponential distributions is not the canonical link (the reciprocal link).

The numeric value in the last column of Table 3 can be used in combination with DIST=BYOBS. The BYOBS(*variable*) syntax designates a variable whose value identifies the distribution to which an observation belongs. If the variable is numeric, its values must match values in the last column of Table 3. If the variable is not numeric, an observation's distribution is identified by the first four characters of the

distribution's name in the left-most column of the table. Distributions whose numeric value is "NA" cannot be used with DIST=BYOBS.

If the variable in BYOBS(*variable*) is a data set variable, it can also be used in the CLASS statement of the GLIMMIX procedure. For example, this provides a convenient method to model multivariate data jointly while varying fixed effects components across outcomes. Assume that, for example, for each patient, a count and a continuous outcome were observed; the count data are modeled as Poisson data and the continuous data as gamma variates. The statements

```
proc sort data=yourdata;
  by patient;
run;
data yourdata;
  set yourdata;
  by patient;
  if first.patient then dist='POIS' else dist='GAMM';
run;
proc glimmix data=yourdata;
  class treatment dist;
  model y = dist treatment*dist / dist=byobs(dist);
run;
```

fit a Poisson and a gamma regression model simultaneously. The two models have separate intercepts and treatment effects. To correlate the outcomes, you can share a random effect between the observations from the same patient:

```
proc glimmix data=yourdata;
  class treatment dist patient;
  model y = dist treatment*dist / dist=byobs(dist);
  random intercept / subject=patient;
run;
```

or you could use an R-side correlation structure

```
proc glimmix data=yourdata;
  class treatment dist patient;
  model y = dist treatment*dist / dist=byobs(dist);
  random _residual_ / subject=patient type=un;
run;
```

Although DIST=BYOBS(*variable*) is used to model multivariate data, you only need a single response variable in PROC GLIMMIX. The responses are in "univariate" form. This allows, for example, different missing value patterns across the responses. It does, however, require that all response variables be numeric.

The default links that are assigned when DIST=BYOBS is in effect correspond to the respective default links in [Table 3](#).

When you choose DIST=LOGNORMAL, the GLIMMIX procedure models the logarithm of the response variable as a normal random variable. That is, the mean and variance are estimated on the logarithmic scale, assuming a normal distribution,  $\log\{Y\} \sim N(\mu, \sigma^2)$ . This enables you to draw on options that require a distribution in the exponential family; for example, using a scoring algorithm in a GLM. To convert means and variances for  $\log\{Y\}$  into those of  $Y$ , use the relationships

$$\begin{aligned} E[Y] &= \exp\{\mu\}\sqrt{\omega} \\ \text{Var}[Y] &= \exp\{2\mu\}\omega(\omega - 1) \\ \omega &= \exp\{\sigma^2\} \end{aligned}$$

The DIST=TCENTRAL option models the data as a shifted and scaled central  $t$  variable. This enables you to model data with heavy-tailed distributions. If  $Y$  denotes the response and  $X$  has a  $t_\nu$  distribution with  $\nu$  degrees of freedom, then PROC GLIMMIX models

$$Y = \mu + \phi \sqrt{\frac{\nu - 2}{\nu}} X$$

By default,  $\nu = 3$ . You can supply different degrees of freedom for the  $t$  variate as in the following statements

```
proc glimmix;
  class a b;
  model y = b x b*x / dist=tcentral(9.6);
  random a;
run;
```

The GLIMMIX procedure does not accept values for the degrees of freedom parameter less than 3.0. If the  $t$  distribution is used with the DIST=BYOBS(*variable*) specification, the degrees of freedom are fixed at  $\nu = 3$ . For mixed models where parameters are estimated based on linearization, choosing DIST=TCENTRAL instead of DIST=NORMAL affects only the residual variance, which increases by the factor  $\nu/(\nu - 2)$ .

The DIST=BETA option implements the parameterization of the beta distribution in Ferrari and Cribari-Neto (2004). If  $Y$  has a beta( $\alpha, \beta$ ) density, so that  $E[Y] = \mu = \alpha/(\alpha + \beta)$ , this parameterization uses the variance function  $a(\mu) = \mu(1 - \mu)$  and  $\text{var}[Y] = a(\mu)/(1 + \phi)$ .

See the section “Maximum Likelihood” beginning on page 114 for the log likelihoods of the distributions fitted by the GLIMMIX procedure.

## E

requests that Type I, Type II, and Type III L matrix coefficients be displayed for all specified effects.

**E1 | EI**

requests that Type I L matrix coefficients be displayed for all specified effects.

**E2 | EII**

requests that Type II L matrix coefficients be displayed for all specified effects.

**E3 | EIII**

requests that Type III L matrix coefficients be displayed for all specified 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; corresponding to Type I, Type II, and Type III tests. The default value is 3. You can specify several types by separating the values with a comma or a space. The ODS table names are “Tests1”, “Tests2”, and “Tests3” for the Type I, Type II, and Type III tests, respectively.

**INTERCEPT**

adds a row to the tables for Type I, II, and III tests corresponding to the overall intercept.

**LINK =** *keyword*

specifies the link function in the generalized linear mixed model. The keywords and their associated built-in link functions are shown in Table 4.

**Table 4.** Built-in Link Functions of the GLIMMIX Procedure

<b>LINK=</b>	<b>Link Function</b>	$g(\mu) = \eta =$	<b>Numeric Value</b>
CUMCLL   CCLL	cumulative complementary log-log	$\log(-\log(1 - \pi))$	NA
CUMLOGIT   CLOGIT	cumulative logit	$\log(\gamma/(1 - \pi))$	NA
CUMLOGLOG	cumulative log-log	$-\log(-\log(\pi))$	NA
CUMPROBIT   CPROBIT	cumulative probit	$\Phi^{-1}(\pi)$	NA
CLOGLOG   CLL	complementary log-log	$\log(-\log(1 - \mu))$	5
GLOGIT   GENLOGIT	generalized logit		NA
IDENTITY   ID	identity	$\mu$	1
LOG	log	$\log(\mu)$	4
LOGIT	logit	$\log(\mu/(1 - \mu))$	2
LOGLOG	log-log	$-\log(-\log(\mu))$	6
PROBIT	probit	$\Phi^{-1}(\mu)$	3
POWER( $\lambda$ )   POW( $\lambda$ )	power with exponent $\lambda = \textit{number}$	$\begin{cases} \mu^\lambda & \text{if } \lambda \neq 0 \\ \log(\mu) & \text{if } \lambda = 0 \end{cases}$	NA
POWERMINUS2	power with exponent -2	$1/\mu^2$	8
RECIPROCAL   INVERSE	reciprocal	$1/\mu$	7
BYOBS( <i>variable</i> )	varied	varied	NA

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

lative category probability. The cumulative and generalized logit link functions are appropriate only for the multinomial distribution. When you choose a cumulative link function, PROC GLIMMIX assumes that the data are ordinal. When you specify LINK=GLOGIT, the GLIMMIX procedure assumes that the data are nominal (not ordered).

The numeric value in the rightmost column of Table 4 can be used in conjunction with LINK=BYOBS(*variable*). This syntax designates a *variable* whose values identify the link function associated with an observation. If the variable is numeric, its values must match those in the last column of Table 4. If the variable is not numeric, an observation's link function is determined by the first four characters of the link's name in the first column. Those link functions whose numeric value is "NA" cannot be used with LINK=BYOBS(*variable*).

You can define your own link function through programming statements. See the section "User-Defined Link or Variance Function" on page 110 for more information on how to specify a link function. If a user-defined link function is in effect, the specification in the LINK= option is ignored. If you specify neither the LINK= option nor a user-defined link function, then the default link function is chosen according to Table 3.

#### **LWEIGHT=FIRSTORDER | FIRO**

#### **LWEIGHT=NONE**

#### **LWEIGHT=VAR**

determines how weights are used in constructing the coefficients for Type I through Type III **L** matrices. The default is LWEIGHT=VAR, and the values of the WEIGHT variable are used in forming cross-product matrices. If you specify LWEIGHT=FIRO, the weights incorporate the WEIGHT variable as well as the first-order weights of the linearized model. For LWEIGHT=NONE, the **L** matrix coefficients are based on the raw cross-product matrix, whether a WEIGHT variable is specified or not.

#### **NOCENTER**

requests that the columns of the **X** matrix are not centered and scaled. By default, the columns of **X** are centered and scaled. Unless the NOCENTER option is in effect, **X** is replaced by **X\*** during estimation. The columns of **X\*** are computed as follows:

- In models with an intercept, the intercept column remains the same and the *j*th entry in row *i* of **X\*** is

$$x_{ij}^* = \frac{x_{ij} - \bar{x}_j}{\sqrt{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}}$$

- In models without intercept, no centering takes place and the *j*th entry in row *i* of **X\*** is

$$x_{ij}^* = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}}$$

The effects of centering and scaling are removed when results are reported. For example, if the covariance matrix of the fixed effects is printed with the COVB option



of the MODEL statement, the covariances are reported in terms of the original parameters, not the centered and scaled versions. If you specify the `STDCOE` option, fixed effects parameter estimates and their standard errors are reported in terms of the standardized (scaled and/or centered) coefficients in addition to the usual results in noncentered form.

#### **NOINT**

requests that no intercept be included in the fixed-effects model. An intercept is included by default.

#### **ODDSRATIO**<(odds-ratio-options)>

#### **OR**<(odds-ratio-options)>

requests estimates of odds ratios and their confidence limits provided the link function is either the logit, cumulative logit, or generalized logit. Odds ratios are produced for the following:

- Classification main effects, if they appear in the MODEL statement
- Continuous variables in the MODEL statement, unless they appear in an interaction with a classification effect.
- Continuous variables in the MODEL statement at fixed levels of a classification effect, if the MODEL statement contains an interaction of the two
- Continuous variables in the MODEL statement if they interact with other continuous variables

You can specify the following *odds-ratio-options* to create customized odds ratio results.

`DIFF`<=*diff-type*> controls the type of differences for classification main effects.

By default, odds ratios compare the odds of a response for level  $j$  of a factor to the odds of the response for the last level of that factor (`DIFF=LAST`). The `DIFF=FIRST` option compares the levels against the first level, `DIFF=ALL` produces odds ratios based on all pairwise differences, and `DIFF=NONE` suppresses odds ratios for classification main effects.

`AT var-list=value-list` specifies the reference values for continuous variables in the model. By default, the average value serves as the reference. Consider, for example, the statements

```
proc glimmix;
  class A;
  model y = A x A*x / dist=binary oddsratio;
run;
```

Odds ratios for `A` are based on differences of least-squares means for which `x` is set to its mean. Odds ratios for `x` are computed by differencing two sets of least-squares mean for the `A` factor. One set is computed at  $x = \bar{x} + 1$ , and the second set is computed at  $x = \bar{x}$ . The MODEL statement

```

model y = A x A*x / dist=binary
                    oddsratio(at x=3);

```

changes the reference value for  $x$  to 3.

UNIT *var-list=value-list* specifies the units in which the effects of continuous variable in the model are assessed. By default, odds ratios are computed for a change of one unit from the average. Consider a model with a classification factor  $A$  with 4 levels. The statements

```

proc glimmix;
  class A;
  model y = A x A*x / dist=binary
                    oddsratio(diff=all
                               unit x=2);
run;

```

produce an “Odds Ratio Estimates” table with 10 rows. The first  $4 \times 3/2 = 6$  rows correspond to pairwise differences of levels of  $A$ . The underlying log odds ratios are computed as differences of  $A$  least-squares means. In the least-squares mean computation the covariate  $x$  is set to  $\bar{x}$ . The next four rows compare least-squares means for  $A$  at  $x = \bar{x} + 2$  and at  $x = \bar{x}$ . You can combine the AT and UNIT options to produce custom odds ratios. For example, the statements

```

proc glimmix;
  class A;
  model y = A x x*z / dist=binary
                    oddsratio(diff=all
                               at x = 3
                               unit x z = 2 4);
run;

```

produce an “Odds Ratio Estimates” table with 8 rows. The first  $4 \times 3/2 = 6$  rows correspond to pairwise differences of levels of  $A$ . The underlying log odds ratios are computed as differences of  $A$  least-squares means. In the least-squares mean computation the covariate  $x$  is set to 3, and the covariate  $x^*z$  is set to  $3\bar{z}$ . The next odds ratio measures the effect of a change in  $x$ . It is based on differencing the linear predictor for  $x = 3 + 2$  and  $x^*z = (3 + 2)\bar{z}$  with the linear predictor for  $x = 3$  and  $x^*z = 3\bar{z}$ . The last odds ratio expresses a change in  $z$  by contrasting the linear predictors based on  $x = 3$  and  $x^*z = 3(\bar{z} + 4)$  with the predictor based on  $x = 3$  and  $x^*z = 3\bar{z}$ .

To compute odds and odds ratios for general estimable functions and least-squares means, see the [ODDSRATIO](#) option of the [LSMEANS](#) statement and the [EXP](#) options of the [ESTIMATE](#) and [LSMESTIMATE](#) statements.

For important details concerning interpretation and computation of odds ratios with the GLIMMIX procedure, see the “[Odds and Odds Ratio Estimation](#)” section (page 138).

**OFFSET=***variable*

specifies a variable to be used as an offset for the linear predictor. An offset plays the role of a fixed effect whose coefficient is known to be 1. You can use an offset in a Poisson model, for example, when counts have been obtained in time intervals of different lengths. With a log link function, you can model the counts as Poisson variables with the logarithm of the time interval as the offset variable. The offset variable cannot appear in the [CLASS](#) statement or elsewhere in the MODEL or [RANDOM](#) statements.

**REFLINP=***r*

specifies a value for the linear predictor of the reference level in the generalized logit model for nominal data. By default  $r = 0$ .

**SOLUTION | S**

requests that a solution for the fixed-effects parameters be produced. Using notation from the “[Notation for the Generalized Linear Mixed Model](#)” section beginning on page 7, the fixed-effects parameter estimates are  $\hat{\beta}$ , and their (approximate) estimated standard errors are the square roots of the diagonal elements of  $\widehat{\text{var}}[\hat{\beta}]$ . This matrix commonly is of the form  $(\mathbf{X}'\hat{\mathbf{V}}^{-1}\mathbf{X})^{-1}$  in GLMMs. You can output this approximate variance matrix with the [COVB](#) option. See the section “[Details](#)” on page 113 section on the construction of  $\hat{\mathbf{V}}$  in the various models.

Along with the estimates and their approximate standard errors, a  $t$  statistic is computed as the estimate divided by its standard error. The degrees of freedom for this  $t$  statistic matches the one appearing in the “Tests of Fixed Effects” table under the effect containing the parameter. [DDFM=KENWARDROGER](#) or [DDFM=SATTERTH](#), the degrees of freedom are computed separately for each fixed-effect estimate, unless you override the value for any specific effect with the [DDF=value-list](#) option. The “Pr > |t|” column contains the two-tailed  $p$ -value corresponding to the  $t$  statistic and associated degrees of freedom. You can use the [CL](#) option to request confidence intervals for the fixed-effects parameters; they are constructed around the estimate by using a radius of the standard error times a percentage point from the  $t$  distribution.

**STDCOEF**

reports solutions for fixed effects in terms of the standardized (scaled and/or centered) coefficients. This option has no effect when the [NOCENTER](#) option is specified or in models for multinomial data.

**ZETA=***number*

tunes the sensitivity in forming Type III functions. Any element in the estimable function basis with an absolute value less than *number* is set to 0. The default is 1E–8.

## NLOPTIONS Statement

**NLOPTIONS** < options >;

Most models fit with the GLIMMIX procedure have one or more nonlinear parameters. Estimation requires nonlinear optimization methods. You can control the optimization through options of the NLOPTIONS statement.

Several estimation methods of the GLIMMIX procedure (METHOD=RSPL, MSPL, RMPL, MMPL) are doubly iterative in the following sense. The generalized linear mixed model is approximated by a linear mixed model based on current values of the covariance parameter estimates. The resulting linear mixed model is then fit, which is itself an iterative process (with some exceptions). On convergence, new covariance parameters and fixed effects estimates are obtained and the approximated linear mixed model is updated. Its parameters are again estimated iteratively. It is thus reasonable to refer to *outer* and *inner* iterations. The outer iterations involve the repeated updates of the linear mixed models, and the inner iterations are the iterative steps that lead to parameter estimates in any given linear mixed model. The NLOPTIONS statement controls the inner iterations. The outer iteration behavior can be controlled with options of the PROC GLIMMIX statement; for example, MAXLMMUPDATE, PCONV=, ABSPCONV=. If the estimation method involves a singly iterative approach, then there is no need for the outer cycling and the model is fit in a single optimization controlled by the NLOPTIONS statement (see the “Singly or Doubly Iterative Fitting” section on page 151).

The nonlinear optimization options are described in alphabetical order after Table 5, 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. The objective function being minimized, its  $p \times 1$  gradient vector and  $p \times p$  Hessian matrix are denoted as  $f(\psi)$ ,  $g(\psi)$ , and  $H(\psi)$ , respectively. Superscripts in parentheses denote the iteration count; for example,  $f(\psi)^{(k)}$  is the value of the objective function at iteration  $k$ . Depending on the formulation of the generalized linear mixed model and the estimation technique chosen, the parameter vector  $\psi$  may consist of fixed-effects only, covariance parameters only, or fixed-effects and covariance parameters.

**Table 5.** Options to Control Optimization

Option	Description
<b>Optimization Specifications</b>	
HESCAL=	type of Hessian scaling
INHESIAN=	start for approximated Hessian
LINESEARCH=	line-search method
LSPRECISION=	line-search precision
RESTART=	iteration number for update restart
TECHNIQUE=	minimization technique
UPDATE=	update technique
<b>Termination Criteria Specifications</b>	
ABSCONV=	absolute function convergence criterion

Table 5. (continued)

Option	Description
ABSFCNV=	absolute function convergence criterion
ABSGCONV=	absolute gradient convergence criterion
ABSXCONV=	absolute parameter convergence criterion
FCONV=	relative function convergence criterion
FSIZE=	used in FCONV, GCONV criterion
GCONV=	relative gradient convergence criterion
MAXFUNC=	maximum number of function calls
MAXITER=	maximum number of iterations
MAXTIME=	upper limit seconds of CPU time
MINITER=	minimum number of iterations
XCONV=	relative parameter convergence criterion
XSIZE=	used in XCONV criterion
<b>Step Length Options</b>	
DAMPSTEP=	dampens steps in line search
INSTEP=	initial trust region radius
MAXSTEP=	maximum trust region radius
<b>Remote Monitoring Options</b>	
SOCKET=	specify 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.

**ABSFCNV=r < [n] >**

**ABSFTOL=r < [n] >**

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

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

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. Termination requires the maxi-

imum 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  $\xi^{(k)}$  with the smallest function value to the other  $p$  simplex points  $\boldsymbol{\psi}_l^{(k)} \neq \xi^{(k)}$ :

$$\delta^{(k)} = \sum_{\boldsymbol{\psi}_l^{(k)} \neq \xi^{(k)}} \| \boldsymbol{\psi}_l^{(k)} - \xi^{(k)} \|_1$$

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.

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

The default is  $r=10^{-FDIGITS}$  where FDIGITS is the value of the FDIGITS= option of the PROC GLIMMIX statement. 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 is small,

$$\frac{g(\psi^{(k)})^T [H^{(k)}]^{-1} g(\psi^{(k)})}{\max(|f(\psi^{(k)})|, \text{FSIZE})} \leq r$$

where FSIZE is defined by the FSIZE= option. For the CONGRA technique (where a reliable Hessian estimate  $H$  is not available), the following criterion is used:

$$\frac{\|g(\psi^{(k)})\|_2^2 \|s(\psi^{(k)})\|_2}{\|g(\psi^{(k)}) - g(\psi^{(k-1)})\|_2 \max(|f(\psi^{(k)})|, \text{FSIZE})} \leq r$$

This criterion is not used by the NMSIMP technique. The default value is  $r = 1E-8$ . 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 matrix used in NRRIDG, TRUREG, NEWRAP, or DBLDOG optimization. If HS is not equal to 0, the first iteration and each restart iteration sets 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. 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 is 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  $d_i$  is 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  $rI$ .

By default, if you do not specify the option INHESSIAN= $r$ , the initial estimate of the approximate Hessian is set to the multiple of the identity matrix  $rI$ , where the scalar  $r$  is computed from the magnitude of the initial gradient.

**INSTEP= $r$**

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, 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, 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, using TECH=NMSIMP, the INSTEP= $r$  option defines the size of the start simplex.



**LINESEARCH=*i*****LIS=*i***

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques. Refer to Fletcher (1987) for an introduction to line-search techniques. The value of *i* can be 1, . . . , 8. For CONGRA, QUANEW, and NEWRAP, the default value is  $i = 2$ .

- |       |   |
|-------|---|
| 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 LSPRECISION= option. |
| 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 LSPRECISION= 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 golden section line search (Polak 1971), which uses only function values for linear approximation.  |
| LIS=7 | specifies 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 may be necessary (Fletcher 1987). The second line-search method (which is the default for the NEWRAP, QUANEW, and CONGRA techniques) and the third line-search method approach exact line search for small LSPRECISION= values. If you have numerical problems, you should try to decrease the LSPRECISION= value to obtain a more precise line search. The default values are shown in [Table 6](#).

**Table 6.** Default Values for Linesearch 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, refer to Fletcher (1987).

**MAXFUNC=*i*****MAXFU=*i***

specifies the maximum number  $i$  of function calls in the optimization process. The default values are

- TRUREG, NRRIDG, NEWRAP: 125
- QUANEW, DBLDOG: 500
- CONGRA: 1000
- NMSIMP: 3000

Note that the optimization can terminate only after completing a full iteration. Therefore, the number of function calls that is 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

- TRUREG, NRRIDG, NEWRAP: 50
- QUANEW, DBLDOG: 200
- CONGRA: 400
- NMSIMP: 1000

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. Note that 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.

**RESTART=*i* > 0**

**REST=*i* > 0**

specifies that the QUANEW or CONGRA algorithm is restarted with a steepest descent/ascent search direction after, at most, *i* iterations. Default values are

- CONGRA: UPDATE=PB: restart is performed automatically, *i* is not used.
- CONGRA: UPDATE≠PB:  $i = \min(10p, 80)$ , where *p* is the number of parameters.
- QUANEW: *i* is the largest integer available.

**SOCKET=*fileref***

specifies the fileref that contains the information needed for remote monitoring. See the section “[Remote Monitoring](#)” on page 158 for more details.

**TECHNIQUE=*value***

**TECH=*value***

specifies the optimization technique. You can find additional information on choosing an optimization technique in the section “[Choosing an Optimization Algorithm](#)” on page 153. Valid values for the TECHNIQUE= option are

- 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.
- NMSIMP  
performs a Nelder-Mead simplex optimization.
- NONE  
does not perform any optimization. This option can be used
  - 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

In a GLMM, specifying `TECHNIQUE=NONE` has the same effect as specifying the `NOITER` option in the `PARMS` statement.

- **NEWRAP**  
performs a Newton-Raphson optimization combining 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.

The GLIMMIX procedure applies the default optimization technique shown in Table 7, depending on your model.

**Table 7.** Default Techniques

Model Family	Setting	TECHNIQUE=
GLM	DIST=NORMAL LINK=IDENTITY	NONE
GLM	otherwise	NEWRAP
GLMM	PARMS NOITER	NONE
GLMM	otherwise	QUANEW

**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.

Valid methods are

- **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).

**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**= $r > 0$

specifies the XSIZE parameter of the relative parameter termination criterion. The default value is  $r = 0$ . For more details, see the **XCONV**= option.

---

## OUTPUT Statement

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

The OUTPUT statement creates a data set that contains predicted values and residual diagnostics, computed after fitting the model. By default, all variables in the original data set are included in the output data set.

You can use the **ID** statement to select a subset of the variables from the input data set as well as computed variables for adding to the output data set. If you reassign a data set variable through programming statements, the value of the variable from the input data set supersedes the recomputed value when observations are written to the output data set. If you list the variable in the **ID** statement, however, PROC GLIMMIX saves the current value of the variable after the programming statements have been executed.

For example, suppose that data set **Scores** contains the variables **score**, **machine**, and **person**. The following statements fit a model with fixed machine and random

person effects. The variable `score` divided by 100 is assumed to follow an inverse Gaussian distribution. The (conditional) mean and residuals are saved to the data set `igausout`. Because no `ID` statement is given, the variable `score` in the output data set contains the values from the input data set.

```
proc glimmix;
  class machine person;
  score = score/100;
  p = 4*_linp_;
  model score = machine / dist=invgauss;
  random int / sub=person;
  output out=igausout pred=p resid=r;
run;
```

On the contrary, the following statements list explicitly which variables to save to the OUTPUT data set. Because the variable `score` is listed in the `ID` statement, and is (re-)assigned through programming statements, the values of `score` saved to the OUTPUT data set are the input values divided by 100.

```
proc glimmix;
  class machine person;
  score = score / 100;
  model score = machine / dist=invgauss;
  random int / sub=person;
  output out=igausout pred=p resid=r;
  id machine score _xbeta_ _zgamma_;
run;
```

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

**OUT=SAS data set**

**DATA=SAS data set**

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

**keyword<(keyword-options)><=name>**

specifies a statistic to include in the output data set and optionally assigns the variable the name `name`. You can use the *keyword-options* to control which type of a particular statistic to compute. The *keyword-options* can take on the following values:

BLUP	uses the predictors of the random effects in computing the statistic.
ILINK	computes the statistic on the scale of the data.
NOBLUP	does not use the predictors of the random effects in computing the statistic.
NOILINK	computes the statistic on the scale of the link function.

The default is to compute statistics using BLUPs on the scale of the link function (the linearized scale). For example, the OUTPUT statement

```
output out=out1 pred=predicted lcl=lower;
```

and the OUTPUT statement

```
output out=out1 pred(blup noilink)=predicted
lcl(blup noilink)=lower;
```

are equivalent. If a particular combination of keyword and keyword options is not supported, the statistic is not computed and a message is produced in the SAS Log.

A *keyword* can appear multiple times in the OUTPUT statement. Table 8 lists the keywords and the default names assigned by the GLIMMIX procedure if you do not specify a *name*. In this table,  $y$  denotes the observed response, and  $p$  the linearized pseudo-data. See the section “Pseudo-Likelihood Estimation Based on Linearization” on page 120 for details on notation and the section “Notes on Output Statistics” for further details regarding the output statistics.

**Table 8.** Keywords for Output Statistics

Keyword	Options	Description	Expression	Name
PREDICTED	Default	Linear predictor	$\hat{\eta} = \mathbf{x}'\hat{\beta} + \mathbf{z}'\hat{\gamma}$	Pred
	NOBLUP	Marginal linear predictor	$\hat{\eta}_m = \mathbf{x}'\hat{\beta}$	PredPA
	ILINK	Predicted mean	$g^{-1}(\hat{\eta})$	PredMu
	NOBLUP ILINK	Marginal mean	$g^{-1}(\hat{\eta}_m)$	PredMuPA
STDERR	Default	Standard deviation of linear predictor	$\sqrt{\text{var}[\hat{\eta} - \mathbf{z}'\hat{\gamma}]}$	StdErr
	NOBLUP	Standard deviation of marginal linear predictor	$\sqrt{\text{var}[\hat{\eta}_m]}$	StdErrPA
	ILINK	Standard deviation of mean	$\sqrt{\text{var}[g^{-1}(\hat{\eta} - \mathbf{z}'\hat{\gamma})]}$	StdErr
	NOBLUP ILINK	Standard deviation of marginal mean	$\sqrt{\text{var}[g^{-1}(\hat{\eta}_m)]}$	StdErrMuPA
RESIDUAL	Default	Residual	$r = p - \hat{\eta}$	Resid
	NOBLUP	Marginal Residual	$r_m = p - \hat{\eta}_m$	ResidPA
	ILINK	Residual on mean scale	$r_y = y - g^{-1}(\hat{\eta})$	ResidMu
	NOBLUP ILINK	Marginal residual on mean scale	$r_{ym} = y - g^{-1}(\hat{\eta}_m)$	ResidMuPA
PEARSON	Default	Pearson-type residual	$r / \sqrt{\widehat{\text{var}}[p \gamma]}$	Pearson
	NOBLUP	Marginal Pearson-type residual	$r_m / \sqrt{\widehat{\text{var}}[p]}$	PearsonPA
	ILINK	Conditional Pearson-type mean residual	$r_y / \sqrt{\widehat{\text{var}}[Y \gamma]}$	PearsonMu
STUDENT	Default	Studentized residual	$r / \sqrt{\widehat{\text{var}}[r]}$	Student
	NOBLUP	Studentized marginal residual	$r_m / \sqrt{\widehat{\text{var}}[r_m]}$	StudentPA
LCL	Default	Lower prediction limit for linear predictor		LCL

Table 8. (continued)

Keyword	Options	Description	Expression	Name
UCL	NOBLUP	Lower confidence limit for marginal linear predictor		LCLPA
	ILINK	Lower prediction limit for mean		LCLMu
	NOBLUP ILINK	Lower confidence limit for marginal mean		LCLMuPA
	Default	Upper prediction limit for linear predictor		UCL
	NOBLUP	Upper confidence limit for marginal linear predictor		UCLPA
	ILINK	Upper prediction limit for mean		UCLMu
VARIANCE	NOBLUP ILINK	Upper confidence limit for marginal mean		UCLMuPA
	Default	Conditional variance of pseudo-data	$\widehat{\text{var}}[p \gamma]$	Variance
	NOBLUP	Marginal variance of pseudo-data	$\widehat{\text{var}}[p]$	VariancePA
	ILINK	Conditional variance of response	$\widehat{\text{var}}[Y \gamma]$	Variance_Dep
	NOBLUP ILINK	Marginal variance of response	$\widehat{\text{var}}[Y]$	Variance_DepPA

Studentized residuals are computed only on the linear scale (scale of the link), unless the link is the identity, in which case the two scales are equal. The keywords RESIDUAL, PEARSON, STUDENT, VARIANCE are not available with the multinomial distribution. You can use the following shortcuts to request statistics: PRED for PREDICTED, STD for STDERR, RESID for RESIDUAL, VAR for VARIANCE.

You can specify the following options of the OUTPUT statement after the slash (/).

#### ALLSTATS

requests that all statistics are computed. If you do not use a keyword to assign a name, the GLIMMIX procedure uses the default name.

#### ALPHA=*number*

determines the coverage probability for two-sided confidence and prediction intervals. The coverage probability is computed as  $1 - \textit{number}$ . The value of *number* must be between 0 and 1; the default is 0.05.

#### DERIVATIVES

##### DER

adds derivatives of model quantities to the output data set. If, for example, the model fit requires the (conditional) log likelihood of the data, then the DERIVATIVES option writes for each observation the evaluations of the first and second derivatives of



the log likelihood with respect to `_LINF_` and `_PHI_` to the output data set. The particular derivatives produced by the GLIMMIX procedure depend on the type of model and the estimation method.

#### **NOMISS**

requests that records are written to the output data only for those observations that were used in the analysis. By default, the GLIMMIX procedure produces output statistics for all observations in the input data set.

#### **NOUNIQUE**

requests that names are not made unique in the case of naming conflicts. By default, the GLIMMIX procedure avoids naming conflicts by assigning a unique name to each output variable. If you specify the NOUNIQUE option, variables with conflicting names are not renamed. In that case, the first variable added to the output data set takes precedence.

#### **NOVAR**

requests that variables from the input data set are not added to the output data set. This option does not apply to variables listed in the `BY` or `ID` statements.

#### **OBSCAT**

requests that in models for multinomial data statistics are written to the output data set only for the response levels that corresponds to the observed level of the observation.

#### **SYMBOLS**

##### **SYM**

adds to the output data set computed variables that are defined or referenced in the program.

---

## **PARMS Statement**

**PARMS** < (*value-list*) ... > < / *options* > ;

The PARMS statement specifies initial values for the covariance or scale parameters, or it requests a grid search over several values of these parameters in generalized linear mixed models.

The *value-list* specification can take any of several forms:

$m$	a single value
$m_1, m_2, \dots, m_n$	several values
$m$ to $n$	a sequence where $m$ equals the starting value, $n$ equals the ending value, and the increment equals 1
$m$ to $n$ by $i$	a sequence where $m$ equals the starting value, $n$ equals the ending value, and the increment equals $i$
$m_1, m_2$ to $m_3$	mixed values and sequences

### Using the PARMs Statement with a GLM

If you are fitting a GLM or a GLM with overdispersion, the scale parameters are listed at the end of the “Parameter Estimates” table in the same order as *value-list*. If you specify more than one set of initial values, PROC GLIMMIX uses only the first value listed for each parameter. Grid searches using scale parameters are not possible for these models, since the fixed effects are part of the optimization.

### Using the PARMs Statement with a GLMM

If you are fitting a generalized linear mixed model, the *value-list* corresponds to the parameters as listed in the “Covariance Parameter Estimates” table. Note that this order can change depending on whether a residual variance is profiled or not; see the [NOPROFILE](#) option of the [PROC GLIMMIX](#) statement.

If you specify more than one set of initial values, PROC GLIMMIX performs a grid search of the objective function surface and uses the best point on the grid for subsequent analysis. Specifying a large number of grid points can result in long computing times.

### Options of the PARMs Statement

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

#### **HOLD=***value-list*

specifies which parameter values PROC GLIMMIX should hold equal to the specified values. For example, the statement

```
parms (5) (3) (2) (3) / hold=1,3;
```

constrains the first and third covariance parameters to equal 5 and 2, respectively. Covariance or scale parameters that are held fixed with the HOLD= option are treated as constrained parameters in the optimization. This is different from evaluating the objective function, gradient, and Hessian matrix at known values of the covariance parameters. A constrained parameter introduces a singularity in the optimization process. The covariance matrix of the covariance parameters (see the [ASYCOV](#) option of the [PROC GLIMMIX](#) statement) is then based on the projected Hessian matrix. As a consequence, the variance of parameters subjected to a HOLD= is zero. Such parameters do not contribute to the computation of denominator degrees of freedom with the [DDFM=KENWARDROGER](#) and [DDFM=SATTERTH](#) methods, for example. If you wish to treat the covariance parameters as known, without imposing constraints on the optimization, you should use the [NOITER](#) option.

When you place a hold on all parameters (or when you specify the NOITER) option in a GLMM, you may notice that PROC GLIMMIX continues to produce an iteration history. Unless your model is a linear mixed model, several recomputations of the pseudo-response may be required in linearization-based methods to achieve agreement between the pseudo-data and the covariance matrix. In other words, the GLIMMIX procedure continues to update the profiled fixed-effects estimates (and BLUPs) until convergence is achieved.

In certain models, placing a hold on covariance parameters implies that the procedure processes the parameters in the same order as if the `NOPROFILE` was in effect. This can change the order of the covariance parameters when you place a hold on one or more parameters. Models that are subject to this re-ordering are those with R-side covariance structures whose scale parameter could be profiled. This includes the `TYPE=CS`, `TYPE=SP`, `TYPE=AR`, `TYPE=TOEP`, and `TYPE=ARMA` covariance structures.

**LOWERB=***value-list*

enables you to specify lower boundary constraints for the covariance or scale parameters. The *value-list* specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the same order that PROC GLIMMIX uses for the *value-list*, and each number corresponds to the lower boundary constraint. A missing value instructs PROC GLIMMIX to use its default constraint, and if you do not specify numbers for all of the covariance parameters, PROC GLIMMIX assumes that the remaining ones are missing.

This option is useful, for example, when you want to constrain the **G** matrix to be positive definite in order to avoid the more computationally intensive algorithms required when **G** becomes singular. The corresponding code for a random coefficients model is as follows:

```
proc glimmix;
  class person;
  model y = time;
  random int time / type=chol sub=person;
  parms / lowerb=1e-4, ., 1e-4;
run;
```

Here, the `TYPE=CHOL` structure is used in order to specify a Cholesky root parameterization for the  $2 \times 2$  unstructured blocks in **G**. This parameterization ensures that the **G** matrix is nonnegative definite, and the PARMs statement then ensures that it is positive definite by constraining the two diagonal terms to be greater than or equal to  $1E-4$ .

**NOITER**

requests that no optimization of the covariance parameters be performed. This option has no effect in generalized linear models.

If you specify the NOITER option, PROC GLIMMIX uses the values for the covariance parameters given in the PARMs statement to perform statistical inferences. Note that the NOITER option is not equivalent to specifying a `HOLD=` value for all covariance parameters. If you use the NOITER option, covariance parameters are not constrained in the optimization. This prevents singularities that might otherwise occur in the optimization process.

If a residual variance is profiled, the parameter estimates can change from the initial values you provide as the residual variance is recomputed. To prevent an update of the residual variance, combine the NOITER option with the `NOPROFILE` option of the PROC GLIMMIX statements, as in the following code.

```

proc glimmix noprofile;
  class A B C rep mp sp;
  model y = A | B | C;
  random rep mp sp;
  parms (180) (200) (170) (1000) / noiter;
run;

```

When you specify the NOITER option, you may notice that the GLIMMIX procedure continues to produce an iteration history. Unless your model is a linear mixed model, several recomputations of the pseudo-response may be required in linearization-based methods to achieve agreement between the pseudo-data and the covariance matrix. In other words, the GLIMMIX procedure continues to update the profiled fixed-effects estimates (and BLUPs) until convergence is achieved. To prevent these updates, use the `MAXLMMUPDATE=` option of the `PROC GLIMMIX` statement.

Specifying the NOITER option in the PARMs statement of a GLMM has the same effect as choosing `TECHNIQUE=NONE` on the `NLOPTIONS` statement. If you wish to base initial fixed-effects estimates on the results of fitting a generalized linear model, then you can combine the NOITER option with the `TECHNIQUE=` option. For example, the statements

```

proc glimmix startglm inititer=10;
  class clinic A;
  model y/n = A / link=logit dist=binomial;
  random clinic;
  parms (0.4) / noiter;
  nloptions technique=newrap;
run;

```

determine the starting values for the fixed effects by fitting a logistic model (without random effects), using the Newton-Raphson algorithm. The initial GLM fit stops at convergence or after at most 10 iterations, whichever comes first. The pseudo-data for the linearized GLMM is computed from the GLM estimates. The variance of the Clinic random effect is held constant at 0.4 during subsequent iterations that update the fixed effects only.

If you also want to combine the GLM fixed effects estimates with known and fixed covariance parameter values, without updating the fixed effects, you can add the `MAXLMMUPDATE=0` option:

```

proc glimmix startglm inititer=10 maxlmmupdate=0;
  class clinic A;
  model y/n = A / link=logit dist=binomial;
  random clinic;
  parms (0.4) / noiter;
  nloptions technique=newrap;
run;

```

Finally, the NOITER option can be useful if you wish to obtain minimum variance quadratic unbiased estimates (with 0 priors), also known as MIVQUE0 estimates

(Goodnight 1978). Since MIVQUE0 estimates are starting values for covariance parameters—unless you provide (*value-list*) in the PARMS statement—the following statements produce MIVQUE0 mixed model estimates.

```
proc glimmix noprofile;
  class A B;
  model y = A;
  random int / subject=B;
  parms / noiter;
run;
```

**PARMSDATA=SAS-data-set**

**PDATA=SAS-data-set**

reads in covariance parameter values from a SAS data set. The data set should contain the numerical variable ESTIMATE or the numerical variables COVP1–COVP $q$ , where  $q$  denotes the number of covariance parameters.

If the PARMSDATA= data set contains multiple sets of covariance parameters, the GLIMMIX procedure evaluates the initial objective function for each set and commences the optimization step using the set with the lowest function value as the starting values. For example, the SAS statements

```
data data_covp;
  input covp1-covp4;
  datalines;
  180 200 170 1000
  170 190 160 900
  160 180 150 800
;
proc glimmix;
  class A B C rep mainEU smalleEU;
  model yield = A|B|C;
  random rep mainEU smalleEU;
  parms / pdata=data_covp;
run;
```

request that the objective function is evaluated for three sets of initial value. Each set comprises four covariance parameters.

The order of the observations in a data set with the numerical variable ESTIMATE corresponds to the order of the covariance parameters in the “Covariance Parameter Estimates” table. In a GLM, the PARMSDATA= option can be used to set the starting value for the exponential family scale parameter. A grid search is not conducted for GLMs if you specify multiple values.

The PARMSDATA= data set must not contain missing values.

If the GLIMMIX procedure is processing the input data set in BY groups, you can add the BY variables to the PARMSDATA= data set. If this data set is sorted by the BY variables, the GLIMMIX procedure matches the covariance parameter values to the current BY group. If the PARMSDATA= data set does not contain all BY variables,

the data set is processed in its entirety for every BY group and a message is written to the LOG. This enables you to provide a single set of starting values across BY groups, as in the following statements.

```
data data_covp;
  input covp1-covp4;
  datalines;
  180 200 170 1000
;
proc glimmix;
  class A B C rep mainEU smalleEU;
  model yield = A|B|C;
  random rep mainEU smalleEU;
  parms / pdata=data_covp;
  by year;
run;
```

The same set of starting values is used for each value of the `year` variable.

#### **UPPERB=value-list**

enables you to specify upper boundary constraints on the covariance parameters. The *value-list* specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the same order that PROC GLIMMIX uses for the *value-list*, and each number corresponds to the upper boundary constraint. A missing value instructs PROC GLIMMIX to use its default constraint. If you do not specify numbers for all of the covariance parameters, PROC GLIMMIX assumes that the remaining ones are missing.

---

## **RANDOM Statement**

**RANDOM** *random-effects* < / options > ;

Using notation from the “[Notation for the Generalized Linear Mixed Model](#)” section beginning on page 7, the RANDOM statement defines the **Z** matrix of the mixed model, the random effects in the  $\gamma$  vector, the structure of **G**, and the structure of **R**.

The **Z** matrix is constructed exactly as the **X** matrix for the fixed effects, and the **G** matrix is constructed to correspond to the effects constituting **Z**. The structures of **G** and **R** are defined by using the **TYPE=** option described on page 99. The random effects can be classification or continuous effects, and multiple RANDOM statements are possible.

Some reserved keywords have special significance in the *random-effects* list. You can specify INTERCEPT (or INT) as a random effect to indicate the intercept. PROC GLIMMIX does not include the intercept in the RANDOM statement by default as it does in the MODEL statement. You can specify the `_RESIDUAL_` keyword (or RESID, RESIDUAL, `_RESID_`) before the option slash (/) to indicate a residual-type (R-side) random component that defines the **R** matrix. Basically, the `_RESIDUAL_` keyword takes the place of the *random-effect* if you want to specify R-side variances and covariance structures. These keywords take precedence over variables in the data set with the same name. If your data or the covariance structure requires that an effect

is specified, you can use the RESIDUAL option to instruct the GLIMMIX procedure to model the R-side variances and covariances.

In order to add an overdispersion component to the variance function, simply specify a single residual random component. For example, the statements

```
proc glimmix;
  model count = x x*x / dist=poisson;
  random _residual_;
run;
```

fit a polynomial Poisson regression model with overdispersion. The variance function  $a(\mu) = \mu$  is replaced by  $\phi a(\mu)$ .

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

**ALPHA=number**

requests that a  $t$  type confidence interval with confidence level  $1 - \text{number}$  be constructed for the predictors of random effects on this statement. The value of *number* must be between 0 and 1; the default is 0.05. Specifying the ALPHA= option implies the CL option.

**CL**

requests that  $t$  type confidence limits be constructed for each of the predictors of random effects on this statement. The confidence level is 0.95 by default; this can be changed with the ALPHA= option. The CL option implies the SOLUTION option.

**G**

requests that the estimated **G** matrix be displayed for G-side random effects associated with this RANDOM statement. PROC GLIMMIX displays blanks for values that are 0.

**GC**

displays the lower-triangular Cholesky root of the estimated **G** matrix for G-side random effects.

**GCI**

displays the inverse Cholesky root of the estimated **G** matrix for G-side random effects.

**GCOORD=LAST**

**GCOORD=FIRST**

**GCOORD=MEAN**

determines how the GLIMMIX procedure associates coordinates for TYPE=SP() covariance structures with effect levels for G-side random effects. In these covariance structures, you specify one or more variables that identify the coordinates of a data point. The levels of classification variables, on the other hand, can occur multiple times for a particular subject. For example, in the following statements

```
proc glimmix;
  class A B;
```

```

model y = B;
random A / type=sp(pow)(x);
run;

```

the same level of **A** can occur multiple times, and the associated values of **x** may be different. The **GCOORD=LAST** option determines the coordinates for a level of the random effect from the last observation associated with the level. Similarly, the **GCOORD=FIRST** and **GCOORD=MEAN** options determine the coordinate from the first observation and from the average of the observations. Observations not used in the analysis are not considered in determining the first, last, or average coordinate.

**G CORR**

displays the correlation matrix corresponding to the estimated **G** matrix for G-side random effects.

**GI**

displays the inverse of the estimated **G** matrix for G-side random effects.

**GROUP=effect****GRP=effect**

identifies groups by which to vary the covariance parameters. Each new level of the grouping effect produces a new set of covariance parameters. Continuous variables and computed variables are permitted as group effects. PROC GLIMMIX does not sort by the values of the continuous variable; rather, it considers the data to be from a new group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of groups and also prevents the production of a large “Class Levels Information” table.

Specifying a GROUP effect can greatly increase the number of estimated covariance parameters, which may adversely affect the optimization process.

**KNOTINFO**

displays the number and coordinates of the knots as determined by the **KNOTMETHOD=** option.

**KNOTMETHOD=KDTREE** <(tree-options)>**KNOTMETHOD=EQUAL** <(numberlist)>**KNOTMETHOD=DATA** (SAS-data-set)

determines the method of constructing knots for the radial smoother fit with the **TYPE=RSMOOTH** covariance structure. Unless you select the **TYPE=RSMOOTH** covariance structure, the **KNOTMETHOD=** option has no effect. The default is **KNOTMETHOD=KDTREE**.

PROC GLIMMIX fits a low-rank smoother, meaning that the number of knots is considerably less than the number of observations. By default, PROC GLIMMIX determines the knot locations based on the vertices of a *k-d* tree (Friedman, Bentley, and Finkel 1977; Cleveland and Grosse 1991). The *k-d* tree is a tree data structure that is useful for efficiently determining the *m* nearest neighbors of a point. The *k-d* tree also can be used to obtain a grid of points that adapts to the configuration of the data. The process starts with a hypercube that encloses the values of the random



effects. The space is then partitioned recursively by splitting cells at the median of the data in the cell for the random effect. The procedure is repeated for all cells that contain more than a specified number of points,  $b$ . The value  $b$  is called the bucket size.

The  $k$ - $d$  tree is thus a division of the data into cells such that cells representing leaf nodes contain at most  $b$  values. You control the building of the  $k$ - $d$  tree through the `BUCKET=tree-option`. You control the construction of knots from the cell coordinates of the tree with the other options as follows:

`BUCKET=number` determines the bucket size  $b$ . A larger bucket size will result in fewer knots. For  $k$ - $d$  trees in more than one dimension, the correspondence between bucket size and number of knots is difficult to determine. It depends on the data configuration and on other suboptions. In the multivariate case, you may need to try out different bucket sizes to obtain the desired number of knots. The default value of *number* is 4 for univariate trees (a single random effect), and  $\lfloor 0.1n \rfloor$  in the multidimensional case.

`KNOTTYPE=type` specifies whether the knots are based on vertices of the tree cells or the centroid. The two possible values of *type* are VERTEX and CENTER. The default is `KNOTTYPE=VERTEX`. For multidimensional smoothing, such as smoothing across irregularly shaped spatial domains, the `KNOTTYPE=CENTER` option is useful to move knot locations away from the bounding hypercube toward the convex hull.

`NEAREST` specifies that knot coordinates are the coordinates of the nearest neighbor of either the centroid or vertex of the cell, as determined by the `KNOTTYPE=` suboption.

`TREEINFO` displays details about the construction of the  $k$ - $d$  tree, such as the cell splits and the split values.

See the “[Knot Selection](#)” section on page 133 for a detailed example of how the specification of the bucket size translates into the construction of a  $k$ - $d$  tree and the spline knots.

When you use the `NOFIT` option of the PROC GLIMMIX statement, the GLIMMIX procedure computes the knots but does not fit the model. This can be useful if you want to compare knot selections with different suboptions of `KNOTMETHOD=KDTREE`. Suppose you want to determine the number of knots based on a particular bucket size. The statements

```
proc glimmix nofit;
  model y = Latitude Longitude;
  random Latitude Longitude / type=rsmooth
    knotmethod=kdtree (knottype=vertex near
    bucket=10 knotinfo);
run;
```

compute and display the knots in a bivariate smooth, constructed from nearest neighbors of the vertices of a  $k$ - $d$  tree with bucket size 10.

The KNOTMETHOD=EQUAL option enables you to define a regular grid of knots. By default, PROC GLIMMIX constructs 10 knots for one-dimensional smooths and 5 knots in each dimension for smoothing in higher dimensions. You can specify a different number of knots with the optional *numberlist*. Missing values in the *numberlist* are replaced with the default values. A minimum of two knots in each dimension is required. For example, the statements

```
proc glimmix;
  model y=;
  random x1 x2 / type=rsmooth knotmethod=equal(5 7);
run;
```

use a rectangular grid of 35 knots, five knots for  $x_1$  combined with seven knots for  $x_2$ .

You can specify a data set that contains variables whose values give the knot coordinates with the KNOTMETHOD=DATA option. The data set must contain numeric variables with the same name as the radial smoothing *random-effects*. PROC GLIMMIX uses only the unique knot coordinates in the knot data set. This option is useful to provide knot coordinates different from those that can be produced from a  $k$ - $d$  tree. For example, in spatial problems where the domain is irregularly shaped, you may want to determine knots by a space-filling algorithm. The following SAS statements invoke the OPTEX procedure to compute 45 knots that uniformly cover the convex hull of the data locations (refer to Chapter 30, “Introduction,” and Chapter 31, “Details of the OPTEX Procedure,” in the *SAS/QC User’s Guide* for details about the OPTEX procedure).

```
proc optex coding=none;
  model latitude longitude / noint;
  generate n=45 criterion=u method=m_fedorov;
  output out=knotdata;
run;
proc glimmix;
  model y = Latitude Longitude;
  random Latitude Longitude / type=rsmooth
  knotmethod=data(knotdata);
run;
```

## RESIDUAL RSIDE

specifies that the random effects listed in this statement are R-side effects. You use the RESIDUAL option of the RANDOM statement if the nature of the covariance structure requires you to specify an effect.

## SOLUTION | S

requests that the solution  $\hat{\gamma}$  for the random-effects parameters be produced, if the statement defines G-side random effects.

The numbers displayed in the “Std Err Pred” column of the “Solution for Random Effects” table are not the standard errors of the  $\hat{\gamma}$  displayed in the Estimate column; rather, they are the square roots of the prediction errors  $\hat{\gamma}_i - \gamma_i$ , where  $\hat{\gamma}_i$  is the  $i$ th EBLUP and  $\gamma_i$  is the  $i$ th random-effect parameter.

**SUBJECT=effect**

**SUB=effect**

identifies the subjects in your generalized linear mixed model. Complete independence is assumed across subjects. Specifying a subject effect is equivalent to nesting all other effects in the RANDOM statement within the subject effect.

Continuous variables and computed variables are permitted with the SUBJECT= option. PROC GLIMMIX does not sort by the values of the continuous variable but considers the data to be from a new subject whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects and also prevents the production of a large “Class Levels Information” table.

**TYPE=covariance-structure**

specifies the covariance structure of **G** for G-side effects and of **R** for R-side effects.

Although a variety of structures are available, many applications call for either simple diagonal (TYPE=VC) or unstructured covariance matrices. The TYPE=VC (variance components) option is the default structure, and it models a different variance component for each random effect. It is recommended to model unstructured covariance matrices in terms of their Cholesky parameterization (TYPE=CHOL rather than TYPE=UN).

If you want different covariance structures in different parts of **G**, you must use multiple RANDOM statements with different TYPE= options.

Valid values for *covariance-structure* are as follows. Examples are shown in [Table 9](#) (page 106). The variances and covariances in the formulas that follow in the TYPE= descriptions are expressed in terms of generic random variables  $\xi_i$  and  $\xi_j$ . They represent the G-side random effects or the residual random variables for which the **G** or **R** matrices are constructed.

TYPE=AR(1) specifies a first-order autoregressive structure,

$$\text{cov}[\xi_i, \xi_j] = \sigma^2 \rho^{|i^* - j^*|}$$

The values  $i^*$  and  $j^*$  are derived for the  $i$ th and  $j$ th observations and are not necessarily the observation numbers. For example, in the statements

```
proc glimmix;
  class time patient;
  model y = x x*x;
  random time / sub=patient type=ar(1);
run;
```

the values correspond to the class levels for the time effect of the  $i$ th and  $j$ th observation within a particular subject.

PROC GLIMMIX imposes the constraint  $|\rho| < 1$  for stationarity.

TYPE=ARMA(1,1) specifies the first-order autoregressive moving average structure,

$$\text{cov}[\xi_i, \xi_j] = \begin{cases} \sigma^2 & i = j \\ \sigma^2 \gamma \rho^{|i^* - j^*| - 1} & i \neq j \end{cases}$$

Here,  $\rho$  is the autoregressive parameter,  $\gamma$  models a moving average component, and  $\sigma^2$  is a scale parameter. In the notation of Fuller (1976, p. 68),  $\rho = \theta_1$  and

$$\gamma = \frac{(1 + b_1 \theta_1)(\theta_1 + b_1)}{1 + b_1^2 + 2b_1 \theta_1}$$

The example in Table 9 and  $|b_1| < 1$  imply that

$$b_1 = \frac{\beta - \sqrt{\beta^2 - 4\alpha^2}}{2\alpha}$$

where  $\alpha = \gamma - \rho$  and  $\beta = 1 + \rho^2 - 2\gamma\rho$ . PROC GLIMMIX imposes the constraints  $|\rho| < 1$  and  $|\gamma| < 1$  for stationarity, although for some values of  $\rho$  and  $\gamma$  in this region the resulting covariance matrix is not positive definite. When the estimated value of  $\rho$  becomes negative, the computed covariance is multiplied by  $\cos(\pi d_{ij})$  to account for the negativity.

TYPE=CHOL<(q)> specifies an unstructured variance-covariance matrix parameterized through its Cholesky root. This parameterization ensures that the resulting variance-covariance matrix is at least positive semidefinite. If all diagonal values are nonzero, it is positive definite. For example, a  $2 \times 2$  unstructured covariance matrix can be written as

$$\text{var}[\xi] = \begin{bmatrix} \theta_1 & \theta_{12} \\ \theta_{12} & \theta_2 \end{bmatrix}$$

Without imposing constraints on the three parameters, there is no guarantee that the estimated variance matrix is positive definite. Even if  $\theta_1$  and  $\theta_2$  are nonzero, a large value for  $\theta_{12}$  can lead to a negative eigenvalue of  $\text{var}[\xi]$ . The Cholesky root of a positive definite matrix  $\mathbf{A}$  is a lower triangular matrix  $\mathbf{C}$  such that  $\mathbf{C}\mathbf{C}' = \mathbf{A}$ . The Cholesky root of the above  $2 \times 2$  matrix can be written as

$$\mathbf{C} = \begin{bmatrix} \alpha_1 & 0 \\ \alpha_{12} & \alpha_2 \end{bmatrix}$$

The elements of the unstructured variance matrix are then simply  $\theta_1 = \alpha_1^2$ ,  $\theta_{12} = \alpha_1 \alpha_{12}$ , and  $\theta_2 = \alpha_{12}^2 + \alpha_2^2$ . Similar operations yield the generalization to covariance matrices of higher orders.

For example, the statements

```
proc glimmix;
  class sub;
  model y = x;
  random _residual_ / subject=sub type=un;
run;
```

model the covariance matrix of each subject as an unstructured matrix. The statements

```
proc glimmix;
  class sub;
  model y = x;
  random _residual_ / subject=sub type=chol;
run;
```

accomplish the same, but the estimated  $\mathbf{R}$  matrix is guaranteed to be nonnegative definite.

The GLIMMIX procedure constrains the diagonal elements of the Cholesky root to be positive. This guarantees a unique solution when the matrix is positive definite.

The optional order parameter  $q > 0$  determines how many bands below the diagonal are modeled. Elements in the lower triangular portion of  $\mathbf{C}$  in bands higher than  $q$  are set to zero. If you consider the resulting covariance matrix  $\mathbf{A} = \mathbf{C}\mathbf{C}'$ , then the order parameter has the effect of zeroing all off-diagonal elements that are at least  $q$  positions away from the diagonal.

Because of its good computational and statistical properties, the Cholesky root parameterization is generally recommended over a completely unstructured covariance matrix (TYPE=UN, see below). However, it is computationally slightly more involved.

TYPE=CS

specifies the compound-symmetry structure, which has constant variance and constant covariance

$$\text{cov}[\xi_i, \xi_j] = \begin{cases} \phi + \sigma & i = j \\ \sigma & i \neq j \end{cases}$$

The compound symmetry structure arises naturally with nested random effects, such as when subsampling error is nested within experimental error. The models constructed with the following two sets of GLIMMIX statements have the same marginal variance matrix, provided  $\sigma$  is positive:

```
proc glimmix;
  class block A;
  model y = block A;
  random block*A / type=vc;
run;
```

```

proc glimmix;
  class block A;
  model y = block A;
  random _residual_ / subject=block*A
                                     type=cs;
run;

```

In the first case, the `block*A` random effect models the G-side experimental error. Because the distribution defaults to the normal, the  $\mathbf{R}$  matrix is of form  $\phi\mathbf{I}$ , (see Table 10), and  $\phi$  is the subsampling error variance. The marginal variance for the data from a particular experimental unit is thus  $\sigma_{b*a}^2\mathbf{J} + \phi\mathbf{I}$ . This matrix is of compound symmetric form.

Hierarchical random assignments or selections, such as subsampling or split-plot designs, give rise to compound symmetric covariance structures. This implies exchangeability of the observations on the subunit, leading to constant correlations between the observations. Compound symmetric structures are thus usually not appropriate for processes where correlations decline according to some metric, for example, spatial and temporal processes.

Note that R-side compound-symmetry structures do not impose any constraint on  $\sigma$ . You can thus use an R-side `TYPE=CS` structure to emulate a variance-component model with unbounded estimate of the variance component.

`TYPE=FA0( $q$ )` specifies a factor-analytic structure with  $q$  factors of the form  $\text{var}[\boldsymbol{\xi}] = \boldsymbol{\Lambda}\boldsymbol{\Lambda}'$ , where  $\boldsymbol{\Lambda}$  is a  $t \times q$  rectangular matrix and  $t$  is the dimension of  $\mathbf{Y}$ . When  $q > 1$ ,  $\boldsymbol{\Lambda}$  is a lower triangular matrix. When  $q < t$ , that is, when the number of factors is less than the dimension of the matrix, this structure is nonnegative definite but not of full rank. In this situation, you can use it for approximating an unstructured covariance matrix.

`TYPE=RSMOOTH<(m)>` specifies a radial smoother covariance structure for G-side random effects. This results in an approximate low-rank thin plate spline where the smoothing parameter is obtained by the estimation method selected with the `METHOD=` option of the `PROC GLIMMIX` statement. The smoother is based on the automatic smoother in Ruppert, Wand, and Carroll (2003, Chapter 13.4–13.5.), but with a different method of selecting the spline knots. See the section “Radial Smoothing Based on Mixed Models” on page 131 for further details on the construction of the smoother and the knot selection.

Radial smoothing is possible in one or more dimensions. A univariate smoother is obtained with a single random effect, while multiple random effects in a `RANDOM` statement yield a multivariate smoother. Only continuous random effects are permitted with this covariance structure. If  $n_r$  denotes the number of continuous random effects in the `RANDOM` statement, then the covari-

ance structure of the random effects  $\gamma$  is determined as follows. Suppose that  $\mathbf{z}_i$  denotes the vector of random effects for the  $i$ th observation. Let  $\boldsymbol{\tau}_k$  denote the  $(n_r \times 1)$  vector of knot coordinates,  $k = 1, \dots, K$ , and  $K$  is the total number of knots. The Euclidean distance between the knots is computed as

$$d_{kp} = \|\boldsymbol{\tau}_k - \boldsymbol{\tau}_p\| = \sqrt{\sum_{j=1}^{n_r} (\tau_{jk} - \tau_{jp})^2}$$

and the distance between knots and effects is computed as

$$h_{ik} = \|\mathbf{z}_i - \boldsymbol{\tau}_k\| = \sqrt{\sum_{j=1}^{n_r} (z_{ij} - \tau_{jk})^2}$$

The  $\mathbf{Z}$  matrix for the GLMM is constructed as

$$\mathbf{Z} = \tilde{\mathbf{Z}}\boldsymbol{\Omega}^{-1/2}$$

where the  $(n \times K)$  matrix  $\tilde{\mathbf{Z}}$  has typical element

$$[\tilde{\mathbf{Z}}]_{ik} = \begin{cases} h_{ik}^p & n_r \text{ odd} \\ h_{ik}^p \log\{h_{ik}\} & n_r \text{ even} \end{cases}$$

and the  $(K \times K)$  matrix  $\boldsymbol{\Omega}$  has typical element

$$[\boldsymbol{\Omega}]_{kp} = \begin{cases} d_{kp}^p & n_r \text{ odd} \\ d_{kp}^p \log\{d_{kp}\} & n_r \text{ even} \end{cases}$$

The exponent in these expressions equals  $p = 2m - n_r$ , where the optional value  $m$  corresponds to the derivative penalized in the thin plate spline. A larger value of  $m$  will yield a smoother fit. The GLIMMIX procedure requires  $p > 0$  and chooses by default  $m = 2$  if  $n_r < 3$  and  $m = (n_r + 1)/2$  otherwise.

Finally, the components of  $\gamma$  are assumed to have equal variance  $\sigma_r^2$ . The “smoothing parameter”  $\lambda$  of the low-rank spline is related to the variance components in the model,  $\lambda^2 = f(\phi, \sigma_r^2)$ . See Ruppert, Wand, and Carroll (2003) for details. If the conditional distribution does not provide a scale parameter  $\phi$ , you can add a single R-side residual parameter.

The knot selection is controlled with the `KNOTMETHOD=` option. The GLIMMIX procedure selects knots automatically based on the vertices of a  $k$ - $d$  tree or reads knots from a data set that you supply. See the section “[Radial Smoothing Based on Mixed Models](#)” on page 131 for further details on radial smoothing in the GLIMMIX procedure and its connection to a mixed model formulation.

TYPE=SIMPLE is an alias for TYPE=VC.

TYPE=SP(EXP)(*c-list*) models an exponential spatial or temporal covariance structure, where the covariance between two observations depends on a distance metric  $d_{ij}$ . The *c-list* contains the names of the numeric variables used as coordinates to determine distance. For a stochastic process in  $R^k$ , there are  $k$  elements in *c-list*. If the  $(k \times 1)$  vectors of coordinates for observations  $i$  and  $j$  are  $\mathbf{c}_i$  and  $\mathbf{c}_j$ , then PROC GLIMMIX computes the Euclidean distance

$$d_{ij} = \|\mathbf{c}_i - \mathbf{c}_j\| = \sqrt{\sum_{m=1}^k (c_{mi} - c_{mj})^2}$$

The covariance between two observations is then

$$\text{cov}[\xi_i, \xi_j] = \sigma^2 \exp\{-d_{ij}/\alpha\}$$

The parameter  $\alpha$  is *not* what is commonly referred to as the range parameter in geostatistical applications. The practical range of a (second-order stationary) spatial process is the distance  $d^{(p)}$  at which the correlations fall below 0.05. For the SP(EXP) structure, this distance is  $d^{(p)} = 3\alpha$ . PROC GLIMMIX constrains  $\alpha$  to be positive.

TYPE=SP(GAU)(*c-list*) models a gaussian covariance structure,

$$\text{cov}[\xi_i, \xi_j] = \sigma^2 \exp\{-d_{ij}^2/\alpha^2\}$$

See TYPE=SP(EXP) for the computation of the distance  $d_{ij}$ . The parameter  $\alpha$  is related to the range of the process as follows. If the practical range  $d^{(p)}$  is defined as the distance at which the correlations fall below 0.05, then  $d^{(p)} = \sqrt{3}\alpha$ . PROC GLIMMIX constrains  $\alpha$  to be positive. See TYPE=SP(EXP) for the computation of the distance  $d_{ij}$  from the variables specified in *c-list*.

TYPE=SP(POW)(*c-list*) models a power covariance structure,

$$\text{cov}[\xi_i, \xi_j] = \sigma^2 \rho^{d_{ij}}$$

where  $\rho \geq 0$ . This is a reparameterization of the exponential structure, TYPE=SP(EXP). Specifically,  $\log\{\rho\} = -1/\alpha$ . See TYPE=SP(EXP) for the computation of the distance  $d_{ij}$  from the variables specified in *c-list*. When the estimated value of  $\rho$  becomes negative, the computed covariance is multiplied by  $\cos(\pi d_{ij})$  to account for the negativity.

TYPE=SP(SPH)(*c-list*) models a spherical covariance structure,

$$\text{cov}[\xi_i, \xi_j] = \begin{cases} \sigma^2 \left\{ 1 - \frac{3d_{ij}}{2\alpha} + \frac{1}{2} \left( \frac{d_{ij}}{\alpha} \right)^3 \right\} & d_{ij} \leq \alpha \\ 0 & d_{ij} > \alpha \end{cases}$$



The spherical covariance structure has a true range parameter. The covariances between observations are exactly zero when their distance exceeds  $\alpha$ . See TYPE=SP(EXP) for the computation of the distance  $d_{ij}$  from the variables specified in *c-list*.

TYPE=TOEP models a Toeplitz covariance structure. This structure can be viewed as an autoregressive structure with order equal to the dimension of the matrix,

$$\text{cov}[\xi_i, \xi_j] = \begin{cases} \sigma^2 & i = j \\ \sigma_{|i-j|} & i \neq j \end{cases}$$

TYPE=TOEP( $q$ ) specifies a banded Toeplitz structure,

$$\text{cov}[\xi_i, \xi_j] = \begin{cases} \sigma^2 & i = j \\ \sigma_{|i-j|} & |i - j| < q \end{cases}$$

This can be viewed as a moving-average structure with order equal to  $q - 1$ . The specification TYPE=TOEP(1) is the same as  $\sigma^2\mathbf{I}$ , and it can be useful for specifying the same variance component for several effects.

TYPE=UN<( $q$ )> specifies a completely general (unstructured) covariance matrix parameterized directly in terms of variances and covariances,

$$\text{cov}[\xi_i, \xi_j] = \sigma_{ij}$$

The variances are constrained to be nonnegative, and the covariances are unconstrained. This structure is not constrained to be nonnegative definite in order to avoid nonlinear constraints; however, you can use the TYPE=CHOL structure if you want this constraint to be imposed by a Cholesky factorization. If you specify the order parameter  $q$ , then PROC GLIMMIX estimates only the first  $q$  bands of the matrix, setting elements in all higher bands equal to 0.

TYPE=UNR<( $q$ )> specifies a completely general (unstructured) covariance matrix parameterized in terms of variances and correlations,

$$\text{cov}[\xi_i, \xi_j] = \sigma_i \sigma_j \rho_{ij}$$

where  $\sigma_i$  denotes the standard deviation and the correlation  $\rho_{ij}$  is zero when  $i = j$  and when  $|i - j| \geq q$ , provided the order parameter  $q$  is given. This structure fits the same model as the TYPE=UN( $q$ ) option, but with a different parameterization. The  $i$ th variance parameter is  $\sigma_i^2$ . The parameter  $\rho_{ij}$  is the correlation between the  $i$ th and  $j$ th measurements; it satisfies  $|\rho_{ij}| < 1$ . If you specify the order parameter  $q$ , then PROC GLIMMIX estimates only the first  $q$  bands of the matrix, setting all higher bands equal to zero.

TYPE=VC specifies standard variance components and is the default structure for both G-side and R-side covariance structures. In a G-side covariance structure, a distinct variance component is assigned to each effect. In an R-side structure TYPE=VC is usually used only to add overdispersion effects or with the GROUP= option to specify a heterogeneous variance model.

**Table 9.** Covariance Structure Examples

Description	Structure	Example
Variance Components	VC (default)	$\begin{bmatrix} \sigma_B^2 & 0 & 0 & 0 \\ 0 & \sigma_B^2 & 0 & 0 \\ 0 & 0 & \sigma_{AB}^2 & 0 \\ 0 & 0 & 0 & \sigma_{AB}^2 \end{bmatrix}$
Compound Symmetry	CS	$\begin{bmatrix} \sigma + \phi & \sigma & \sigma & \sigma \\ \sigma & \sigma + \phi & \sigma & \sigma \\ \sigma & \sigma & \sigma + \phi & \sigma \\ \sigma & \sigma & \sigma & \sigma + \phi \end{bmatrix}$
Unstructured	UN	$\begin{bmatrix} \sigma_1^2 & \sigma_{21} & \sigma_{31} & \sigma_{41} \\ \sigma_{21} & \sigma_2^2 & \sigma_{32} & \sigma_{42} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 & \sigma_{43} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_4^2 \end{bmatrix}$
Banded Main Diagonal	UN(1)	$\begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{bmatrix}$
First-Order Autoregressive	AR(1)	$\sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$
Toeplitz	TOEP	$\begin{bmatrix} \sigma^2 & \sigma_1 & \sigma_2 & \sigma_3 \\ \sigma_1 & \sigma^2 & \sigma_1 & \sigma_2 \\ \sigma_2 & \sigma_1 & \sigma^2 & \sigma_1 \\ \sigma_3 & \sigma_2 & \sigma_1 & \sigma^2 \end{bmatrix}$
Toeplitz with Two Bands	TOEP(2)	$\begin{bmatrix} \sigma^2 & \sigma_1 & 0 & 0 \\ \sigma_1 & \sigma^2 & \sigma_1 & 0 \\ 0 & \sigma_1 & \sigma^2 & \sigma_1 \\ 0 & 0 & \sigma_1 & \sigma^2 \end{bmatrix}$
Spatial Power	SP(POW)(c)	$\sigma^2 \begin{bmatrix} 1 & \rho^{d_{12}} & \rho^{d_{13}} & \rho^{d_{14}} \\ \rho^{d_{21}} & 1 & \rho^{d_{23}} & \rho^{d_{24}} \\ \rho^{d_{31}} & \rho^{d_{32}} & 1 & \rho^{d_{34}} \\ \rho^{d_{41}} & \rho^{d_{42}} & \rho^{d_{43}} & 1 \end{bmatrix}$

Table 9. (continued)

Description	Structure	Example
First-Order Autoregressive Moving-Average	ARMA(1,1)	$\sigma^2 \begin{bmatrix} 1 & \gamma & \gamma\rho & \gamma\rho^2 \\ \gamma & 1 & \gamma & \gamma\rho \\ \gamma\rho & \gamma & 1 & \gamma \\ \gamma\rho^2 & \gamma\rho & \gamma & 1 \end{bmatrix}$
Unstructured Correlations	UNR	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_{21} & \sigma_1\sigma_3\rho_{31} & \sigma_1\sigma_4\rho_{41} \\ \sigma_2\sigma_1\rho_{21} & \sigma_2^2 & \sigma_2\sigma_3\rho_{32} & \sigma_2\sigma_4\rho_{42} \\ \sigma_3\sigma_1\rho_{31} & \sigma_3\sigma_2\rho_{32} & \sigma_3^2 & \sigma_3\sigma_4\rho_{43} \\ \sigma_4\sigma_1\rho_{41} & \sigma_4\sigma_2\rho_{42} & \sigma_4\sigma_3\rho_{43} & \sigma_4^2 \end{bmatrix}$

**V**<=*value-list*>

requests that blocks of the estimated marginal variance-covariance matrix  $\mathbf{V}(\hat{\boldsymbol{\theta}})$  be displayed in generalized linear mixed models. This matrix is based on the last linearization as described in the section “The Pseudo-Model” on page 120. You can use the *value-list* to select the subjects for which the matrix is displayed. If *value-list* is not specified, the  $\mathbf{V}$  matrix for the first subject is chosen.

Note that the *value-list* refers to subjects as the processing units in the “Dimensions” table. For example, the statements

```
proc glimmix;
  class A B;
  model y = B;
  random int / subject=A;
  random int / subject=A*B v=2;
run;
```

request that the estimated marginal variance matrix for the second subject be displayed. The subject effect for processing in this case is the **A** effect, since it is contained in the **A\*B** interaction. If there is only a single subject as per the “Dimensions” table, then the **V** option displays an  $(n \times n)$  matrix.

See the section “Processing by Subjects” on page 129 for how the GLIMMIX procedure determines the number of subjects in the “Dimensions” table.

The GLIMMIX procedure displays blanks for values that are 0.

**VC**<=*value-list*>

displays the lower-triangular Cholesky root of the blocks of the estimated  $\mathbf{V}(\hat{\boldsymbol{\theta}})$  matrix. See the **V** option for the specification of *value-list*.

**VCI**<=*value-list*>

displays the inverse Cholesky root of the blocks of the estimated  $\mathbf{V}(\hat{\boldsymbol{\theta}})$  matrix. See the **V** option for the specification of *value-list*.

**VCORR**<=*value-list*>

displays the correlation matrix corresponding to the blocks of the estimated  $\mathbf{V}(\hat{\boldsymbol{\theta}})$  matrix. See the **V** option for the specification of *value-list*.

**VI**<=*value-list*>

displays the inverse of the blocks of the estimated  $\mathbf{V}(\hat{\boldsymbol{\theta}})$  matrix. See the **V** option for the specification of *value-list*.

---

## WEIGHT Statement

**WEIGHT** *variable* ;

The **WEIGHT** statement replaces  $\mathbf{R}$  with  $\mathbf{W}^{-1/2}\mathbf{R}\mathbf{W}^{-1/2}$ , where  $\mathbf{W}$  is a diagonal matrix containing the weights. Observations with nonpositive or missing weights are not included in the resulting PROC GLIMMIX analysis. If a **WEIGHT** statement is not included, all observations used in the analysis are assigned a weight of 1.

---

## Programming Statements

This section lists the programming statements available in PROC GLIMMIX to compute various aspects of the generalized linear mixed model or output quantities. For example, you can compute model effects, weights, frequency, subject, group, and other variables. You can use programming statements to define the mean and variance functions. This section also documents the differences between programming statements in PROC GLIMMIX and programming statements in the DATA step. The syntax of programming statements used in PROC GLIMMIX is identical to that used in the NLMIXED procedure (see Chapter 51 in the *SAS/STAT User's Guide*), and the MODEL procedure (refer to the *SAS/ETS User's Guide*). Most of the programming statements that can be used in the SAS DATA step can also be used in the GLIMMIX procedure. Refer to *SAS Language Reference: Dictionary* for a description of SAS programming statements. 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, the SAS 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; however, the following statement is not supported:

```
do i = 'A', 'B', 'C';
```

- The LAG function is not supported with PROC GLIMMIX.
- The PUT statement, used mostly for program debugging in PROC GLIMMIX, supports only some of the features of the DATA step PUT statement, and it has some features are not available with the DATA step PUT statement:

- The PROC GLIMMIX PUT statement does not support line pointers, factored lists, iteration factors, overprinting, \_INFILE\_, the colon (:) format modifier, or “\$”.

- The PROC GLIMMIX 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 PROC GLIMMIX 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 user-defined link function.

When coding your programming statements, you should avoid defining variables that begin with an underscore (`_`), as they may conflict with internal variables created by PROC GLIMMIX.

## User-Defined Link or Variance Function

### Implied Variance Functions

While link functions are not unique for each distribution (see Table 4 on page 71 for the default link functions), the distribution does determine the variance function  $a(\mu)$ . This function expresses the variance of an observation as a function of the mean, apart from weights, frequencies, and additional scale parameters. The implied variance functions  $a(\mu)$  of the GLIMMIX procedure are shown in Table 10 for the supported distributions. For the binomial distribution,  $n$  denotes the number of trials in the *events/trials* syntax. For the negative binomial distribution,  $k$  denotes the scale parameter. The multiplicative scale parameter  $\phi$  is not included for the other distributions. The last column of the table indicates whether  $\phi$  has a value different from 1.0 for the particular distribution.

**Table 10.** Variance Functions in PROC GLIMMIX

<b>DIST=</b>	<b>Distribution</b>	<b>Variance function</b> $a(\mu)$	$\phi \equiv 1$
BETA	beta	$\mu(1 - \mu)$	No
BINARY	binary	$\mu(1 - \mu)$	Yes
BINOMIAL   BIN   B	binomial	$\mu(1 - \mu)/n$	Yes
EXPONENTIAL   EXPO	exponential	$\mu^2$	Yes
GAMMA   GAM	gamma	$\mu^2$	No
GAUSSIAN   G   NORMAL   N	normal	1	No
GEOMETRIC   GEOM	geometric	$\mu + \mu^2$	Yes
INVGAUSS   IGAUSSIAN   IG	inverse Gaussian	$\mu^3$	No
LOGNORMAL   LOGN	log-normal	1	No
NEGBINOMIAL   NEGBIN   NB	negative binomial	$\mu + k\mu^2$	Yes
POISSON   POI   P	Poisson	$\mu$	Yes
TCENTRAL   TDIST   T	$t$	1	No

To change the variance function, you can use SAS programming statements and the predefined automatic variables, as outlined in the following section. Your definition of a variance function will override the DIST= option and its implied variance function. This has the following implication for parameter estimation with the GLIMMIX procedure. When a user-defined link is available, the distribution of the data is determined from the DIST= option, or the respective default for the type of response. In a GLM, for example, this enables maximum likelihood estimation. If a user-defined variance function is provided, the DIST= option is not honored and the distribution of the data is assumed unknown. In a GLM framework, only quasi-likelihood estimation is then available to estimate the model parameters.

## Automatic Variables

To specify your own link or variance function you can use SAS programming statements and draw on the following automatic variables:

**\_LINP\_** is the current value of the linear predictor. It either equals  $\hat{\eta} = \mathbf{x}'\hat{\beta} + \mathbf{z}'\hat{\gamma} + o$  or  $\hat{\eta} = \mathbf{x}'\hat{\beta} + o$ , where  $o$  is the value of the offset variable, or 0 if no offset is specified. The estimated BLUPs  $\hat{\gamma}$  are used in the calculation of the linear predictor during the model fitting phase if a linearization expands about the current values of  $\gamma$ . During the computation of output statistics, the EBLUPs are used if statistics depend on them. For example, the statements

```
proc glimmix;
  model y = x / dist=binary;
  random int / subject=b;
  p = 1/(1+exp(-_linp_));
  output out=glimmixout;
  id p;
run;
```

add the variable `p` to the output data set `glimmixout`. Because no output statistics are requested in the `OUTPUT` statement that depend on the BLUPs, the value of `_LINP_` in this example equals  $\mathbf{x}'\hat{\beta}$ . On the contrary, the statements

```
proc glimmix;
  model y = x / dist=binary;
  random int / subject=b;
  p = 1/(1+exp(-_linp_));
  output out=glimmixout resid(blup)=r;
  id p;
run;
```

also request conditional residuals on the logistic scale. The value of `_LINP_` when computing the variable `p` is  $\mathbf{x}'\hat{\beta} + \mathbf{z}'\hat{\gamma}$ . To ensure that computed statistics are formed from  $\mathbf{x}'\hat{\beta}$  and  $\mathbf{z}'\hat{\gamma}$  terms as needed, it is recommended to use the automatic variables `_XBETA_` and `_ZGAMMA_` instead of `_LINP_`.

**\_MU\_** expresses the mean of an observation as a function of the linear predictor,  $\hat{\mu} = g^{-1}(\hat{\eta})$ .

**\_N\_** is the observation number in the sequence of the data read.

**\_VARIANCE\_** is the estimate of the variance function,  $a(\hat{\mu})$ .

**\_XBETA\_** equals  $\mathbf{x}'\hat{\beta}$ .

**\_ZGAMMA\_** equals  $\mathbf{z}'\hat{\gamma}$ .

The automatic variable `_N_` is incremented whenever the procedure reads an observation from the data set. Observations that are not used in the analysis, for example, because of missing values or invalid weights, are counted. The counter is reset to

1 at the start of every new BY group. Only in some circumstances will `_N_` equal the actual observation number. The symbol should thus be used sparingly to avoid unexpected results.

You must observe the following syntax rules when using the automatic variables. The `_LINF_` symbol cannot appear on the left-hand-side of programming statements; you cannot make an assignment to the `_LINF_` variable. The value of the linear predictor is controlled by the `CLASS`, `MODEL`, and `RANDOM` statements as well as the current parameter estimates and solutions. You can, however, use the `_LINF_` variable on the right-hand side of other operations. Suppose, for example, that you wish to transform the linear predictor prior to applying the inverse log link. The following statements are not valid because the linear predictor is part of an assignment.

```
proc glimmix;
  _linp_ = sqrt(abs(_linp_));
  _mu_   = exp(_linp_);
  model count = logtstd / dist=poisson;
run;
```

The next statements achieve the desired result.

```
proc glimmix;
  _mu_ = exp(sqrt(abs(_linp_)));
  model count = logtstd / dist=poisson;
run;
```

If the value of the linear predictor is altered in any way through programming statements, you need to ensure that an assignment to `_MU_` follows. The assignment to variable `P` in the next set of GLIMMIX statements is without effect.

```
proc glimmix;
  p = _linp_ + rannor(454);
  model count = logtstd / dist=poisson;
run;
```

A user-defined link function is implied by expressing `_MU_` as a function of `_LINF_`. That is, if  $\mu = g^{-1}(\eta)$ , you are providing an expression for the inverse link function with programming statements. It is neither necessary nor possible to give an expression for the inverse operation,  $\eta = g(\mu)$ . The variance function is determined by expressing `_VARIANCE_` as a function of `_MU_`. If the `_MU_` variable appears in an assignment statement inside PROC GLIMMIX, the `LINK=` option of the `MODEL` statement is ignored. If the `_VARIANCE_` function appears in an assignment statement, the `DIST=` option is ignored. Furthermore, the associated variance function per Table 10 is not honored. In short, user-defined expressions take precedence over built-in defaults.

If you specify your own link and variance function, the assignment to `_MU_` must precede an assignment to the variable `_VARIANCE_`.



The following two sets of GLIMMIX statements yield the same parameter estimates, but the models differ statistically.

```
proc glimmix;
  class block entry;
  model y/n = block entry / dist=binomial link=logit;
run;

proc glimmix;
  class block entry;
  prob = 1 / (1+exp(- _linp_));
  _mu_ = n * prob ;
  _variance_ = n * prob *(1-prob);
  model y = block entry;
run;
```

The first GLIMMIX invocation models the proportion  $y/n$  as a binomial proportion with a logit link. The `DIST=` and `LINK=` options are superfluous in this case, since the GLIMMIX procedure defaults to the binomial distribution in light of the `events/trials` syntax. The logit link is that distribution's default link. The second set of GLIMMIX statements models the count variable  $y$  and takes the binomial sample size into account through assignments to the mean and variance function. In contrast to the first set of GLIMMIX statements, the distribution of  $y$  is unknown. Only its mean and variance are known. The model parameters are estimated by maximum likelihood in the first case and by quasi-likelihood in the second case.

---

## Details

---

### Generalized Linear Models Theory

A generalized linear model consists of

- a linear predictor  $\eta = \mathbf{x}'\boldsymbol{\beta}$
- a monotonic mapping between the mean of the data and the linear predictor
- a response distribution in the exponential family of distributions

A density or mass function in this family can be written as

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

for some functions  $b(\cdot)$  and  $c(\cdot)$ . The parameter  $\theta$  is called the natural (canonical) parameter. The parameter  $\phi$  is a scale parameter and it is not present in all exponential family distributions. See [Table 10](#) on page 110 for a list of distributions for which  $\phi \equiv 1$ . In the case where observations are weighted, the scale parameter is replaced with  $\phi/w$  in the above density (or mass function), where  $w$  is the weight associated with the observation  $y$ .

The mean and variance of the data are related to the components of the density,  $E[Y] = \mu = b'(\theta)$ ,  $\text{var}[Y] = \phi b''(\theta)$ , where primes denote first and second derivatives. If you express  $\theta$  as a function of  $\mu$ , the relationship is known as the natural link or the canonical link function. In other words, modeling data with a canonical link assumes that  $\theta = \mathbf{x}'\boldsymbol{\beta}$ ; the effect contributions are additive on the canonical scale. The second derivative of  $b(\cdot)$ , expressed as a function of  $\mu$ , is the variance function of the generalized linear model,  $a(\mu) = b''(\theta(\mu))$ . Note that because of this relationship, the distribution determines the variance function and the canonical link function. You cannot, however, proceed in the opposite direction. If you provide a user-specified variance function, the GLIMMIX procedure assumes that only the first two moments of the response distribution are known. The full distribution of the data is then unknown and maximum likelihood estimation is not possible. Instead, the GLIMMIX procedure then estimates parameters by quasi-likelihood.

### Maximum Likelihood

The GLIMMIX procedure forms the log likelihoods of generalized linear models as

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

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

- Beta :

$$l(\mu_i, \phi; y_i, w_i) = \log \left\{ \frac{\Gamma(\phi/w_i)}{\Gamma(\mu\phi/w_i)\Gamma((1-\mu)\phi/w_i)} \right\} \\ + (\mu\phi/w_i - 1) \log\{y_i\} \\ + ((1-\mu)\phi/w_i - 1) \log\{1 - y_i\}$$

$\text{var}[Y] = \mu(1-\mu)/(1+\phi)$ ,  $\phi > 0$ . See Ferrari and Cribari-Neto (2004).

- Binary:

$$l(\mu_i, \phi; y_i, w_i) = w_i(y_i \log\{\mu_i\} + (1 - y_i) \log\{1 - \mu_i\})$$

$\text{var}[Y] = \mu(1 - \mu)$ ,  $\phi \equiv 1$ .

- Binomial:

$$l(\mu_i, \phi; y_i, w_i) = w_i(y_i \log\{\mu_i\} + (n_i - y_i) \log\{1 - \mu_i\}) \\ + w_i(\log\{n_i!\} - \log\{y_i!\} - \log\{(n_i - y_i)!\})$$

where  $y_i$  and  $n_i$  are the *events* and *trials* in the *events/trials* syntax, and  $0 < \mu < 1$ .  $\text{var}[y/n] = \mu(1 - \mu)/n$ ,  $\phi \equiv 1$ .

- Exponential:

$$l(\mu_i, \phi; y_i, w_i) = \begin{cases} -\log\{\mu_i\} - y_i/\mu_i & w_i = 1 \\ w_i \log\left\{\frac{w_i y_i}{\mu_i}\right\} - \frac{w_i y_i}{\mu_i} - \log\{y_i \Gamma(w_i)\} & w_i \neq 1 \end{cases}$$

$$\text{var}[Y] = \mu^2, \phi \equiv 1.$$

- Gamma:

$$l(\mu_i, \phi; y_i, w_i) = w_i \phi \log\left\{\frac{w_i y_i \phi}{\mu_i}\right\} - \frac{w_i y_i \phi}{\mu_i} - \log\{y_i\} - \log\{\Gamma(w_i \phi)\}$$

$$\text{var}[Y] = \phi \mu^2, \phi > 0.$$

- Geometric:

$$\begin{aligned} l(\mu_i, \phi; y_i, w_i) &= y_i \log\left\{\frac{\mu_i}{w_i}\right\} - (y_i + w_i) \log\left\{1 + \frac{\mu_i}{w_i}\right\} \\ &+ \log\left\{\frac{\Gamma(y_i + w_i)}{\Gamma(w_i)\Gamma(y_i + 1)}\right\} \end{aligned}$$

$$\text{var}[Y] = \mu + \mu^2, \phi \equiv 1.$$

- Inverse Gaussian:

$$l(\mu_i, \phi; y_i, w_i) = -\frac{1}{2} \left[ \frac{w_i(y_i - \mu_i)^2}{y_i \phi \mu_i^2} + \log\left\{\frac{\phi y_i^3}{w_i}\right\} + \log\{2\pi\} \right]$$

$$\text{var}[Y] = \phi \mu^3, \phi > 0.$$

- “Lognormal”:

$$l(\mu_i, \phi; \log\{y_i\}, w_i) = -\frac{1}{2} \left[ \frac{w_i(\log\{y_i\} - \mu_i)^2}{\phi} + \log\left\{\frac{\phi}{w_i}\right\} + \log\{2\pi\} \right]$$

$$\text{var}[\log\{Y\}] = \phi, \phi > 0.$$

If you specify `DIST=LOGNORMAL` with response variable `Y`, the `GLIMMIX` procedure assumes that  $\log\{Y\} \sim N(\mu, \sigma^2)$ . Note that the above density is not the density of `Y`.

- Multinomial:

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

$$\phi \equiv 1.$$

- Negative Binomial:

$$\begin{aligned} l(\mu_i, \phi; y_i, w_i) &= y_i \log\left\{\frac{k\mu_i}{w_i}\right\} - (y_i + w_i/k) \log\left\{1 + \frac{k\mu_i}{w_i}\right\} \\ &+ \log\left\{\frac{\Gamma(y_i + w_i/k)}{\Gamma(w_i/k)\Gamma(y_i + 1)}\right\} \end{aligned}$$

$$\text{var}[Y] = \mu + k\mu^2, k > 0, \phi \equiv 1.$$

The negative binomial distribution is a member of the two-parameter exponential family. The parameter  $k$  is related to the scale of the data, since it is part of the variance function. However, it cannot be factored from the variance, as is the case with the  $\phi$  parameter in many other distributions. The parameter  $k$  is designated as “Scale” in the “Parameter Estimates” table of the GLIMMIX procedure.

- Normal (Gaussian):

$$l(\mu_i, \phi; y_i, w_i) = -\frac{1}{2} \left[ \frac{w_i(y_i - \mu_i)^2}{\phi} + \log \left\{ \frac{\phi}{w_i} \right\} + \log\{2\pi\} \right]$$

$$\text{var}[Y] = \phi, \phi > 0.$$

- Poisson:

$$l(\mu_i, \phi; y_i, w_i) = w_i(y_i \log\{\mu_i\} - \mu_i - \log\{y_i!\})$$

$$\text{var}[Y] = \mu, \phi \equiv 1.$$

- Shifted T:

$$z_i = -\log\{\phi^*/\sqrt{w_i}\} + \log\{\Gamma(0.5(\nu + 1))\} \\ - \log\{\Gamma(0.5\nu)\} - 0.5 * \log\{\pi\nu\}$$

$$l(\mu_i, \phi; y_i, w_i) = -(\nu/2 + 0.5) \log \left\{ 1 + w_i/\nu \left( \frac{y_i - \mu_i}{\phi^*} \right)^2 \right\} + z_i$$

$$\phi^* = \phi\sqrt{(\nu - 2)/\nu}, \phi > 0, \nu > 0. \text{ var}[Y] = (\phi^*)^2(\nu - 2)/\nu.$$

Define the parameter vector for the generalized linear model as  $\boldsymbol{\theta} = \boldsymbol{\beta}$  if  $\phi \equiv 1$ , and as  $\boldsymbol{\theta} = [\boldsymbol{\beta}', \phi]'$  otherwise.  $\boldsymbol{\beta}$  denotes the fixed effects parameters in the linear predictor. For the Negative Binomial distribution the relevant parameter vector is  $\boldsymbol{\theta} = [\boldsymbol{\beta}', k]'$ . The gradient and Hessian of the log likelihood are then

$$\mathbf{g} = \frac{\partial L(\boldsymbol{\theta}; \mathbf{y})}{\partial \boldsymbol{\theta}}$$

$$\mathbf{H} = \frac{\partial^2 L(\boldsymbol{\theta}; \mathbf{y})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'}$$

The GLIMMIX procedure computes the gradient vector and Hessian matrix analytically, unless your programming statements involve functions whose derivatives are determined by finite differences. If the procedure is in scoring mode,  $\mathbf{H}$  is replaced by its expected value. PROC GLIMMIX is in scoring mode when the number  $n$  of SCORING= $n$  iterations has not been exceeded and the optimization technique uses second derivatives, or when the Hessian is computed at convergence and the EXPHESSIAN option is in effect.

In models for independent data with known distribution, parameter estimates are obtained by the method of maximum likelihood. No parameters are profiled from the optimization. The default optimization technique for GLMs is the Newton-Raphson algorithm, except for Gaussian models with identity link, which do not require iterative model fitting. In the case of a Gaussian model, the scale parameter is estimated by restricted maximum likelihood, because this estimate is unbiased. The results from PROC GLIMMIX agree with those of the GLM and REG procedure for such models. You can obtain the maximum likelihood estimate of the scale parameter with the `NOREML` option of the `PROC GLIMMIX` statement. To change the optimization algorithm, use the `TECHNIQUE=` option of the `NLOPTIONS` statement.

Standard errors of the parameter estimates are obtained from the negative of the inverse of the (observed or expected) second derivative matrix  $\mathbf{H}$ .

### Quasi-Likelihood for Independent Data

Quasi-likelihood estimation uses only the first and second moment of the response. In the case of independent data, this requires only a specification of the mean and variance of your data. The GLIMMIX procedure estimates parameters by quasi-likelihood, if the following conditions are met:

- The response distribution is unknown, because of a user-specified variance function.
- There are no G-side random effects.
- There are no R-side covariance structures or at most an overdispersion parameter.

Under some mild regularity conditions, the function

$$Q(\mu_i, y_i) = \int_{y_i}^{\mu_i} \frac{y_i - t}{\phi a(\mu_i)} dt$$

known as the log quasi-likelihood of the  $i$ th observation, has some properties of a log likelihood function (McCullagh and Nelder 1989, p. 325). For example, the expected value of its derivative is zero, and the variance of its derivative equals the negative of the expected value of the second derivative. Consequently,

$$QL(\boldsymbol{\mu}, \phi, \mathbf{y}) = \sum_{i=1}^n f_i w_i \frac{Y_i - \mu_i}{\phi a(\mu_i)}$$

can serve as the score function for estimation. Quasi-likelihood estimation takes as the gradient and “Hessian” matrix—with respect to the fixed effects parameters  $\boldsymbol{\beta}$ —the quantities

$$\begin{aligned} \mathbf{g}_{ql} &= [g_{ql,j}] = \left[ \frac{\partial QL(\boldsymbol{\mu}, \phi, \mathbf{y})}{\partial \beta_j} \right] = \mathbf{D}'\mathbf{V}^{-1}(\mathbf{Y} - \boldsymbol{\mu})/\phi \\ \mathbf{H}_{ql} &= [h_{ql,jk}] = \left[ \frac{\partial^2 QL(\boldsymbol{\mu}, \phi, \mathbf{y})}{\partial \beta_j \partial \beta_k} \right] = \mathbf{D}'\mathbf{V}^{-1}\mathbf{D}/\phi \end{aligned}$$

In this expression  $\mathbf{D}$  is a matrix of derivatives of  $\boldsymbol{\mu}$  with respect to the elements in  $\boldsymbol{\beta}$ , and  $\mathbf{V}$  is a diagonal matrix containing variance functions,  $\mathbf{V} = [a(\mu_1), \dots, a(\mu_n)]$ . Notice that  $\mathbf{H}_{ql}$  is not the second derivative matrix of  $Q(\boldsymbol{\mu}, \mathbf{y})$ . Rather, it is the negative of the expected value of  $\partial \mathbf{g}_{ql} / \partial \boldsymbol{\beta}$ .  $\mathbf{H}_{ql}$  thus has the form of a “scoring Hessian.”

The GLIMMIX procedure fixes the scale parameter  $\phi$  at 1.0 by default. To estimate the parameter, add the

```
random _residual_;
```

statement. The resulting estimator (McCullagh and Nelder 1989, p. 328) is

$$\hat{\phi} = \frac{1}{m} \sum_{i=1}^n f_i w_i \frac{y_i - \hat{\mu}_i}{a(\hat{\mu}_i)}$$

where  $m = f - \text{rank}\{\mathbf{X}\}$  if the **NOREML** option is in effect,  $m = f$ , otherwise, and  $f$  is the sum of the frequencies.

See [Example 4](#) for an application of quasi-likelihood estimation with PROC GLIMMIX.

### Effects of Adding Overdispersion

You can add a multiplicative overdispersion parameter to a generalized linear model in the GLIMMIX procedure with the

```
random _residual_;
```

statement. For models in which  $\phi \equiv 1$ , this effectively lifts the constraint of the parameter. In models that already contain a  $\phi$  or  $k$  scale parameter—such as, the normal, gamma, or negative binomial models—the statement adds a multiplicative scalar (the overdispersion parameter,  $\phi_o$ ) to the variance function.

The overdispersion parameter is estimated from Pearson’s statistic after all other parameters have been determined by (restricted) maximum likelihood or quasi-likelihood. This estimate is

$$\hat{\phi}_o = \frac{1}{\phi^p m} \sum_{i=1}^n f_i w_i \frac{(y_i - \mu_i)^2}{a(\mu_i)}$$

where  $m = f - \text{rank}\{\mathbf{X}\}$  if the **NOREML** option is in effect, and  $m = f$ , otherwise, and  $f$  is the sum of the frequencies. The power  $p$  is  $-1$  for the Gamma distribution and 1 otherwise.

Adding an overdispersion parameter does not alter any of the other parameter estimates. It only changes the variance-covariance matrix of the estimates by a certain factor. If overdispersion arises from correlations among the observations, then you should investigate more complex random effects structures.

## Generalized Linear Mixed Models Theory

### *Model or Integral Approximation*

In a generalized linear model, the log likelihood is well defined and an objective function for estimation of the parameters is simple to construct based on the independence of the data. In a GLMM, several problems must be overcome before an objective function can be computed.

- The model may be vacuous in the sense that no valid joint distribution can be constructed either in general, or for a particular set of parameter values. For example, if  $\mathbf{Y}$  is an equicorrelated ( $n \times 1$ ) vector of binary responses with the same success probability and a symmetric distribution, then the lower bound on the correlation parameter depends on  $n$  and  $\pi$  (Gilliland and Schabenberger 2001). If further restrictions are placed on the joint distribution, as in Bahadur (1961), the correlation is also restricted from above.
- The dependency between mean and variance for nonnormal data places constraints on the possible correlation models that simultaneously yield valid joint distributions and a desired conditional distributions. Thus, for example, aspiring for conditional Poisson variates that are marginally correlated according to a spherical spatial process may not be possible.
- Even if the joint distribution is feasible mathematically, it still may be out of reach computationally. When data are independent, conditional on the random effects, the marginal log likelihood can in principle be constructed by integrating out the random effects from the joint distribution. However, numerical integration is practical only when the number of random effects is small.

Because of these special features of generalized linear mixed models, many estimation methods have been put forth in the literature. The two basic approaches are one, to approximate the objective function, and two to approximate the model. Algorithms in the second category can be expressed in terms of Taylor series (linearizations) and are hence also known as linearization methods. They employ expansions to approximate the model by one based on pseudo-data with fewer nonlinear components. The process of computing the linear approximation must be repeated several times until some criterion indicates lack of further progress. Schabenberger and Gregoire (1996) list numerous algorithms based on Taylor series for the case of clustered data alone. The fitting methods based on linearizations are usually doubly iterative. The generalized linear mixed model is approximated by a linear mixed model based on current values of the covariance parameter estimates. The resulting linear mixed model is then fit, which is itself an iterative process. On convergence, the new parameter estimates are used to update the linearization, which results in a new linear mixed model. The process stops when parameter estimates between successive linear mixed model fits change within a specified tolerance only.

Integral approximation methods approximate the log likelihood of the GLMM and submit the approximated function to numerical optimization. Various techniques are used to compute the approximation: Laplace methods, quadrature methods, Monte

Carlo integration, and Markov Chain Monte Carlo methods. The advantage of integral approximation methods is to provide an actual objective function for optimization. This enables you to perform likelihood ratio tests among nested models and to compute likelihood-based fit statistics. The estimation process is singly iterative. The disadvantage of integral approximation methods is the difficulty of accommodating crossed random effects, multiple subject effects, and complex R-side covariance structures. The number of random effects should be small for integral approximation methods to be practically feasible.

The advantages of linearization-based methods include a relatively simple form of the linearized model that typically can be fit based on only the mean and variance in the linearized form. Models for which the joint distribution is difficult—or impossible—to ascertain can be fit with linearization-based approaches. Models with correlated errors, a large number of random effects, crossed random effects, and multiple types of subjects are thus excellent candidates for linearization methods. The disadvantages of this approach include the absence of a true objective function for the overall optimization process and potentially biased estimates of the covariance parameters, especially for binary data. The objective function to be optimized after each linearization update is dependent on the current pseudo-data. The process can fail at both levels of the double iteration scheme.

This version of the GLIMMIX procedure fits generalized linear mixed models based on linearizations. The default estimation method in GLIMMIX for models containing random effects is a technique known as restricted pseudo-likelihood (RPL) (Wolfinger and O’Connell 1993) estimation with an expansion around the current estimate of the best linear unbiased predictors of the random effects (METHOD=RSPL).

### **Pseudo-Likelihood Estimation Based on Linearization**

#### **The Pseudo-Model**

Recall from the “Notation for the Generalized Linear Mixed Model” section (beginning on page 7) that

$$E[\mathbf{Y}|\boldsymbol{\gamma}] = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}) = g^{-1}(\boldsymbol{\eta}) = \boldsymbol{\mu}$$

where  $\boldsymbol{\gamma} \sim N(\mathbf{0}, \mathbf{G})$  and  $\text{var}[\mathbf{Y}|\boldsymbol{\gamma}] = \mathbf{A}^{1/2}\mathbf{R}\mathbf{A}^{1/2}$ . Following Wolfinger and O’Connell (1993), a first-order Taylor series of  $\boldsymbol{\mu}$  about  $\tilde{\boldsymbol{\beta}}$  and  $\tilde{\boldsymbol{\gamma}}$  yields

$$g^{-1}(\boldsymbol{\eta}) \doteq g^{-1}(\tilde{\boldsymbol{\eta}}) + \tilde{\boldsymbol{\Delta}}\mathbf{X}(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) + \tilde{\boldsymbol{\Delta}}\mathbf{Z}(\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}})$$

where

$$\tilde{\boldsymbol{\Delta}} = \left( \frac{\partial g^{-1}(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \right)_{\tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\gamma}}}$$

is a diagonal matrix of derivatives of the conditional mean evaluated at the expansion locus. Rearranging terms yields the expression

$$\tilde{\boldsymbol{\Delta}}^{-1}(\boldsymbol{\mu} - g^{-1}(\tilde{\boldsymbol{\eta}})) + \mathbf{X}\tilde{\boldsymbol{\beta}} + \mathbf{Z}\tilde{\boldsymbol{\gamma}} \doteq \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$$



The left-hand side is the expected value, conditional on  $\gamma$ , of

$$\tilde{\mathbf{\Delta}}^{-1}(\mathbf{Y} - g^{-1}(\tilde{\boldsymbol{\eta}})) + \mathbf{X}\tilde{\boldsymbol{\beta}} + \mathbf{Z}\tilde{\boldsymbol{\gamma}} \equiv \mathbf{P}$$

and

$$\text{var}[\mathbf{P}|\boldsymbol{\gamma}] = \tilde{\mathbf{\Delta}}^{-1} \mathbf{A}^{1/2} \mathbf{R} \mathbf{A}^{1/2} \tilde{\mathbf{\Delta}}^{-1}$$

You can thus consider the model

$$\mathbf{P} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

which is a linear mixed model with pseudo-response  $\mathbf{P}$ , fixed effects  $\boldsymbol{\beta}$ , random effects  $\boldsymbol{\gamma}$ , and  $\text{var}[\boldsymbol{\epsilon}] = \text{var}[\mathbf{P}|\boldsymbol{\gamma}]$ .

### Objective Functions

Now define

$$\mathbf{V}(\boldsymbol{\theta}) = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \tilde{\mathbf{\Delta}}^{-1} \mathbf{A}^{1/2} \mathbf{R} \mathbf{A}^{1/2} \tilde{\mathbf{\Delta}}^{-1}$$

as the marginal variance in the linear mixed pseudo-model, where  $\boldsymbol{\theta}$  is the  $(q \times 1)$  parameter vector containing all unknowns in  $\mathbf{G}$  and  $\mathbf{R}$ . Based on this linearized model, an objective function can be defined, assuming that the distribution of  $\mathbf{P}$  is known. The GLIMMIX procedure assumes that  $\boldsymbol{\epsilon}$  has a normal distribution. The maximum log pseudo-likelihood (MxPL) and restricted log pseudo-likelihood (RxPL) for  $\mathbf{P}$  are then

$$\begin{aligned} l(\boldsymbol{\theta}, \mathbf{p}) &= -\frac{1}{2} \log |\mathbf{V}(\boldsymbol{\theta})| - \frac{1}{2} \mathbf{r}' \mathbf{V}(\boldsymbol{\theta})^{-1} \mathbf{r} - \frac{f}{2} \log \{2\pi\} \\ l_R(\boldsymbol{\theta}, \mathbf{p}) &= -\frac{1}{2} \log |\mathbf{V}(\boldsymbol{\theta})| - \frac{1}{2} \mathbf{r}' \mathbf{V}(\boldsymbol{\theta})^{-1} \mathbf{r} \\ &\quad - \frac{1}{2} \log |\mathbf{X}' \mathbf{V}(\boldsymbol{\theta})^{-1} \mathbf{X}| - \frac{f-k}{2} \log \{2\pi\} \end{aligned}$$

with  $\mathbf{r} = \mathbf{p} - \mathbf{X}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\mathbf{p}$ .  $f$  denotes the sum of the frequencies used in the analysis, and  $k$  denotes the rank of  $\mathbf{X}$ . The fixed effects parameters  $\boldsymbol{\beta}$  are profiled from these expressions. The parameters in  $\boldsymbol{\theta}$  are estimated by the optimization techniques specified in the **NLOPTIONS** statement. The objective function for minimization is  $-2l(\boldsymbol{\theta}, \mathbf{p})$  or  $-2l_R(\boldsymbol{\theta}, \mathbf{p})$ . At convergence, the profiled parameters are estimated and the random effects are predicted as

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\mathbf{X}'\mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{p} \\ \hat{\boldsymbol{\gamma}} &= \hat{\mathbf{G}}\mathbf{Z}'\mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\hat{\mathbf{r}} \end{aligned}$$

With these statistics, the pseudo-response and error weights of the linearized model are recomputed and the objective function is minimized again. The predictors  $\hat{\boldsymbol{\gamma}}$  are

the estimated BLUPs in the approximated linear model. This process continues until the relative change between parameter estimates at two successive (outer) iterations is sufficiently small. See the `PCONV=` option of the PROC GLIMMIX statement for the computational details on how the GLIMMIX procedure compares parameter estimates across optimizations.

If the conditional distribution contains a scale parameter  $\phi \neq 1$  (Table 10 on page 110), the GLIMMIX procedure profiles this parameter in GLMMs from the log pseudo-likelihoods as well. To this end define

$$\mathbf{V}(\boldsymbol{\theta}^*) = \tilde{\boldsymbol{\Delta}}^{-1} \mathbf{A}^{1/2} \mathbf{R}^* \mathbf{A}^{1/2} \tilde{\boldsymbol{\Delta}}^{-1} + \mathbf{Z} \mathbf{G}^* \mathbf{Z}'$$

where  $\boldsymbol{\theta}^*$  is the covariance parameter vector with  $q - 1$  elements. The matrices  $\mathbf{G}^*$  and  $\mathbf{R}^*$  are appropriately reparameterized versions of  $\mathbf{G}$  and  $\mathbf{R}$ . For example, if  $\mathbf{G}$  has a variance component structure and  $\mathbf{R} = \phi \mathbf{I}$ , then  $\boldsymbol{\theta}^*$  contains ratios of the variance components and  $\phi$ , and  $\mathbf{R}^* = \mathbf{I}$ . The solution for  $\hat{\phi}$  is

$$\hat{\phi} = \hat{\mathbf{r}}' \mathbf{V}(\hat{\boldsymbol{\theta}}^*)^{-1} \hat{\mathbf{r}} / m$$

where  $m = f$  for MxPL and  $m = f - k$  for RxPL. Substitution into the previous functions yields the profiled log pseudo-likelihoods,

$$\begin{aligned} l(\boldsymbol{\theta}^*, \mathbf{p}) &= -\frac{1}{2} \log |\mathbf{V}(\boldsymbol{\theta}^*)| - \frac{f}{2} \log \{ \mathbf{r}' \mathbf{V}(\boldsymbol{\theta}^*)^{-1} \mathbf{r} \} - \frac{f}{2} (1 + \log \{ 2\pi / f \}) \\ l_R(\boldsymbol{\theta}^*, \mathbf{p}) &= -\frac{1}{2} \log |\mathbf{V}(\boldsymbol{\theta}^*)| - \frac{f-k}{2} \log \{ \mathbf{r}' \mathbf{V}(\boldsymbol{\theta}^*)^{-1} \mathbf{r} \} \\ &\quad - \frac{1}{2} \log |\mathbf{X}' \mathbf{V}(\boldsymbol{\theta}^*)^{-1} \mathbf{X}| - \frac{f-k}{2} (1 + \log \{ 2\pi / (f-k) \}) \end{aligned}$$

Profiling of  $\phi$  can be suppressed with the `NOPROFILE` option of the PROC GLIMMIX statement.

Where possible, the objective function, its gradient, and its Hessian employ the sweep-based *W*-transformation (Hemmerle and Hartley 1973, Goodnight 1979, Goodnight and Hemmerle 1979). Further details about the minimization process in the general linear mixed model can be found in Wolfinger, Tobias, and Sall (1994).

### Estimated Precision of Estimates

The GLIMMIX procedure produces estimates of the variability of  $\hat{\boldsymbol{\beta}}$ ,  $\hat{\boldsymbol{\theta}}$ , and estimates of the prediction variability for  $\hat{\boldsymbol{\gamma}}$ ,  $\text{var}[\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}]$ . Denote as  $\mathbf{S}$  the matrix

$$\mathbf{S} \equiv \widehat{\text{var}}[\mathbf{P}|\boldsymbol{\gamma}] = \tilde{\boldsymbol{\Delta}}^{-1} \mathbf{A}^{1/2} \mathbf{R} \mathbf{A}^{1/2} \tilde{\boldsymbol{\Delta}}^{-1}$$

where all components on the right-hand side are evaluated at the converged estimates. The mixed model equations (Henderson 1984) in the linear mixed (pseudo-)model are then

$$\begin{bmatrix} \mathbf{X}' \mathbf{S}^{-1} \mathbf{X} & \mathbf{X}' \mathbf{S}^{-1} \mathbf{Z} \\ \mathbf{Z}' \mathbf{S}^{-1} \mathbf{X} & \mathbf{Z}' \mathbf{S}^{-1} \mathbf{Z} + \mathbf{G}(\hat{\boldsymbol{\theta}})^{-1} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\gamma}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}' \mathbf{S}^{-1} \mathbf{p} \\ \mathbf{Z}' \mathbf{S}^{-1} \mathbf{p} \end{bmatrix}$$

and

$$\begin{aligned} \mathbf{C} &= \begin{bmatrix} \mathbf{X}'\mathbf{S}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{S}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{S}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{S}^{-1}\mathbf{Z} + \mathbf{G}^{-1} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \hat{\mathbf{\Omega}} & -\hat{\mathbf{\Omega}}\mathbf{X}'\mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{Z}\mathbf{G}(\hat{\boldsymbol{\theta}}) \\ -\mathbf{G}(\hat{\boldsymbol{\theta}})\mathbf{Z}'\mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{X}\hat{\mathbf{\Omega}} & \mathbf{M} + \mathbf{G}(\hat{\boldsymbol{\theta}})\mathbf{Z}'\mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{X}\hat{\mathbf{\Omega}}\mathbf{X}'\mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{Z}\mathbf{G}(\hat{\boldsymbol{\theta}}) \end{bmatrix} \end{aligned}$$

is the approximate estimated variance-covariance matrix of  $[\hat{\boldsymbol{\beta}}', \hat{\boldsymbol{\gamma}}' - \boldsymbol{\gamma}']'$ . Here,  $\hat{\mathbf{\Omega}} = (\mathbf{X}'\mathbf{V}(\hat{\boldsymbol{\theta}})^{-1}\mathbf{X})^{-1}$  and  $\mathbf{M} = (\mathbf{Z}'\mathbf{S}^{-1}\mathbf{Z} + \mathbf{G}(\hat{\boldsymbol{\theta}})^{-1})^{-1}$ .

The square roots of the diagonal elements of  $\hat{\mathbf{\Omega}}$  are reported in the column “Standard Error” of the “Parameter Estimates” table. This table is produced with the **SOLUTION** option in the **MODEL** statement. The prediction standard errors of the random effects solutions are reported in the column “Std Err Pred” of the “Solution for Random Effects” table. This table is produced with the **SOLUTION** option in the **RANDOM** statement.

As a cautionary note,  $\mathbf{C}$  tends to underestimate the true sampling variability of  $[\hat{\boldsymbol{\beta}}', \hat{\boldsymbol{\gamma}}']'$ , because no account is made for the uncertainty in estimating  $\mathbf{G}$  and  $\mathbf{R}$ . Although inflation factors have been proposed (Kackar and Harville 1984; Kass and Steffey 1989; Prasad and Rao 1990), they tend to be small for data sets that are fairly well balanced. PROC GLIMMIX does not compute any inflation factors by default. The DDFM=KENWARDROGER option in the **MODEL** statement prompts PROC GLIMMIX to compute a specific inflation factor (Kenward and Roger 1997), along with Satterthwaite-based degrees of freedom.

If  $\mathbf{G}(\hat{\boldsymbol{\theta}})$  is singular, or if you use the **CHOL** option of the **PROC GLIMMIX** statement, the mixed model equations are modified as follows. Let  $\mathbf{L}$  denote the lower triangular matrix so that  $\mathbf{L}\mathbf{L}' = \mathbf{G}(\hat{\boldsymbol{\theta}})$ . PROC GLIMMIX then solves the equations

$$\begin{bmatrix} \mathbf{X}'\mathbf{S}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{S}^{-1}\mathbf{Z}\mathbf{L} \\ \mathbf{L}'\mathbf{Z}'\mathbf{S}^{-1}\mathbf{X} & \mathbf{L}'\mathbf{Z}'\mathbf{S}^{-1}\mathbf{Z}\mathbf{L} + \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\tau}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{S}^{-1}\mathbf{p} \\ \mathbf{L}'\mathbf{Z}'\mathbf{S}^{-1}\mathbf{p} \end{bmatrix}$$

and transforms  $\hat{\boldsymbol{\tau}}$  and a generalized inverse of the left-hand side coefficient matrix using  $\mathbf{L}$ .

The asymptotic covariance matrix of the covariance parameter estimator  $\hat{\boldsymbol{\theta}}$  is computed based on the observed or expected Hessian matrix of the optimization procedure. Consider first the case where the scale parameter  $\phi$  is not present or not profiled. Since  $\boldsymbol{\beta}$  is profiled from the pseudo-likelihood, the objective function for minimization is  $f(\boldsymbol{\theta}) = -2l(\boldsymbol{\theta}, \mathbf{p})$  for METHOD=MSPL and METHOD=MMPL and  $f(\boldsymbol{\theta}) = -2l_R(\boldsymbol{\theta}, \mathbf{p})$  for METHOD=RSPL and METHOD=RMPL. Denote the observed Hessian (second derivative) matrix as

$$\mathbf{H} = \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'}$$

The GLIMMIX procedure computes the variance of  $\hat{\boldsymbol{\theta}}$  by default as  $2\mathbf{H}^{-1}$ . If the Hessian is not positive definite, a sweep-based generalized inverse is used instead. When the **EXPHESSIAN** option of the **PROC GLIMMIX** statement is used, or when the procedure is in scoring mode at convergence (see the **SCORING** option in the **PROC GLIMMIX** statement), the observed Hessian is replaced with an approximated expected Hessian matrix in these calculations.

Following Wolfinger, Tobias, and Sall (1994), define the following components of the gradient and Hessian in the optimization:

$$\begin{aligned} \mathbf{g}_1 &= \frac{\partial}{\partial \boldsymbol{\theta}} \mathbf{r}'\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{r} \\ \mathbf{H}_1 &= \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \log\{\mathbf{V}(\boldsymbol{\theta})\} \\ \mathbf{H}_2 &= \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \mathbf{r}'\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{r} \\ \mathbf{H}_3 &= \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \log\{|\mathbf{X}'\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{X}|\} \end{aligned}$$

Table 11 gives expressions for the Hessian matrix  $\mathbf{H}$  depending on estimation method, profiling, and scoring.

**Table 11.** Hessian Computation in GLIMMIX

Profiling	Scoring	MxPL	RxPL
No	No	$\mathbf{H}_1 + \mathbf{H}_2$	$\mathbf{H}_1 + \mathbf{H}_2 + \mathbf{H}_3$
No	Yes	$-\mathbf{H}_1$	$-\mathbf{H}_1 + \mathbf{H}_3$
No	Mod.	$-\mathbf{H}_1$	$-\mathbf{H}_1 - \mathbf{H}_3$
Yes	No	$\begin{bmatrix} \mathbf{H}_1 + \mathbf{H}_2/\phi & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & f/\phi^2 \end{bmatrix}$	$\begin{bmatrix} \mathbf{H}_1 + \mathbf{H}_2/\phi + \mathbf{H}_3 & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & (f - k)/\phi^2 \end{bmatrix}$
Yes	Yes	$\begin{bmatrix} -\mathbf{H}_1 & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & f/\phi^2 \end{bmatrix}$	$\begin{bmatrix} -\mathbf{H}_1 + \mathbf{H}_3 & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & (f - k)/\phi^2 \end{bmatrix}$
Yes	Mod.	$\begin{bmatrix} -\mathbf{H}_1 & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & f/\phi^2 \end{bmatrix}$	$\begin{bmatrix} -\mathbf{H}_1 - \mathbf{H}_3 & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & (f - k)/\phi^2 \end{bmatrix}$

The (“Mod.”) expressions for the Hessian under scoring in RxPL estimation refer to a modified scoring method. In some cases, the modification leads to faster convergence than the standard scoring algorithm. The modification is requested with the **SCOREMOD** option in the **PROC GLIMMIX** statement.

Finally, in the case of a profiled scale parameter  $\phi$ , the Hessian for the  $(\boldsymbol{\theta}^*, \phi)$  parameterization is converted into that for the  $\boldsymbol{\theta}$  parameterization as

$$\mathbf{H}(\boldsymbol{\theta}) = \mathbf{B}\mathbf{H}(\boldsymbol{\theta}^*, \phi)\mathbf{B}$$

where

$$\mathbf{B} = \begin{bmatrix} 1/\phi & 0 & \cdots & 0 & 0 \\ 0 & 1/\phi & \cdots & 0 & 0 \\ 0 & \cdots & \cdots & 1/\phi & 0 \\ -\theta_1^*/\phi & -\theta_2^*/\phi & \cdots & -\theta_{q-1}^*/\phi & 1 \end{bmatrix}$$

### Subject-Specific and Population-Averaged (Marginal) Expansions

There are two basic choices for the expansion locus of the linearization. A subject-specific (SS) expansion uses

$$\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}} \quad \tilde{\boldsymbol{\gamma}} = \hat{\boldsymbol{\gamma}}$$

the current estimates of the fixed effects and estimated BLUPs. The population-averaged (PA) expansion expands about the same fixed effects and the expected value of the random effects

$$\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}} \quad \tilde{\boldsymbol{\gamma}} = \mathbf{0}$$

To recompute the pseudo-response and weights in the SS expansion, the BLUPs must be computed every time the objective function in the linear mixed model is maximized. The PA expansion does not require any BLUPs. The four pseudo-likelihood methods implemented in the GLIMMIX procedure are the  $2 \times 2$  factorial combination between two expansion loci and residual versus maximum pseudo-likelihood estimation. The following table shows the combination and the corresponding values of the `METHOD=` option (`PROC GLIMMIX` statement); `METHOD=RSPL` is the default.

Type of PL	Expansion $\hat{\boldsymbol{\gamma}}$	locus $E[\boldsymbol{\gamma}]$
residual	RSPL	RMPL
maximum	MSPL	MMPL

---

## GLM-Mode or GLMM-Mode

The GLIMMIX procedure knows two basic modes of parameter estimation and it can be important for you to understand the differences between the two modes.

In GLM-Mode the data are never correlated and there can be no G-side random effects. Typical examples are logistic regression and normal linear models. When you fit a model in GLM-Mode, the `METHOD=` option in the `PROC GLIMMIX` statement

has no effect. PROC GLIMMIX estimates the parameters of the model by maximum likelihood, (restricted) maximum likelihood, or quasi-likelihood depending on the distributional properties of the model (see “Default Estimation Techniques”). The “Model Information” table informs you which estimation method was applied. In GLM-Mode, the individual observations are considered the sampling units. This has bearing, for example, on how sandwich estimators are computed (see the [EMPIRICAL](#) option and the “Empirical Covariance (“Sandwich”) Estimators” section (page 128).

In GLMM-Mode, the procedure assumes that the model contains random effects or possibly correlated errors, or that the data have a clustered structure. The parameters are then estimated by the techniques specified with the [METHOD=](#) option in the [PROC GLIMMIX](#) statement.

In general, adding one overdispersion parameter to a generalized linear model does not trigger the GLMM-Mode. For example, the model defined by the statements

```
proc glimmix;
  model y = x1 x2 / dist=poisson;
  random _residual_;
run;
```

is fit in GLM-Mode. The parameters of the fixed-effects are estimated by maximum likelihood and the covariance matrix of the fixed-effects parameters is adjusted by the overdispersion parameter.

In a model with uncorrelated data you can trigger the GLMM-Mode by specifying a [SUBJECT=](#) or [GROUP=](#) effect in the [RANDOM](#) statement. For example,

```
proc glimmix;
  class id;
  model y = x1 x2 / dist=poisson;
  random _residual_ / subject=id;
run;
```

fits the model using the residual pseudo-likelihood algorithm.

If in doubt, you can determine whether a model was fit in GLM-Mode or in GLMM-Mode. In GLM-Mode the “Covariance Parameter Estimates” table is not produced. Scale and dispersion parameters in the model appear in the “Parameter Estimates” table.

---

## Satterthwaite Degrees of Freedom Approximation

The [DDFM=SATTERTH](#) option in the [MODEL](#) statement requests denominator degrees of freedom in  $t$  tests and  $F$  tests computed according to a general Satterthwaite approximation. The [DDFM=KENWARDROGER](#) option also entails the computation of Satterthwaite-type degrees of freedom.

The general Satterthwaite approximation computed in PROC GLIMMIX for the test

$$H: \mathbf{L} \begin{bmatrix} \hat{\beta} \\ \hat{\gamma} \end{bmatrix} = \mathbf{0}$$

is based on the  $F$  statistic

$$F = \frac{\begin{bmatrix} \hat{\beta} \\ \hat{\gamma} \end{bmatrix}' \mathbf{L}'(\mathbf{LCL}')^{-1} \mathbf{L} \begin{bmatrix} \hat{\beta} \\ \hat{\gamma} \end{bmatrix}}{\text{rank}(\mathbf{L})}$$

where  $\mathbf{C}$  is the approximate variance matrix of  $[\hat{\beta}', \hat{\gamma}' - \gamma']'$ ; see the section “Estimated Precision of Estimates” on page 122.

The approximation proceeds by first performing the spectral decomposition  $\mathbf{LCL}' = \mathbf{U}'\mathbf{D}\mathbf{U}$ , where  $\mathbf{U}$  is an orthogonal matrix of eigenvectors and  $\mathbf{D}$  is a diagonal matrix of eigenvalues, both of dimension  $\text{rank}(\mathbf{L}) \times \text{rank}(\mathbf{L})$ . Define  $\mathbf{b}_j$  to be the  $j$ th row of  $\mathbf{UL}$ , and let

$$\nu_j = \frac{2(D_j)^2}{\mathbf{g}_j' \mathbf{A} \mathbf{g}_j}$$

where  $D_j$  is the  $j$ th diagonal element of  $\mathbf{D}$  and  $\mathbf{g}_j$  is the gradient of  $\mathbf{b}_j \mathbf{C} \mathbf{b}_j'$  with respect to  $\boldsymbol{\theta}$ , evaluated at  $\hat{\boldsymbol{\theta}}$ . The matrix  $\mathbf{A}$  is the asymptotic variance-covariance matrix of  $\hat{\boldsymbol{\theta}}$ , obtained from the second derivative matrix of the likelihood equations. You can display this matrix with the `ASYCOV` option in the PROC GLIMMIX statement.

Finally, let

$$E = \sum_{j=1}^{\text{rank}(\mathbf{L})} \frac{\nu_j}{\nu_j - 2} I(\nu_j > 2)$$

where the indicator function eliminates terms for which  $\nu_j \leq 2$ . The degrees of freedom for  $F$  are then computed as

$$\nu = \frac{2E}{E - \text{rank}(\mathbf{L})}$$

provided  $E > \text{rank}(\mathbf{L})$ ; otherwise  $\nu$  is set to zero.

In the one-dimensional case, when PROC GLIMMIX computes a  $t$  test, the Satterthwaite degrees of freedom for the  $t$  statistic

$$t = \frac{Y' \begin{bmatrix} \hat{\beta} \\ \hat{\gamma} \end{bmatrix}}{Y' \mathbf{C} \mathbf{I}}$$

are computed as

$$\nu = \frac{2(Y' \mathbf{C} \mathbf{I})^2}{\mathbf{g}' \mathbf{A} \mathbf{g}}$$

where  $\mathbf{g}$  is the gradient of  $Y' \mathbf{C} \mathbf{I}$  with respect to  $\boldsymbol{\theta}$ , evaluated at  $\hat{\boldsymbol{\theta}}$ .

## Empirical Covariance (“Sandwich”) Estimators

The GLIMMIX procedure can compute the classical sandwich estimator of the covariance matrix of the fixed effects, as well as several bias-adjusted estimators. This requires that the model is either a (overdispersed) GLM or a GLMM that can be processed by subjects (see the section “Processing by Subjects” on page 129).

Consider a statistical model of the form

$$\mathbf{Y} = \boldsymbol{\mu} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim (\mathbf{0}, \boldsymbol{\Sigma})$$

The general expression of a sandwich covariance estimator is then

$$c \times \widehat{\boldsymbol{\Omega}} \left( \sum_{i=1}^m \mathbf{A}_i \widehat{\mathbf{D}}_i' \widehat{\boldsymbol{\Sigma}}_i^{-1} \mathbf{F}_i' \mathbf{e}_i \mathbf{e}_i' \mathbf{F}_i \widehat{\boldsymbol{\Sigma}}_i^{-1} \widehat{\mathbf{D}}_i \mathbf{A}_i \right) \widehat{\boldsymbol{\Omega}}$$

where  $\mathbf{e}_i = \mathbf{y}_i - \widehat{\boldsymbol{\mu}}_i$ ,  $\boldsymbol{\Omega} = (\mathbf{D}'\boldsymbol{\Sigma}^{-1}\mathbf{D})^{-}$ .

For a GLMM estimated by one of the pseudo-likelihood techniques that involve linearization, you can make the following substitutions:  $\mathbf{Y} \rightarrow \mathbf{P}$ ,  $\boldsymbol{\Sigma} \rightarrow \mathbf{V}(\boldsymbol{\theta})$ ,  $\mathbf{D} \rightarrow \mathbf{X}$ ,  $\widehat{\boldsymbol{\mu}} \rightarrow \mathbf{X}\widehat{\boldsymbol{\beta}}$ . These matrices are defined in the section “Pseudo-Likelihood Estimation Based on Linearization” on page 120.

The various estimators computed by the GLIMMIX procedure differ in the choice of the constant  $c$  and the matrices  $\mathbf{F}_i$  and  $\mathbf{A}_i$ . You obtain the classical estimator, for example, with  $c = 1$ , and  $\mathbf{F}_i = \mathbf{A}_i$  equal to the identity matrix.

The EMPIRICAL=ROOT estimator of Kauermann and Carroll (2001) is based on the approximation

$$\text{var} [\mathbf{e}_i \mathbf{e}_i'] \approx (\mathbf{I} - \mathbf{H}_i) \boldsymbol{\Sigma}_i$$

where  $\mathbf{H}_i = \mathbf{D}_i \boldsymbol{\Omega} \mathbf{D}_i' \boldsymbol{\Sigma}_i^{-1}$ . The EMPIRICAL=FIRORES estimator is based on the approximation

$$\text{var} [\mathbf{e}_i \mathbf{e}_i'] \approx (\mathbf{I} - \mathbf{H}_i) \boldsymbol{\Sigma}_i (\mathbf{I} - \mathbf{H}_i')$$

of Mancl and DeRouen (2001). Finally, the EMPIRICAL=FIROEEQ estimator is based on approximating an unbiased estimating equation (Fay and Graubard 2001). For this estimator,  $\mathbf{A}_i$  is a diagonal matrix with entries

$$[\mathbf{A}_i]_{jj} = (1 - \min\{r, [\mathbf{Q}]_{jj}\})^{-1/2}$$

where  $\mathbf{Q} = \mathbf{D}_i' \widehat{\boldsymbol{\Sigma}}_i^{-1} \mathbf{D}_i \widehat{\boldsymbol{\Omega}}$ . The optional number  $0 \leq r < 1$  is chosen to provide an upper bound on the correction factor. For  $r = 0$ , the classical sandwich estimator results. PROC GLIMMIX chooses as default value  $r = 3/4$ . The diagonal entries of  $\mathbf{A}_i$  are then no greater than 2.



Table 12 summarizes the components of the computation for the GLMM based on linearization, where  $m$  denotes the number of subjects and  $k$  is the rank of  $\mathbf{X}$ .

**Table 12.** Empirical Covariance Estimators for a Linearized GLMM

EMPIRICAL=	$c$	$\mathbf{A}_i$	$\mathbf{F}_i$
CLASSICAL	1	$\mathbf{I}$	$\mathbf{I}$
DF	$\begin{cases} \frac{m}{m-k} & m > k \\ 1 & \text{otherwise} \end{cases}$	$\mathbf{I}$	$\mathbf{I}$
ROOT	1	$\mathbf{I}$	$(\mathbf{I} - \mathbf{H}'_i)^{-1/2}$
FIRORES	1	$\mathbf{I}$	$(\mathbf{I} - \mathbf{H}'_i)^{-1}$
FIROEEQ(r)	1	$\text{Diag}\{(1 - \min\{r, [\mathbf{Q}]_{jj}\})^{-1/2}\}$	$\mathbf{I}$

Computation of an empirical variance estimator requires that the data can be processed by independent sampling units. This is always the case in GLMs. In this case,  $m$  equals the sum of all frequencies. In GLMMs, the empirical estimators require that the data consist of multiple subjects. In that case,  $m$  equals the number of subjects as per the “Dimensions” table. The following section discusses how the GLIMMIX procedure determines whether the data can be processed by subjects. The section “GLM-Mode or GLMM-Mode” on page 125 explains how PROC GLIMMIX determines whether a model is fit in GLM-mode or in GLMM-mode.

## Processing by Subjects

Some mixed models can be expressed in different but mathematically equivalent ways with PROC GLIMMIX statements. While equivalent statements lead to equivalent statistical models, the data processing and estimation phase may be quite different, depending on how you write the GLIMMIX statements. For example, the particular use of the `SUBJECT=` option in the `RANDOM` statement affects data processing and estimation. Certain options are only available when the data are processed by subject, for example, the `EMPIRICAL` option in the `PROC GLIMMIX` statement. Consider a GLIMMIX model where variables `A` and `Rep` are classification variables with  $a$  and  $r$  levels, respectively. The following statements produce the same random effects structure:

1. `class Rep A;`  
`random Rep*A;`
2. `class Rep A;`  
`random intercept / subject=Rep*A;`
3. `class Rep A;`  
`random Rep / subject=A;`
4. `class Rep A;`  
`random A / subject=Rep;`

In the first case, PROC GLIMMIX will not process the data by subjects because no `SUBJECT=` option was given. The computation of empirical covariance estimators,

for example, will not be possible. The marginal variance-covariance matrix has the same block-diagonal structure as for cases 2–4, where each block consists of the observations belonging to a unique combination of `Rep` and `A`. More importantly, the dimension of the  $\mathbf{Z}$  matrix of this model will be  $n \times ra$ , and  $\mathbf{Z}$  will be sparse. In the second case, the  $\mathbf{Z}_i$  matrix for each of the  $ra$  subjects is a vector of ones.

If the data can be processed by subjects, the procedure typically executes faster and requires less memory. The differences can be substantial, especially if the number of subjects is large. Recall that fitting of generalized linear mixed models may be doubly iterative. Small gains in efficiency for any one optimization can produce large overall savings.

If you interpret the intercept as “1”, then a `RANDOM` statement with `TYPE=VC` (the default) and no `SUBJECT=` option can be converted into a statement with subject by dividing the random effect by the eventual subject effect. However, the presence of the `SUBJECT=` option does not imply processing by subject. If a `RANDOM` statement does not have a `SUBJECT=` effect, processing by subjects is not possible unless the random effect is a pure R-side overdispersion effect. In the following example, the data will not be processed by subjects

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

because the first `RANDOM` statement specifies a G-side component and does not use a `SUBJECT=` option. To allow processing by subjects, you can write the equivalent model

```
proc glimmix;
  class A B;
  model y = B;
  random int / subject=A;
  random B / subject=A;
run;
```

If you denote a variance component effect  $X$  with subject effect  $S$  as  $X-(S)$ , then the “calculus of random effects” applied to the first `RANDOM` statement reads  $A = \text{Int} * A = \text{Int}-(A) = A-(\text{Int})$ . For the second statement there are even more equivalent formulations:  $A * B = A * B * \text{Int} = A * B-(\text{Int}) = A-(B) = B-(A) = \text{Int}-(A * B)$ .

If there are multiple subject effects, processing by subjects is possible if the effects are equal or contained in each other. Note that in the last example the  $A * B$  interaction is a random effect. An equivalent specification to the last would be

```
proc glimmix;
  class A B;
  model y = B;
  random int / subject=A;
  random A / subject=B;
run;
```

Processing by subjects would not be possible in this case because the two subject effects are not syntactically equal or contained in each other. A case where subject effects are syntactically contained would be

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

The A main effect is contained in the A\*B interaction. The GLIMMIX procedure chooses as the subject effect for processing the effect that is contained in all other subject effects. In this case, the subjects are defined by the levels of A.

You can examine the “Model Information” and “Dimensions” tables to see whether the GLIMMIX procedure processes the data by subjects and which effect is used to define subjects. The “Model Information” table displays whether the marginal variance matrix is diagonal (GLM models), blocked, or not blocked. The “Dimensions” table tells you how many subjects (=blocks) there are.

Finally, nesting or crossing of interaction effects in subject effects are equivalent. The following two random statements are equivalent:

```
class Rep A;
random intercept / subject=Rep*A;

class Rep A;
random intercept / subject=Rep(A);
```

---

## Radial Smoothing Based on Mixed Models

The radial smoother implemented with the `TYPE=RSMOOTH` option of the `RANDOM` statement is an approximate low-rank thin plate spline as described in Ruppert, Wand, and Carroll (2003, Chapter 13.4–13.5). The following sections discuss in more detail the mathematical-statistical connection between mixed models and penalized splines and the determination of the number of spline knots and their location as implemented by the GLIMMIX procedure.

### From Penalized Splines to Mixed Models

The connection between splines and mixed models arises from the similarity of the penalized spline fitting criterion to the minimization problem that yields the mixed model equations and solutions for  $\beta$  and  $\gamma$ . This connection is made explicit in the following paragraphs. An important distinction between classical spline fitting and its mixed model smoothing variant, however, lies in the nature of the spline coefficient. Although they address similar minimization criteria, the solutions for the spline coefficients in the GLIMMIX procedure are the solutions of random effects, not fixed effects. Standard errors of predicted values, for example, account for this source of variation.

Consider the linearized mixed pseudo-model from the section “The Pseudo-Model” on page 120,  $\mathbf{P} = \mathbf{X}\beta + \mathbf{Z}\gamma + \epsilon$ . One derivation of the mixed model equations, whose solutions are  $\hat{\beta}$  and  $\hat{\gamma}$ , is to maximize the joint density of  $f(\gamma, \epsilon)$  with respect to  $\beta$  and  $\gamma$ . This is not a true likelihood problem, since  $\gamma$  is not a parameter, but a random vector.

In the special case with  $\text{var}[\epsilon] = \phi\mathbf{I}$  and  $\text{var}[\gamma] = \sigma^2\mathbf{I}$ , the maximization of  $f(\gamma, \epsilon)$  is equivalent to the minimization of

$$Q(\beta, \gamma) = \phi^{-1}(\mathbf{p} - \mathbf{X}\beta - \mathbf{Z}\gamma)'(\mathbf{p} - \mathbf{X}\beta - \mathbf{Z}\gamma) + \sigma^{-2}\gamma'\gamma$$

Now consider a linear spline as in Ruppert, Wand, and Carroll (2003, p. 108),

$$p_i = \beta_0 + \beta_1 x_i + \sum_{j=1}^K \gamma_j (x_i - t_j)_+$$

where the  $\gamma_j$  denote the spline coefficients at knots  $t_1, \dots, t_K$ . The truncated line function is defined as

$$(x - t)_+ = \begin{cases} x - t & x > t \\ 0 & \text{otherwise} \end{cases}$$

If you collect the intercept and regressor  $x$  into the matrix  $\mathbf{X}$ , and if you collect the truncated line functions into the  $(n \times K)$  matrix  $\mathbf{Z}$ , then fitting the linear spline amounts to minimization of the penalized spline criterion

$$Q^*(\beta, \gamma) = (\mathbf{p} - \mathbf{X}\beta - \mathbf{Z}\gamma)'(\mathbf{p} - \mathbf{X}\beta - \mathbf{Z}\gamma) + \lambda^2\gamma'\gamma$$

where  $\lambda$  is the smoothing parameter.

Because minimizing  $Q^*(\beta, \gamma)$  with respect to  $\beta$  and  $\gamma$  is equivalent to minimizing  $Q^*(\beta, \gamma)/\phi$ , both problems lead to the same solution, and  $\lambda = \phi/\sigma$  is the smoothing parameter. The mixed model formulation of spline smoothing has the advantage that the smoothing parameter is selected “automatically.” It is a function of the covariance parameter estimates, which, in turn, are estimated according to the method you specify with the `METHOD=` option in the `PROC GLIMMIX` statement.

To accommodate nonnormal responses and general link functions, the GLIMMIX procedure uses  $\text{var}[\epsilon] = \phi \tilde{\mathbf{\Delta}}^{-1} \mathbf{A} \tilde{\mathbf{\Delta}}^{-1}$ , where  $\mathbf{A}$  is the matrix of variance functions and  $\mathbf{\Delta}$  is the diagonal matrix of mean derivatives defined earlier. The correspondence between spline smoothing and mixed modeling is then one between a weighted linear mixed model and a weighted spline. In other words, the minimization criterion that yields the estimates  $\hat{\beta}$  and solutions  $\hat{\gamma}$  is then

$$Q(\beta, \gamma) = \phi^{-1} (\mathbf{p} - \mathbf{X}\beta - \mathbf{Z}\gamma)' \tilde{\mathbf{\Delta}} \mathbf{A}^{-1} \tilde{\mathbf{\Delta}} (\mathbf{p} - \mathbf{X}\beta - \mathbf{Z}\gamma) + \sigma^{-2} \gamma' \gamma$$

If you choose the `TYPE=RSMOOTH` covariance structure, PROC GLIMMIX chooses radial basis functions as the spline basis and transforms them to approximate a thin-plate spline as in Chapter 13.4 of Ruppert, Wand, and Carroll (2003). For computational expediency, the number of knots is chosen to be less than the number of data points. Ruppert, Wand, and Carroll (2003) recommend one knot per every four unique regressor values for one-dimensional smoothers. In the multivariate case, general recommendations are more difficult, because the optimal number and placement of knots depends on the spatial configuration of samples. Their recommendation for a bivariate smoothers is one knot per four samples, but at least 20 and no more than 150 knots (Ruppert et al. 2003, p. 257).

The magnitude of the variance component  $\sigma^2$  is dependent on the metric of the random effects. For example, if you apply radial smoothing in time, the variance changes if you measure time in days or minutes. If the solution for the variance component is near zero, then a rescaling of the random effect data can help the optimization problem by moving the solution for the variance component away from the boundary of the parameter space.

### Knot Selection

The GLIMMIX procedure computes knots for low-rank smoothing based on the vertices or centroids of a  $k$ - $d$  tree. The default is to use the vertices of the tree as the knot locations, if you use the `TYPE=RSMOOTH` covariance structure. The construction of this tree amounts to a partitioning of the random regressor space until all partitions contain at most  $b$  observations. The number  $b$  is called the *bucket size* of the  $k$ - $d$  tree. You can exercise control over the construction of the tree by changing the bucket size with the `BUCKET=` suboption of the `KNOTMETHOD=KDTREE` option in the `RANDOM` statement. A large bucket size leads to fewer knots, but it is not correct to assume that  $K$ , the number of knots, is simply  $\lfloor n/b \rfloor$ . The number of vertices depends on the configuration of the values in the regressor space. Also, coordinates of the bounding hypercube are vertices of the tree. In the one-dimensional case, for example, the extreme values of the random effect are vertices.

To demonstrate how the  $k$ - $d$  tree partitions the random effects space based on observed data and the influence of the bucket size, consider the following example from Chapter 41, “The LOESS Procedure” (*SAS/STAT User’s Guide*). The SAS data set `Gas` contains the results of an engine exhaust emission study (Brinkman 1981). The covariate in this analysis, `E`, is a measure of the air-fuel mixture richness. The response, `NOx`, measures the nitric oxide concentration (in micrograms per joule, and normalized).

```

data Gas;
  input NOx E;
  format NOx E f5.3;
  datalines;
4.818 0.831
2.849 1.045
3.275 1.021
4.691 0.97
4.255 0.825
5.064 0.891
2.118 0.71
4.602 0.801
2.286 1.074
0.97 1.148
3.965 1
5.344 0.928
3.834 0.767
1.99 0.701
5.199 0.807
5.283 0.902
3.752 0.997
0.537 1.224
1.64 1.089
5.055 0.973
4.937 0.98
1.561 0.665
;

```

There are 22 observations in the data set, and the values of the covariate are unique. If you want to smooth these data with a low-rank radial smoother, you need to choose the number of knots, as well as their placement within the support of the variable *E*. The *k-d* tree construction depends on the observed values of the variable *E*; it is independent of the values of nitric oxide in the data. The following statements construct a tree based on a bucket size of  $b = 11$  and display information about the tree and the selected knots:

```

ods select KDtree KnotInfo;
proc glimmix data=gas nofit;
  model NOx = e;
  random e / type=rsmooth
            knotmethod=kdtree(bucket=11 treeinfo knotinfo);
run;

```

The **NOFIT** option prevents the GLIMMIX procedure from fitting the model. This option is useful if you want to investigate the knot construction for various bucket sizes. The **TREEINFO** and **KNOTINFO** suboptions of the **KNOTMETHOD=KDTREE** option request displays of the *k-d* tree and the knot coordinates derived from it. Construction of the tree commences by splitting the data in half. For  $b = 11$ ,  $n = 22$ , neither of the two splits contains more than  $b$  observations and the process stops. With a single split value, and the two extreme values, the tree has two terminal nodes and leads to three knots (Figure 11). Note

that for one-dimensional problems, vertices of the  $k$ - $d$  tree always coincide with data values.

The GLIMMIX Procedure				
kd-Tree for RSmooth(E)				
Node Number	Left Child	Right Child	Split Direction	Split Value
0	1	2	E	0.9280
1			TERMINAL	
2			TERMINAL	

Radial Smoother		
Knots for RSmooth(E)		
Knot Number	E	
1	0.6650	
2	0.9280	
3	1.2240	

**Figure 11.**  $K$ - $d$  Tree and Knots for Bucket Size 11

If the bucket size is reduced to  $b = 8$ , the statements

```
ods select KDtree KnotInfo;
proc glimmix data=gas nofit;
  model NOx = e;
  random e / type=rsmooth
            knotmethod=kdtree(bucket=8 treeinfo knotinfo);
run;
```

produce the tree and knots in [Figure 12](#). The initial split value of 0.9280 leads to two sets of 11 observations. In order to achieve a partition into cells that contain at most eight observations, each initial partition is split at its median one more time. Note that one split value is greater and one split value is less than 0.9280.

The GLIMMIX Procedure				
kd-Tree for RSmooth(E)				
Node Number	Left Child	Right Child	Split Direction	Split Value
0	1	2	E	0.9280
1	3	4	E	0.8070
2	5	6	E	1.0210
3			TERMINAL	
4			TERMINAL	
5			TERMINAL	
6			TERMINAL	

Radial Smoother Knots for RSmooth(E)		
Knot Number		E
1		0.6650
2		0.8070
3		0.9280
4		1.0210
5		1.2240

**Figure 12.** K-d Tree and Knots for Bucket Size 8

A further reduction in bucket size to  $b = 4$  leads to the tree and knot information shown in [Figure 13](#).



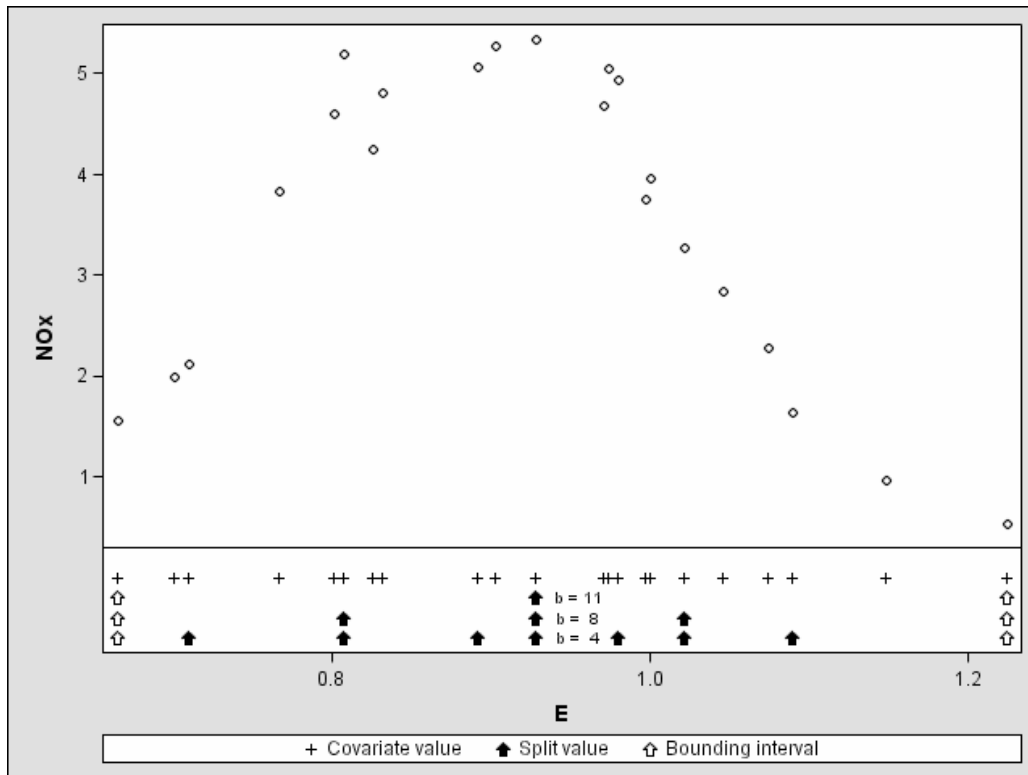
The GLIMMIX Procedure				
kd-Tree for RSmooth(E)				
Node Number	Left Child	Right Child	Split Direction	Split Value
0	1	2	E	0.9280
1	3	4	E	0.8070
2	9	10	E	1.0210
3	5	6	E	0.7100
4	7	8	E	0.8910
5			TERMINAL	
6			TERMINAL	
7			TERMINAL	
8			TERMINAL	
9	11	12	E	0.9800
10	13	14	E	1.0890
11			TERMINAL	
12			TERMINAL	
13			TERMINAL	
14			TERMINAL	

Radial Smoother		
Knots for RSmooth(E)		
Knot Number		E
1		0.6650
2		0.7100
3		0.8070
4		0.8910
5		0.9280
6		0.9800
7		1.0210
8		1.0890
9		1.2240

**Figure 13.** K-d Tree and Knots for Bucket Size 4

The split value for  $b = 11$  is also a split value for  $b = 8$ , the split values for  $b = 8$  are a subset of those for  $b = 4$ , and so forth. Figure 14 displays the data and the location of split values for the three cases. For a one-dimensional problem (a univariate smoother), the vertices comprise the split values and the values on the bounding interval.



**Figure 14.** Vertices of  $k$ - $d$  Trees for Various Bucket Sizes

You may wish to move away from the boundary, in particular if the data configuration is irregular or for multivariate smoothing. The `KNOTTYPE=CENTER` suboption of the `KNOTMETHOD=` option chooses centroids of the leaf node cells instead of vertices. This tends to move the outer knot locations closer to the convex hull, but not necessarily to data locations. In the emission example, choosing a bucket size of  $b = 11$  and centroids as knot locations yields two knots at  $E=0.7956$  and  $E=1.076$ . If you choose the `NEAREST` suboption, then the nearest neighbor of a vertex or centroid will serve as the knot location. In this case, the knot locations are a subset of the data locations, regardless of the dimension of the smooth.

## Odds and Odds Ratio Estimation

In models with a logit, generalized logit, or cumulative logit link, you can obtain estimates of odds ratios through the `ODDSRATIO` options in the `PROC GLIMMIX`, `LSMEANS`, and `MODEL` statement. This section provides details about the computation and interpretation of the computed quantities. Note that for these link functions the `EXP` option in the `ESTIMATE` and `LSMESTIMATE` statements also produces odds or odds ratios.

Consider first a model with a dichotomous outcome variable, linear predictor  $\eta = \mathbf{x}'\boldsymbol{\beta} + \mathbf{z}'\boldsymbol{\gamma}$ , and logit link function. Suppose that  $\eta_0$  represents the linear predictor for a condition of interest. For example, in a simple logistic regression model with  $\eta = \alpha + \beta x$ ,  $\eta_0$  may correspond to the linear predictor at a particular value of the covariate, say  $\eta_0 = \alpha + \beta x_0$ .

The modeled probability is  $\pi = 1/(1 + \exp\{-\eta\})$ , and the odds for  $\eta = \eta_0$  are

$$\frac{\pi_0}{1 - \pi_0} = \frac{1/(1 + \exp\{-\eta_0\})}{\exp\{-\eta_0\}/(1 + \exp\{-\eta_0\})} = \exp\{\eta_0\}$$

Since  $\eta_0$  is a logit, it represents the log odds. The odds ratio  $\psi(\eta_1, \eta_0)$  is defined as the ratio of odds for  $\eta_1$  and  $\eta_0$ ,

$$\psi(\eta_1, \eta_0) = \exp\{\eta_1 - \eta_0\}$$

The odds ratio compares the odds of the outcome under the condition expressed by  $\eta_1$  to the odds under the condition expressed by  $\eta_0$ . In the simple logistic regression example above, this ratio equals  $\exp\{\beta(x_1 - x_0)\}$ . The exponentiation of the estimate of  $\beta$  is thus an estimate of the odds ratio comparing conditions for which  $x_1 - x_0 = 1$ . If  $x$  and  $x + 1$  represent standard and experimental conditions, for example,  $\exp\{\beta\}$  compares the odds of the outcome under the experimental condition to the odds under the standard condition. For many other types of models, odds ratios can be expressed as simple functions of parameter estimates. For example, suppose you are fitting a logistic model with a single classification effect with three levels

```
proc glimmix;
  class A;
  model y = A / dist=binary;
run;
```

The estimated linear predictor for level  $j$  of **A** is  $\hat{\eta}_j = \hat{\beta} + \hat{\alpha}_j$ ,  $j = 1, 2, 3$ . Because the **X** matrix is singular in this model due to the presence of an overall intercept, the solution for the intercept estimates  $\beta + \alpha_3$ , and the solution for the  $j$ th treatment effect estimates  $\alpha_j - \alpha_3$ . Exponentiating the solutions for  $\alpha_1$  and  $\alpha_2$  thus produces odds ratios comparing the odds for these levels against the third level of **A**.

Results designated as odds or odds ratios in the GLIMMIX procedure may reduce to simple exponentiations of solutions in the “Parameter Estimates” table, but they are computed by a different mechanism if the model contains classification variables. The computations rely on general estimable functions; for the **MODEL**, **LSMEANS**, and **LSMESTIMATE** statements, these functions are based on least-squares means. This enables you to obtain odds ratio estimates in more complicated models that involve main effects and interactions, including interactions between continuous and classification variables.

In all cases, the results represent the exponentiation of a linear function of the fixed-effects parameters,  $\eta = I'\beta$ . If  $L_\eta$  and  $U_\eta$  are the confidence limits for  $\eta$  on the logit scale, confidence limits for the odds or the odds ratio are obtained as  $\exp\{L_\eta\}$  and  $\exp\{U_\eta\}$ .

### The Odds Ratio Estimates Table

This table is produced by the `ODDSRATIO` option in the `MODEL` statement. It consists of estimates of odds ratios and their confidence limits. Odds ratios are produced for the following:

- Classification main effects, if they appear in the `MODEL` statement
- Continuous variables in the `MODEL` statement, unless they appear in an interaction with a classification effect
- Continuous variables in the `MODEL` statement at fixed levels of a classification effect, if the `MODEL` statement contains an interaction of the two.
- Continuous variables in the `MODEL` statements if they interact with other continuous variables

### The Default Table

Consider the following PROC GLIMMIX statements that fit a logistic model with one classification effect, one continuous variable, and their interaction (the `ODDSRATIO` option in the `MODEL` statement requests the “Odds Ratio Estimates” table).

```
proc glimmix;
  class A;
  model y = A x A*x / dist=binary oddsratio;
run;
```

By default, odds ratios are computed as follows:

- The covariate is set to its average,  $\bar{x}$ , and the least-squares means for the `A` effect are obtained. Suppose  $\mathbf{L}^{(1)}$  denotes the matrix of coefficients defining the estimable functions that produce the  $a$  least-squares means  $\mathbf{L}\hat{\boldsymbol{\beta}}$ , and  $\mathbf{I}_j^{(1)}$  denotes the  $j$ th row of  $\mathbf{L}^{(1)}$ . Differences of the least-squares means against the last level of the `A` factor are computed and exponentiated:

$$\begin{aligned} \psi(A_1, A_a) &= \exp \left\{ \left( \mathbf{I}_1^{(1)} - \mathbf{I}_a^{(1)} \right) \hat{\boldsymbol{\beta}} \right\} \\ \psi(A_2, A_a) &= \exp \left\{ \left( \mathbf{I}_2^{(1)} - \mathbf{I}_a^{(1)} \right) \hat{\boldsymbol{\beta}} \right\} \\ &\vdots \\ \psi(A_{a-1}, A_a) &= \exp \left\{ \left( \mathbf{I}_{a-1}^{(1)} - \mathbf{I}_a^{(1)} \right) \hat{\boldsymbol{\beta}} \right\} \end{aligned}$$

The differences are checked for estimability. Notice that this set of odds ratios can also be obtained with the following `LSMESTIMATE` statement (assuming `A` has five levels):

```
lsmestimate A 1 0 0 0 -1,
              0 1 0 0 -1,
              0 0 1 0 -1,
              0 0 0 1 -1 / exp cl;
```

or with this `LSMEANS` statement (assuming the last level of `A` is coded as 5)

```
lsmeans A / diff=control('5') oddsratio cl;
```

- The odds ratios for the covariate must take into account that `x` occurs in an interaction with the `A` effect. A second set of least-squares means are computed, where `x` is set to  $\bar{x} + 1$ . Denote the coefficients of the estimable functions for this set of least-squares means as  $\mathbf{L}^{(2)}$ . Differences of the least-squares means at a given level of factor `A` are then computed and exponentiated:

$$\begin{aligned}\psi(A(\bar{x} + 1)_1, A(\bar{x})_1) &= \exp \left\{ \left( \mathbf{l}_1^{(2)} - \mathbf{l}_1^{(1)} \right) \widehat{\boldsymbol{\beta}} \right\} \\ \psi(A(\bar{x} + 1)_2, A(\bar{x})_2) &= \exp \left\{ \left( \mathbf{l}_2^{(2)} - \mathbf{l}_2^{(1)} \right) \widehat{\boldsymbol{\beta}} \right\} \\ &\vdots \\ \psi(A(\bar{x} + 1)_a, A(\bar{x})_a) &= \exp \left\{ \left( \mathbf{l}_a^{(2)} - \mathbf{l}_a^{(1)} \right) \widehat{\boldsymbol{\beta}} \right\}\end{aligned}$$

The differences are checked for estimability. If the continuous covariate does not appear in an interaction with the `A` variable, only a single odds ratio estimate related to `x` would be produced, relating the odds of a one-unit shift in the regressor from  $\bar{x}$ .

Suppose you fit a model that contains interactions of continuous variables, as with the statements

```
proc glimmix;
  class A;
  model y = A x x*z / dist=binary oddsratio;
run;
```

In the computation of the `A` least-squares means, the continuous effects are set to their means, that is,  $\bar{x}$  and  $\bar{x}\bar{z}$ . In the computation of odds ratios for `x`, linear predictors are computed at  $x = \bar{x}$ ,  $x^*z = \bar{x} \times \bar{z}$ , and at  $x = \bar{x} + 1$ ,  $x^*z = (\bar{x} + 1)\bar{z}$ .

### Modifying the Default Table, Customized Odds Ratios

Several suboptions of the `ODDSRATIO` option in the `MODEL` statement are available to obtain customized odds ratio estimates. For customized odds ratios that can not be obtained with these suboptions, use the `EXP` option in the `ESTIMATE` or `LSMESTIMATE` statements.

The type of differences constructed when the levels of a classification factor are varied is controlled by the `DIFF=` suboption. By default, differences against the last level are taken. `DIFF=FIRST` computes differences from the first level, and `DIFF=ALL` computes odds ratios based on all pairwise differences.

For continuous variables in the model, you can change both the reference value (with the `AT` suboption) and the units of change (with the `UNIT` suboption). By default, a one-unit change from the mean of the covariate is assessed. For example, the statements

```

proc glimmix;
  class A;
  model y = A x A*x / dist=binary
                                oddsratio(diff=all
                                             at    x=4
                                             unit  x=3);
run;

```

produce all pairwise differences for the **A** factor. The covariate **x** is set to the reference value  $x = 4$  in the computation of the least-squares means for the **A** odds ratio estimates. The odds ratios computed for the covariate, are based on differencing this set of least-squares means with a set of least-squares means computed at  $x = 4 + 3$ .

### Odds or Odds Ratio

The odds ratio is the exponentiation of a difference on the logit scale,

$$\psi(\eta_1, \eta_0) = \exp\{(I_1 - I_0)\beta\}$$

and  $\exp\{I_1\beta\}$  and  $\exp\{I_0\beta\}$  are the corresponding odds. If the **ODDSRATIO** option is specified in a suitable model in the **PROC GLIMMIX** statement or the individual statements that support the option, odds ratios are computed in the “Odds Ratio Estimates” table (**MODEL** statement), the “Differences of Least Squares Means” table (**LSMEANS / DIFF**), and the “Simple Effect Comparisons of Least Squares Means” table (**LSMEANS / SLICEDIFF=**). Odds are computed in the “Least Squares Means” table.

### Odds Ratios in Multinomial Models

The GLIMMIX procedure fits two kinds of models to multinomial data. Models with cumulative link functions apply to ordinal data, generalized logit models are fit to nominal data. If you model a multinomial response with **LINK=CUMLOGIT** or **LINK=GLOGIT**, odds ratio results are available for these models.

In the generalized logit model, you model baseline category logits. By default, the GLIMMIX procedure chooses the last category as the reference category. If your nominal response has  $J$  categories, the baseline logit for category  $j$  is

$$\log\{\pi_j/\pi_J\} = \eta_j = \mathbf{x}'\beta_j + \mathbf{z}'\mathbf{u}_j$$

and

$$\pi_j = \frac{\exp\{\eta_j\}}{\sum_{k=1}^J \exp\{\eta_k\}}$$

$$\eta_J = 0$$

As before, suppose that the two conditions to be compared are identified with subscripts 1 and 0. The log odds ratio of outcome  $j$  versus  $J$  for the two conditions is

then

$$\begin{aligned}\log \{ \psi (\eta_{j1}, \eta_{j0}) \} &= \log \left\{ \frac{\pi_{j1}/\pi_{J1}}{\pi_{j0}/\pi_{J0}} \right\} = \log \left\{ \frac{\exp\{\eta_{j1}\}}{\exp\{\eta_{j0}\}} \right\} \\ &= \eta_{j1} - \eta_{j0}\end{aligned}$$

Note that the log odds ratios are again differences on the scale of the linear predictor but they depend on the response category. The GLIMMIX procedure determines the estimable functions whose differences represent log odds ratios as discussed previously but produces separate estimates for each non-reference response category.

In models for ordinal data, PROC GLIMMIX models the logits of cumulative probabilities. Thus, the estimates on the linear scale represent log cumulative odds. The cumulative logits are formed as

$$\log \left\{ \frac{\Pr(Y \leq j)}{\Pr(Y > j)} \right\} = \eta_j = \alpha_j + \mathbf{x}'\boldsymbol{\beta} + \mathbf{z}'\boldsymbol{\gamma} = \alpha_j + \tilde{\eta}$$

so that the linear predictor depends on the response category only through the intercepts (cutoffs)  $\alpha_1, \dots, \alpha_{J-1}$ . The odds ratio comparing two conditions represented by linear predictors  $\eta_{j1}$  and  $\eta_{j0}$  is then

$$\begin{aligned}\psi (\eta_{j1}, \eta_{j0}) &= \exp \{ \eta_{j1} - \eta_{j0} \} \\ &= \exp \{ \tilde{\eta}_1 - \tilde{\eta}_0 \}\end{aligned}$$

and is independent of category.

---

## Parameterization of Generalized Linear Mixed Models

PROC GLIMMIX constructs a generalized linear mixed model according to the specifications in the **CLASS**, **MODEL**, and **RANDOM** statements. Each effect in the **MODEL** statement generates one or more columns in the matrix **X**, and each G-side effect in the **RANDOM** statement generates one or more columns in the matrix **Z**. R-side effects in the **RANDOM** statement do not generate model matrices; they serve only to index observations within subjects. This section shows how the GLIMMIX procedure builds **X** and **Z**.

### Intercept

By default, all models automatically include a column of 1s in **X** to estimate a fixed-effect intercept parameter. You can use the **NOINT** option in the **MODEL** statement to suppress this intercept. The **NOINT** option is useful when you are specifying a classification effect in the **MODEL** statement and you want the parameter estimate to be in terms of the (linked) mean response for each level of that effect, rather than in terms of a deviation from an overall mean.

By contrast, the intercept is not included by default in **Z**. To obtain a column of 1s in **Z**, you must specify in the **RANDOM** statement either the **INTERCEPT** effect or some effect that has only one level.

### Regression Effects

Numeric variables, or polynomial terms involving them, may be included in the model as regression effects (covariates). The actual values of such terms are included as columns of the model matrices  $\mathbf{X}$  and  $\mathbf{Z}$ . You can use the bar operator with a regression effect to generate polynomial effects. For instance,  $X|X|X$  expands to  $X$   $X^*X$   $X^*X^*X$ , a cubic model.

### Main Effects

If a class variable has  $m$  levels, PROC GLIMMIX generates  $m$  columns in the model matrix for its main effect. 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 can be controlled with the ORDER= option in the PROC GLIMMIX statement. Table 13 is an example where  $\beta_0$  denotes the intercept.

**Table 13.** Main Effects Example

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 for them. In other words, PROC GLIMMIX uses an over-parameterized model.

### Interaction Effects

Often a model includes interaction (crossed) effects. With an interaction, PROC GLIMMIX first reorders the terms to correspond to the order of the variables in the CLASS. Thus,  $B^*A$  becomes  $A^*B$  if A precedes B in the CLASS statement. Then, PROC GLIMMIX 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 cross index faster than the leftmost variables (Table 14). Empty columns (that would contain all 0s) are not generated for  $\mathbf{X}$ , but they are for  $\mathbf{Z}$ .

**Table 14.** Interaction Effects Example

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-effect 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 may be able to code them more parsimoniously 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 PROC GLIMMIX is more 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 15).

**Table 15.** Nested Effects Example

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

Note that nested effects are often distinguished from interaction effects by the implied randomization structure of the design. That is, they usually indicate random effects within a fixed-effects framework. The fact that random effects can be modeled directly in the RANDOM statement may make the specification of nested effects in the MODEL statement unnecessary.

### Continuous-Nesting-Class Effects

When a continuous variable nests with a class variable, the design columns are constructed by multiplying the continuous values into the design columns for the class effect (Table 16).

**Table 16.** Continuous-Nesting-Class Effects Example

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 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. The two models are made different by the presence of the continuous variable as a regressor by itself, as well as a contributor to a compound effect (Table 17).

**Table 17.** Continuous-by-Class Effects Example

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 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 the **CONTRAST** or **ESTIMATE** statements to compute some function of the parameter estimates.

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

- Class 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 moves fastest because it is rightmost in the cross list. Then A moves next fastest, and D moves next fastest. The C effect is the slowest since it is leftmost in the nested list.

When numeric levels are used, levels are sorted by their character format, which may not correspond to their numeric sort sequence (for example, noninteger levels). Therefore, it is advisable to include a desired format for numeric levels or to use the **ORDER=INTERNAL** option in the **PROC GLIMMIX** statement to ensure that levels are sorted by their internal values.

### **Implications of the Non-Full-Rank Parameterization**

For models with fixed-effects involving class variables, there are more design columns in **X** constructed than there are degrees of freedom for the effect. Thus, there are linear dependencies among the columns of **X**. In this event, all of the parameters are not estimable; there is an infinite number of solutions to the mixed model equations. The **GLIMMIX** procedure uses a generalized (g2) inverse to obtain values for the estimates (Searle 1971). The solution values are not displayed unless you specify the **SOLUTION** option in the **MODEL** statement. The solution has the characteristic that estimates are 0 whenever the design column for that parameter is a

linear combination of previous columns. With this parameterization, hypothesis tests are constructed to test linear functions of the parameters that are estimable.

Some procedures (such as the CATMOD and LOGISTIC procedures) reparameterize models to full rank using restrictions on the parameters. PROC GLM, PROC MIXED, and PROC GLIMMIX do not reparameterize, making the hypotheses that are commonly tested more understandable. Refer to Goodnight (1978) for additional reasons for not reparameterizing.

### **Missing Level Combinations**

PROC GLIMMIX handles missing level combinations of classification variables in the same manner as PROC GLM and PROC MIXED. These procedures delete fixed-effects parameters corresponding to missing levels in order to preserve estimability. However, PROC GLIMMIX does not delete missing level combinations for random-effects parameters because linear combinations of the random-effects parameters are always estimable. These conventions can affect the way you specify your CONTRAST and ESTIMATE coefficients.

---

## **Response Level Ordering and Referencing**

In models for binary and multinomial data, the response level ordering is important because it reflects

- which probability is modeled with binary data
- how categories are ordered for ordinal data
- which category serves as the reference category in nominal generalized logit models (models for nominal data)

You should view the “Response Profile” table to ensure that the categories are properly arranged and that the desired outcome is modeled. In this table, response levels are arranged by *Ordered Value*. The lowest response level is assigned Ordered Value 1, the next lowest is assigned Ordered Value 2, and so forth. In binary models, the probability modeled is the probability of the response level with the lowest Ordered Value.

You can change which probability is modeled and the Ordered Value in the “Response Profile” table with the DESCENDING, EVENT=, ORDER=, and REF= response variable options in the MODEL statement. See the section “Response Level Ordering” (Chapter 45, *SAS/STAT User’s Guide*) in Chapter 42, “The LOGISTIC Procedure,” (*SAS/STAT User’s Guide*) for examples on how to use these options to affect the probability being modeled for binary data.

For multinomial models, the response level ordering affects two important aspects. In cumulative link models the categories are assumed ordered according to their Ordered Value in the “Response Profile” table. If the response variable is a character variable, or has a format, you should check this table carefully as to whether the Ordered Values reflect the correct ordinal scale.

In generalized logit models (for multinomial data with unordered categories), one response category is chosen as the reference category in the formulation of the generalized logits. By default, the linear predictor in the reference category is set to 0, and the reference category corresponds to the entry in the “Response Profile” table with the highest Ordered Value. You can affect the assignment of Ordered Values with the **DESCENDING** and **ORDER=** options of the **MODEL** statement. You can choose a different reference category with the **REF=** option. The choice of the reference category for generalized logit models affects the results. It is sometimes recommended to choose the category with the highest frequency as the reference (see, for example, Brown and Prescott 1999, p. 160). You can achieve this with the GLIMMIX procedure by combining the **ORDER=** and **REF=** options. For example,

```
proc glimmix;
  class preference;
  model preference(order=freq ref=first) = feature price /
    dist=multinomial link=glogit;
  random intercept / subject=store group=preference;
run;
```

The **ORDER=FREQ** option arranges the categories by descending frequency. The **REF=FIRST** option then selects the response category with the lowest Ordered Value—the most frequent category—as the reference.

---

## Comparing PROC GLIMMIX with PROC MIXED

The GLIMMIX and MIXED procedures have certain functionality in common, but they also have some important differences. Also, the **%GLIMMIX** macro, which fits generalized linear mixed models by linearization methods, essentially calls the MIXED procedure repeatedly. If you are aware of the syntax differences between the procedures, you can easily convert your **%GLIMMIX** code.

Functional differences between PROC GLIMMIX and PROC MIXED for linear models include the following:

- PROC GLIMMIX does not have a **REPEATED** statement. R-side covariance structures are modeled with the **RANDOM** statement, using either the **\_RESIDUAL\_** keyword or the **RESIDUAL** option. For example, the statement

```
repeated / subject=id type=ar(1);
```

in the MIXED procedure and the **PROC GLIMMIX** statement

```
random _residual_ / subject=id type=ar(1);
```

are equivalent. If you need to specify an effect for levelization, the **RESIDUAL** option designates the R-side nature. For example, the statements

```
class time id;
repeated time / subject=id type=ar(1);
```

in the MIXED procedure and the PROC GLIMMIX statements

```
class time id;
random time / subject=id type=ar(1) residual;
```

are equivalent.

- With the GLIMMIX procedure, **MODEL**, **WEIGHT**, and **FREQ** variables, as well as variables specifying the **RANDOM** effects, **SUBJECT=** and **GROUP=** structure, do not have to be in the data set. They can be computed with GLIMMIX programming statements.
- In the GLIMMIX procedure, **RANDOM** statement options apply to the **RANDOM** statement in which they are specified. For example, the statements

```
random a      / s;
random a*b    / G;
random a*b*c  / alpha=0.04;
```

in the GLIMMIX procedure request that the solution vector be printed for the **A** and **A\*B\*C** random effects and that the **G** matrix corresponding to the **A\*B** interaction random effect is printed. Confidence intervals with a 0.96 coverage probability are produced for the solutions of the **A\*B\*C** effect.

In the MIXED procedure, the **S** option, for example, applies to all **RANDOM** statements.

- If you select nonmissing values in the *value-list* of the **DDF=** option of the **MODEL** statement, PROC GLIMMIX uses these values to override degrees of freedom for this effect that may be determined otherwise. For example, the statements

```
proc glimmix;
class block a b;
model y = a b a*b / s ddf=4, ., . ddfm=sat;
random block a*block / s;
lsmeans a b a*b / diff;
run;
```

request that the denominator degrees of freedom for tests and confidence intervals involving the **A** effect are set to 4. In the example, this applies to the “Type III Tests of Fixed Effects,” “Least Squares Means,” and “Differences of Least Squares Means” tables.

In the MIXED procedure, the Satterthwaite approximation overrides the **DDF=** specification.

- The **DDFM=BETWITHIN** degrees of freedom method requires that the data be processed by subjects; see the “Processing by Subjects” section.
- The **LSMEANS** statement in the MIXED procedure does not support the following options: **ADJDFE=**, **LINES**, **PLOT=**, **SIMPLEDIFF=**, **SIMPLEDIFFTYPE=**, and the **DIFF=ANOM** type of LS-Means differences.
- When you add the response variable to the **CLASS** statement, PROC GLIMMIX defaults to the multinomial distribution. Adding the response

variable to the CLASS statement in PROC MIXED, it has no effect on the fitted model.

- For ODS purposes, the name of the table for the solution of fixed effects is “SolutionF” in the MIXED procedure. In PROC GLIMMIX, the name of the table that contains fixed effects solutions is “ParameterEstimates.” In generalized linear models, this table also contains scale parameters and overdispersion parameters.

---

## Singly or Doubly Iterative Fitting

Depending on the structure of your model, the GLIMMIX procedure determines the appropriate approach for estimating the parameters of the model. The elementary algorithms fall into three categories:

### 1. Noniterative algorithms

A closed form solution exists for all model parameters. Standard linear models with homoscedastic, uncorrelated errors can be fit with noniterative algorithms.

### 2. Singly iterative algorithms

A single optimization, consisting of one or more iterations, is performed to obtain solutions for the parameter estimates by numerical techniques. Linear mixed models for normal data can be fit with singly iterative algorithms.

### 3. Doubly iterative algorithms

A model of simpler structure is derived from the target model. The parameters of the simpler model are estimated by noniterative or singly iterative methods. Based on these new estimates, the model of simpler structure is re-derived and another estimation step follows. The process continues until changes in the parameter estimates are sufficiently small between two re-computations of the simpler model, or until some other criterion is met. The re-derivation of the model can often be cast as a change of the response to some pseudo-data along with an update of implicit model weights.

Obviously, noniterative algorithms are preferable to singly iterative ones, which in turn are preferable to doubly iterative algorithms. Two drawbacks of doubly iterative algorithms based on linearization are the fact that likelihood-based measures apply to the pseudo-data, not the original data, and that at the outer level the progress of the algorithm is tied to monitoring the parameter estimates. The advantage of doubly iterative algorithms, however, is to offer—at convergence—the statistical inference tools that apply to the simpler models.

The output and LOG messages contain information about which algorithm is employed. For a noniterative algorithm, PROC GLIMMIX produces a note that no optimization was performed. Noniterative algorithms are employed automatically for normal data with identity link.

You can determine whether a singly or doubly iterative algorithm was used, based on the “Iteration History” table and the “Convergence Status” table (Figure 15).

The GLIMMIX Procedure					
Iteration History					
Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	83.039723731	.	13.63536
1	0	3	82.189661988	0.85006174	0.281308
2	0	3	82.189255211	0.00040678	0.000174
3	0	3	82.189255211	0.00000000	1.05E-10

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

**Figure 15.** Iteration History and Convergence Status in Singly Iterative Fit

The “Iteration History” table contains the **Evaluations** column that shows how many function evaluations were performed in a particular iteration. The convergence status message informs you which convergence criterion was met when the estimation process concluded. In a singly iterative fit, the criterion is one that applies to the optimization. In other words, it is one of the criteria that can be controlled with the **NLOPTIONS** statement: see the **ABSCONV=**, **ABSFCONV=**, **ABSGCONV=**, **ABSXCONV=**, **FCONV=**, or **GCONV=** option.

In a doubly iterative fit, the “Iteration History” table does not contain an **Evaluations** column. Instead it displays the number of iterations within an optimization (column **Subiterations** in [Figure 16](#)).

Iteration History					
Iteration	Restarts	Subiterations	Objective Function	Change	Max Gradient
0	0	5	79.688580269	0.11807224	7.851E-7
1	0	3	81.294622554	0.02558021	8.209E-7
2	0	2	81.438701534	0.00166079	4.061E-8
3	0	1	81.444083567	0.00006263	2.311E-8
4	0	1	81.444265216	0.0000421	0.000025
5	0	1	81.444277364	0.00000383	0.000023
6	0	1	81.444266322	0.00000348	0.000021
7	0	1	81.44427636	0.00000316	0.000019
8	0	1	81.444267235	0.00000287	0.000017
9	0	1	81.444275529	0.00000261	0.000016
10	0	1	81.44426799	0.00000237	0.000014
11	0	1	81.444274843	0.00000216	0.000013
12	0	1	81.444268614	0.00000196	0.000012
13	0	1	81.444274277	0.00000178	0.000011
14	0	1	81.444269129	0.00000162	9.772E-6
15	0	0	81.444273808	0.00000000	9.102E-6

Convergence criterion (PCONV=1.11022E-8) satisfied.

**Figure 16.** Iteration History and Convergence Status in Doubly Iterative Fit



The `Iteration` column then counts the number of optimizations. The “Convergence Status” table indicates that the estimation process concludes when a criteria is met that monitors the parameter estimates across optimization, namely the `PCONV=` or `ABSPCONV=` criterion.

You can control the optimization process with the `GLIMMIX` procedure through the `NLOPTIONS` statement. Its options affect the individual optimizations. In a doubly iterative scheme, these apply to all optimizations.

The default optimization techniques are `TECHNIQUE=NONE` for noniterative estimation, `TECHNIQUE=NEWRAP` for singly iterative methods in GLMs, and `TECHNIQUE=QUANEW` for mixed models.

---

## Default Estimation Techniques

Based on the structure of the model, the `GLIMMIX` procedure selects the estimation technique for estimating the model parameters. If you fit a generalized linear mixed model, you can change the estimation technique with the `METHOD=` option in the `PROC GLIMMIX` `GLIMMIX` statement. The defaults are determined as follows:

- generalized linear model
  - normal distribution: restricted maximum likelihood
  - all other distributions: maximum likelihood
- generalized linear model with overdispersion
 

Parameters ( $\beta$ ;  $\phi$ , if present) are estimated by (restricted) maximum likelihood as for generalized linear models. The overdispersion parameter is estimated from the Pearson statistic after all other parameters have been estimated.
- generalized linear mixed models
 

The default technique is `METHOD=RSPL`, corresponding to maximizing the residual log pseudo-likelihood with an expansion about the current solutions of the best linear unbiased predictors of the random effects. In models for normal data with identity link, `METHOD=RSPL` or `METHOD=RMPL` are equivalent to restricted maximum likelihood estimation, and `METHOD=MSPL` or `METHOD=MMPL` are equivalent to maximum likelihood estimation. This is reflected in the labeling of statistics in the “Fit Statistics” table.

---

## 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 may 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 more easily terminate at stationary points rather than at global optima.

Table 18 shows which derivatives are required for each optimization technique (FOD: first-order derivatives (=gradient evaluation); SOD: second-order derivatives (=Hessian evaluation)).

**Table 18.** First-order and Second-order Derivatives

Algorithm	FOD	SOD
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. The QUANEW and DBLDOG algorithms, in general, 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. The CONGRA algorithm, in general, requires more iterations than QUANEW or DBLDOG, but each iteration can be much faster. Since 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 employs one or more convergence criteria that determine when it has converged. An algorithm is considered to have converged when any one of the convergence criterion is satisfied. For example, under the default set-

tings, 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 (TRUREG)

The trust region method uses the gradient  $g(\psi^{(k)})$  and the Hessian matrix  $H(\psi^{(k)})$ ; thus, it requires that the objective function  $f(\psi)$  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 corresponding to the quality of the quadratic approximation. The trust region method is implemented using 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 may be more efficient.

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

The NEWRAP technique uses the gradient  $g(\psi^{(k)})$  and the Hessian matrix  $H(\psi^{(k)})$ ; 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 may 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  $g(\psi^{(k)})$  and the Hessian matrix  $H(\psi^{(k)})$ ; 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 may be more efficient.

Since 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  $g(\psi^{(k)})$ , and it does not need to compute second-order derivatives since they are approximated. It works well for medium 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 (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  $\alpha$  satisfying 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 performs its own line-search technique. All options and parameters (except the INSTEP= option) controlling the line search in the other algorithms do not apply 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 `INHESIAN[=r]` option to specify a different starting approximation for the Hessian. If you specify only the `INHESIAN` option, the Cholesky factor of a (possibly ridged) finite difference approximation of the Hessian is used to initialize the quasi-Newton update process. The values of the `LCSINGULAR=`, `LCEPSILON=`, and `LCDEACT=` options, which control the processing of linear and boundary constraints, are valid only for the quadratic programming subroutine used in each iteration of the `QUANEW` algorithm.

### Double Dogleg Optimization (DBLDOG)

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

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

The step is requested to remain within a prespecified trust region radius; refer to 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 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`, or `NRRIDG` technique, which requires 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  $\alpha$  satisfying 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 may 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 criteria.

---

## Remote Monitoring

The SAS/EmMonitor is an application for Windows that enables you to monitor a CPU-intensive application running on a remote server. The GLIMMIX procedure supports remote monitoring through its `NLOPTIONS` statement.

On the server side, a `FILENAME` statement assigns a fileref to a `SOCKET`-type device that defines the IP address of the client and the port number for listening. The fileref is then specified in the `SOCKET=` option of the `NLOPTIONS` statement to control the EmMonitor. The following statements show an example of server-side code.

```
filename sock socket 'your.pc.address.com:6943';

proc glimmix data=bigdataset;
  class year;
  model y1 = x1 x2 x3 / dist=gamma link=log;
  random x1 x3 / type=rsmooth knotmethod=kdtree(bucket=8)
```

```

subject=year;
nloptions tech=nrridg gconv=2.e-5 socket=sock;
run;

```

On the client side, the SAS/EmMonitor application is started with the following syntax:

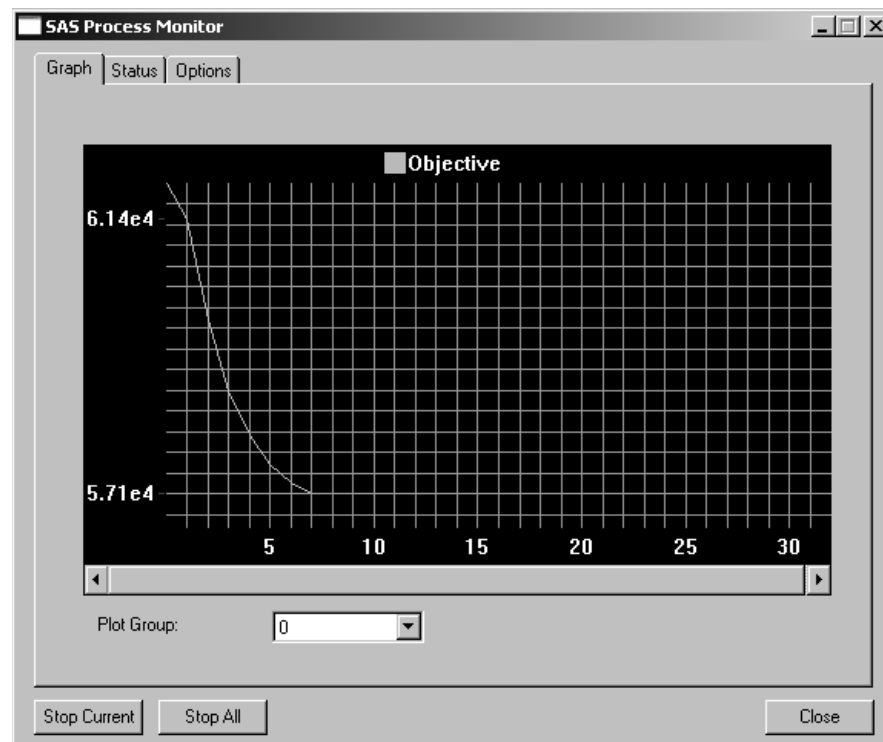
### EmMonitor options

The options are

- p port\_number    define the port number
- t title            define the title of the EmMonitor window
- k                 keep the monitor alive when the iteration is completed

The default port number is 6943.

By default, the PC Client displays on its Graph tab the objective function of the optimization process as the first *Plot Group*, [Figure 17](#). The largest absolute gradient can be monitored as the next *Plot Group*.



**Figure 17.** Graph Tab of SAS/EmMonitor PC Client

You can intervene in the optimization process through the *Stop Current* and *Stop All* buttons. Signaling the server by clicking these buttons effectively terminates the optimization. In a singly iterative optimization, clicking either of the two buttons halts the optimization process when the signal is received on the server side. The current parameter estimates are accepted, and post-optimization processing is based on these

values. Note that this is different from a condition where the usual termination criteria is not met; for example, when the maximum number of iterations is exceeded. If an optimization does not meet the termination criteria, no post-processing occurs.

In doubly iterative processes, the *Stop All* button terminates the overall optimization process and accepts the current estimates. The *Stop Current* button stops the current optimization and uses the current parameter estimates to start the next optimization.

The server does not need to be running when you start the EmMonitor, and you can start or dismiss it at any time during the iteration process. You only need to remember the port number.

If you do not start the PC client, or you close it prematurely, it will have no effect on the server side. In other words, the iteration process will continue until one of the criteria for termination is met.

---

## Default Output

The following sections describe the output that PROC GLIMMIX produces by default. The output is organized into various tables, which are discussed in the order of appearance. Note that the contents of a table may change with the estimation method or the model being fit.

### **Model Information**

The “Model Information” table displays basic information about the fitted model, such as the link and variance functions, the distribution of the response, and the data set. If important model quantities—for example, the response, weights, link, or variance function—are user-defined, the “Model Information” table displays the final assignment to the respective variable, as determined from your programming statements. If the table indicates that the variance matrix is blocked by an effect, then PROC GLIMMIX processes the data by subjects. The “Dimensions” table displays the number of subjects. For more information about processing by subjects, see the section “[Processing by Subjects](#)” on page 129. For ODS purposes, the name of the “Model Information” table is “ModelInfo.”

### **Class Level Information**

The “Class Level Information” table lists the levels of every variable specified in the `CLASS` statement. You should check this information to make sure that the data are correct. You can adjust the order of the `CLASS` variable levels with the `ORDER=` option in the PROC GLIMMIX statement. For ODS purposes, the name of the “Class Level Information” table is “ClassLevels.”

### **Number of Observations**

The “Number of Observations” table displays the number of observations read from the input data set and the number of observations used in the analysis. If you specify a `FREQ` statement, the table also displays the sum of frequencies read and used. If the *events/trials* syntax is used for the response, the table furthermore displays the number of events and trials used in the analysis. For ODS purposes, the name of the “Number of Observations” table is “NObs.”



## Response Profile

For binary and multinomial data, the “Response Profile” table displays the Ordered Value from which the GLIMMIX procedure determines

- the probability being modeled for binary data
- the ordering of categories for ordinal data
- the reference category for generalized logit models

For each response category level, the frequency used in the analysis is reported. The section “[Response Level Ordering and Referencing](#)” on page 148 explains how you can use the [DESCENDING](#), [EVENT=](#), [ORDER=](#), and [REF=](#) options to affect the assignment of Ordered Values to the response categories. For ODS purposes, the name of the “Response Profile” table is “ResponseProfile.”

## Dimensions

The “Dimensions” table displays information from which you can determine the size of relevant matrices in the model. This table is useful in determining CPU time and memory requirements. For ODS purposes, the name of the “Dimensions” table is “Dimensions.”

## Optimization Information

The “Optimization Information” table displays important details about the optimization process.

The optimization technique that is displayed in the table is the technique that applies to any single optimization. For singly iterative methods that is *the* optimization method.

The number of parameters that are updated in the optimization, equals the number of parameters in this table minus the number of equality constraints. The number of constraints are displayed if you fix covariance parameters with the [HOLD=](#) option of the [PARMS](#) statement. The GLIMMIX procedure also lists the number of upper and lower boundary constraints. Note that the procedure may impose boundary constraints for certain parameters, for example, variance components and correlation parameters. Covariance parameters for which a [HOLD=](#) was issued have an upper and lower boundary equal to the parameter value.

If a residual scale parameter is profiled from the optimization, it is also shown in the “Optimization Information” table.

In a GLMM for which the parameters are estimated by one of the linearization methods, you need to initiate the process of computing the pseudo-response. This can be done based on existing estimates of the fixed-effects, or by using the data itself—possibly after some suitable adjustment—as an estimate of the initial mean. The default in PROC GLIMMIX is to use the data itself to derive initial estimates of the mean function and to construct the pseudo-data. The “Optimization Information” table shows how the pseudo-data are determined initially. Note that this issue is separate from the determination of starting values for the covariance parameters. These

are computed as minimum variance quadratic unbiased estimates (with 0 priors, MIVQUE0, Goodnight 1978) or obtained from the *value-list* in the **PARMS** statement.

For ODS purposes, the name of the table is “OptInfo.”

### **Iteration History**

The “Iteration History” table describes the progress of the estimation process. In singly iterative methods, the table displays

- the iteration count, **Iteration**
- the number of restarts, **Restarts**
- the number of function evaluations, **Evaluations**
- the objective function,
- the change in the objective function, **Change**
- the absolute value of the largest (projected) gradient, **MaxGradient**

Note that the change in the objective function is not the convergence criterion monitored by the GLIMMIX procedure. PROC GLIMMIX tracks several convergence criteria simultaneously; see the **ABSCONV=**, **ABSFCONV=**, **ABSGCONV=**, **ABSXCONV=**, **FCONV=**, or **GCONV=** options in the **NLOPTIONS** statement.

For doubly iterative estimation methods, the “Iteration History” table does not display the progress of the individual optimizations; instead, it reports on the progress of the outer iterations. Every row of the table then corresponds to an update of the linearization, the computation of a new set of pseudo-data, and a new optimization. In the listing, PROC GLIMMIX displays

- the optimization count, **Iteration**
- the number of restarts, **Restarts**
- the number of iterations per optimization, **Subiterations**
- the change in the parameter estimates, **Change**
- the absolute value of the largest (projected) gradient at the end of the optimization, **MaxGradient**

By default, the change in the parameter estimates is expressed in terms of the relative **PCONV** criterion. If you request an absolute criterion with the **ABSPCONV** option of the **PROC GLIMMIX** statement, the change reflects the largest absolute difference since the last optimization.

If you specify the **ITDETAILS** option of the **PROC GLIMMIX** statement, parameter estimates and their gradients are added to the “Iteration History” table. For ODS purposes, the name of the “Iteration History” table is “IterHistory.”

### Convergence Status

The “Convergence Status” table contains a status message describing the reason for termination of the optimization. The message is also written to the LOG. For ODS purposes, the name of the “Convergence Status” table is “ConvergenceStatus,” and you can query the nonprinting numeric variable **Status** to check for a successful optimization. This is useful in batch processing, or when processing BY groups, for example, in simulations. Successful optimizations are indicated by the value 0 of the **Status** variable.

### Fit Statistics

The “Fit Statistics” table provides statistics about the estimated model. The first entry of the table corresponds to the negative of twice the (possibly restricted) log likelihood, log pseudo-likelihood, or log quasi-likelihood. If the estimation method permits the true log likelihood or residual log likelihood, the description of the first entry reads accordingly. Otherwise, the fit statistics are preceded by the words *Pseudo-* or *Quasi-*, for Pseudo- and Quasi-Likelihood estimation, respectively.

Note that the (residual) log pseudo-likelihood in a GLMM is the (residual) log likelihood of a linearized model. You should not compare these values across different statistical models, even if the models are nested with respect to fixed and/or G-side random effects. It is possible that between two nested models the larger model has a smaller pseudo-likelihood. For this reason, IC=NONE is the default for GLMMs fit by pseudo-likelihood methods.

See the **IC=** option of the **PROC GLIMMIX** statement and [Table 1](#) for the definition and computation of the information criteria reported in the “Fit Statistics” table.

For generalized linear models, the GLIMMIX procedure reports Pearson’s chi-square statistic

$$X^2 = \sum_i \frac{w_i (y_i - \hat{\mu}_i)^2}{a(\hat{\mu}_i)}$$

where  $a(\hat{\mu}_i)$  is the variance function evaluated at the estimated mean.

For GLMMs, the procedure typically reports a generalized chi-square statistic,

$$X_g^2 = \hat{\mathbf{r}}' \mathbf{V}(\hat{\boldsymbol{\theta}}^*)^{-1} \hat{\mathbf{r}}$$

so that the ratio of  $X^2$  or  $X_g^2$  and the degrees of freedom produces the usual residual dispersion estimate.

If the R-side scale parameter  $\phi$  is not extracted from **V**, the GLIMMIX procedure computes

$$X_g^2 = \hat{\mathbf{r}}' \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1} \hat{\mathbf{r}}$$

as the generalized chi-square statistic. This is the case, for example, if R-side covariance structures are varied by a **GROUP=** effect or if the scale parameter is not profiled

for R-side `TYPE=CS`, `TYPE=SP`, `TYPE=AR`, `TYPE=TOEP`, or `TYPE=ARMA` covariance structures.

If your model contains smooth components (e.g., `TYPE=RSMOOTH`), then the “Fit Statistics” table also displays the residual degrees of freedom of the smoother. These degrees of freedom are computed as

$$df_{smooth,res} = f - \text{trace}(\mathbf{S})$$

where  $\mathbf{S}$  is the “smoother” matrix, that is, the matrix that produces the predicted values on the linked scale.

For ODS purposes, the name of the “Fit Statistics” table is “FitStatistics.”

### Covariance Parameter Estimates

In a GLMM, the “Covariance Parameter Estimates” table displays the estimates of the covariance parameters and their asymptotic standard errors. This table is only produced for generalized linear *mixed* models. In generalized linear models with scale parameter, or when an overdispersion parameter is present, the estimates of parameters related to the dispersion are displayed in the “Parameter Estimates” table.

The standard error of the covariance parameters are determined from the diagonal entries of the asymptotic variance matrix of the covariance parameter estimates. You can display this matrix with the `ASYCOV` option in the `PROC GLIMMIX` statement.

For ODS purposes, the name of the “Covariance Parameter Estimates” table is “CovParms.”

### Type III Tests of Fixed Effects

The “Type III Tests of Fixed Effects” table contains hypothesis tests for the significance of each of the fixed effects specified in the `MODEL` statement. By default, `PROC GLIMMIX` computes these tests by first constructing a Type III  $\mathbf{L}$  matrix for each effect; see Chapter 11, “The Four Types of Estimable Functions” (*SAS STAT User’s Guide*). The  $\mathbf{L}$  matrix is then used to construct the test statistic

$$F = \frac{\hat{\boldsymbol{\beta}}' \mathbf{L}' (\mathbf{L} \mathbf{Q} \mathbf{L}')^{-1} \mathbf{L} \hat{\boldsymbol{\beta}}}{\text{rank}(\mathbf{L})}$$

where the matrix  $\mathbf{Q}$  depends on the estimation method and options. For example, in a GLMM, the default is  $\mathbf{Q} = (\mathbf{X}' \mathbf{V}(\hat{\boldsymbol{\theta}})^{-1} \mathbf{X})^{-}$ , where  $\mathbf{V}(\boldsymbol{\theta})$  is the marginal variance of the pseudo-response. If you specify the `DDFM=KENWARDROGER` option,  $\mathbf{Q}$  is the estimated variance matrix of the fixed effects, adjusted by the method of Kenward and Roger (1997). If the `EMPIRICAL=` option is in effect,  $\mathbf{Q}$  corresponds to the selected sandwich estimator.

You can use the `HTYPE=` option in the `MODEL` statement to obtain tables of Type I (sequential) tests and Type II (adjusted) tests in addition to or instead of the table of Type III (partial) tests.

For ODS purposes, the name of the “Type I Tests of Fixed Effects” through the “Type III Tests of Fixed Effects” tables are “Tests1” through “Tests3,” respectively.

## Notes on Output Statistics

Table 8 on page 87 lists the statistics computed with the OUTPUT statement of the GLIMMIX procedure and their default names. This section provides further details about these statistics.

The distinction between prediction and confidence limits in Table 8 (page 87) stems from the involvement of the predictors of the random effects. If the BLUPs are involved, then the associated standard error used in computing the limits are standard errors of prediction, rather than standard errors of estimation. The prediction limits are *not* limits for the prediction of a new observation.

The Pearson residuals in Table 8 (page 87) are “Pearson-type” residuals, since the residuals are standardized by the square root of the marginal or conditional variance of an observation. Traditionally, Pearson residuals in generalized linear models are divided by the square root of the variance function. The GLIMMIX procedure divides by the square root of the variance so that marginal and conditional residuals have similar expressions. In other words, scale and overdispersion parameters are included.

When residuals or predicted values involve only the fixed effects part of the linear predictor, that is,  $\hat{\eta}_m = \mathbf{x}'\hat{\boldsymbol{\beta}}$ , then all model quantities are computed based on this predictor. For example, if the variance by which to standardize a marginal residual involves the variance function, then the variance function is also evaluated at the marginal mean,  $g^{-1}(\hat{\eta}_m)$ . If the predictor involves the BLUPs, then all relevant expressions and evaluations involve the conditional mean  $g^{-1}(\hat{\eta})$ .

The naming convention to add “PA” to quantities not involving the BLUPs is chosen to appeal to the term “Population-Averaged.” When the link function is nonlinear, these are not truly population-averaged quantities, because  $g^{-1}(\mathbf{x}'\boldsymbol{\beta})$  does not equal  $E[Y]$  in the presence of random effects. For example, if

$$\mu_i = g^{-1}(\mathbf{x}'_i\boldsymbol{\beta} + \mathbf{z}'_i\boldsymbol{\gamma}_i)$$

is the conditional mean for subject  $i$ , then

$$g^{-1}(\mathbf{x}'_i\hat{\boldsymbol{\beta}})$$

does not estimate the average response in the population of subjects, but the response of the average subject (the subject for which  $\boldsymbol{\gamma}_i = \mathbf{0}$ ). For models with identity link, the average response and the response of the average subject are identical.

The GLIMMIX procedure obtains standard errors on the scale of the mean by the Delta method. If the link is a nonlinear function of the linear predictor, these standard errors are only approximate. For example,

$$\text{var}[g^{-1}(\hat{\eta}_m)] \doteq \left( \frac{\partial g^{-1}(t)}{\partial t} \Big|_{\hat{\eta}_m} \right)^2 \text{var}[\hat{\eta}_m]$$

Confidence limits on the scale of the data are usually computed by applying the inverse link function to the confidence limits on the linked scale. The resulting limits on the data scale have the same coverage probability as the limits on the linked scale, but are possibly asymmetric.

In generalized logit models confidence limits on the mean scale are based on symmetric limits about the predicted mean in a category. Suppose that the multinomial response in such a model has  $J$  categories. The probability of a response in category  $i$  is computed as

$$\hat{\mu}_i = \frac{\exp\{\hat{\eta}_i\}}{\sum_{j=1}^J \exp\{\hat{\eta}_j\}}$$

The variance of  $\hat{\mu}_i$  is then approximated as

$$\text{var}[\hat{\mu}_i] \doteq \zeta = \boldsymbol{\varpi}'_i \text{var} \begin{bmatrix} \hat{\eta}_1 & \hat{\eta}_2 & \cdots & \hat{\eta}_J \end{bmatrix} \boldsymbol{\varpi}_i$$

where  $\boldsymbol{\varpi}_i$  is a  $J \times 1$  vector with  $k$ th element

$$\begin{aligned} \hat{\mu}_i(1 - \hat{\mu}_i) & \quad i = k \\ -\hat{\mu}_i\hat{\mu}_k & \quad i \neq k \end{aligned}$$

The confidence limits in the generalized logit model are then obtained as

$$\hat{\mu}_i \pm t_{\nu, \alpha/2} \sqrt{\zeta}$$

where  $t_{\nu, \alpha/2}$  is the  $100 * (1 - \alpha/2)$  percentile from a  $t$  distribution with  $\nu$  degrees of freedom. Confidence limits are truncated if they fall outside the  $[0, 1]$  interval.

---

## ODS Table Names

Each table created by PROC GLIMMIX has a name associated with it, and you must use this name to reference the table when using ODS statements. These names are listed in [Table 19](#).

**Table 19.** ODS Tables Produced by PROC GLIMMIX

Table Name	Description	Required Statement / Option
AsyCorr	asymptotic correlation matrix of covariance parameters	PROC GLIMMIX ASYCORR
AsyCov	asymptotic covariance matrix of covariance parameters	PROC GLIMMIX ASYCOV
CholG	Cholesky root of the estimated $\mathbf{G}$ matrix	RANDOM / GC
CholV	Cholesky root of blocks of the estimated $\mathbf{V}$ matrix	RANDOM / VC
ClassLevels	level information from the CLASS statement	default output

Table 19. (continued)

Table Name	Description	Required Statement / Option
Coef	L matrix coefficients	E option on MODEL, CONTRAST, ESTIMATE, or LSMEANS
Contrasts	results from the CONTRAST statements	CONTRAST
ConvergenceStatus	status of optimization at conclusion	default output when optimization performed
CorrB	approximate correlation matrix of fixed-effects parameter estimates	MODEL / CORRB
CovB	approximate covariance matrix of fixed-effects parameter estimates	MODEL / COVB
CovBI	inverse of approximate covariance matrix of fixed-effects parameter es- timates	MODEL / COVBI
CovParms	estimated covariance parameters in GLMMs	default output (in GLMMs)
Diffs	differences of LS-means	LSMEANS / DIFF (or PDIFF)
Dimensions	dimensions of the model	default output
Estimates	results from ESTIMATE statements	ESTIMATE
FitStatistics	fit statistics	default
G	estimated G matrix	RANDOM / G
GCorr	correlation matrix from the estimated G matrix	RANDOM / GCORR
Hessian	Hessian matrix (observed or ex- pected)	PROC GLIMMIX HESSIAN
InvCholG	inverse Cholesky root of the esti- mated G matrix	RANDOM / GCI
InvCholV	inverse Cholesky root of the blocks of the estimated V matrix	RANDOM / VCI
InvG	inverse of the estimated G matrix	RANDOM / GI
InvV	inverse of blocks of the estimated V matrix	RANDOM / VI
IterHistory	iteration history	default output
kdTree	<i>k-d</i> tree information	RANDOM / TYPE=RSMOOTH KNOTMETHOD= KDTREE(TREEINFO)
KnotInfo	knot coordinates of low-rank spline smoother	RANDOM / TYPE=RSMOOTH KNOTINFO
LSMeans	LS-means	LSMEANS
LSMEstimates	estimates among LS-means	LSMESTIMATE
LSMFtest	<i>F</i> test for LSMESTIMATES	LSMESTIMATE / FTEST
LSMLines	lines display for LS-means	LSMEANS / LINES
ModelInfo	model information	default output

Table 19. (continued)

Table Name	Description	Required Statement / Option
NObs	number of observations read and used, number of trials and events	default output
OddsRatios	odds ratios of parameter estimates	MODEL / ODDSRATIO
OptInfo	optimization information	default output
ParameterEstimates	fixed effects solution; overdispersion and scale parameter in GLMs	MODEL / S
ParmSearch	parameter search values	PARMS
ResponseProfile	response categories and category modeled	default output in models with binary or nominal response
Slices	tests of LS-means slices	LSMEANS / SLICE=
SliceDiffs	differences of simple LS-means effects	LSMEANS / SLICEDIFF=
SolutionR	random effects solution vector	RANDOM / S
StandardizedCoefficients	fixed effects solutions from centered and/or scaled model	MODEL / STDCOEF
Tests1	Type 1 tests of fixed effects	MODEL / HTYPE=1
Tests2	Type 2 tests of fixed effects	MODEL / HTYPE=2
Tests3	Type 3 tests of fixed effects	default output
V	blocks of the estimated <b>V</b> matrix	RANDOM / V
VCorr	correlation matrix from the blocks of the estimated <b>V</b> matrix	RANDOM / VCORR

## ODS Graphics (Experimental)

This section describes the use of ODS for creating statistical graphics with the GLIMMIX procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release. To request these graphs you must specify the ODS GRAPHICS statement with an appropriate model, as discussed in the following. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS” (*SAS/STAT User’s Guide*).

The GLIMMIX procedure does not produce default graphics. Statistical graphics are requested with the **PLOTS** option in the PROC GLIMMIX statement or the **PLOTS** option in the LSMEANS statement. In addition to specifying the ODS GRAPHICS statement and the requisite PROC GLIMMIX option(s), you need to select an output destination that supports ODS statistical graphics, for example, the html destination. Typical PROC GLIMMIX code that produces ODS graphics might look like this

```
ods html;
ods graphics on;

proc glimmix plots=diff;
  class A;
  model y = a;
  lsmeans A;
```



```
run;

ods graphics off;
ods html close;
```

When you request graphics in the LSMEANS statement, their options override the specification on the PROC GLIMMIX statement. In addition, certain options are only available through the **PLOTS** option of the LSMEANS statement.

### Statistical Graphics for LS-Mean Comparisons

The following subsections provide information about the ODS graphics for least-squares mean comparisons produced by the GLIMMIX procedure.

Graphical displays of least-squares means related analyses comprise plots of least-squares means and plots of least-squares mean differences. Plots of the least-squares means themselves are requested with the **PLOTS=MEANPLOT** option. You can add confidence limits or confidence bands to these plots, map the means on the data scale, and display interactions through the sub-options of **PLOTS=MEANPLOT**.

Graphics of least-squares mean differences comprise of plots of all pairwise differences (DiffPlot, Diffogram), plots of differences against a control level (ControlPlot), and plots of differences against an average LS-Mean (AnomPlot).

### Pairwise Difference Plot (Diffogram)

The following data set is from an experiment to investigate how snapdragons grow in various soils (Stenstrom 1940). To eliminate the effect of local fertility variations, the experiment is run in blocks, with each soil type sampled in each block. See the “Examples” section of Chapter 32, “The GLM Procedure,” (*SAS/STAT User’s Guide*) for an in-depth analysis of these data.

```
data plants;
  input Type $ @;
  do Block = 1 to 3;
    input StemLength @;
    output;
  end;
datalines;
Clarion 32.7 32.3 31.5
Clinton 32.1 29.7 29.1
Knox 35.7 35.9 33.1
ONeill 36.0 34.2 31.2
Compost 31.8 28.0 29.2
Wabash 38.2 37.8 31.9
Webster 32.5 31.1 29.7
;
```

The following statements perform the analysis of the experiment with the GLIMMIX procedure.

```

ods html;
ods graphics on;
ods select LSMeans DiffPlot;

proc glimmix data=plants order=data plots=Diffplot;
  class Block Type;
  model StemLength = Block Type;
  lsmeans Type;
  lsmeans Type / plots=diffplot(noabs);
run;

ods graphics off;
ods html close;

```

The PLOTS= option in the PROC GLIMMIX statement requests that plots of pairwise least-squares means differences are produced for effects that are listed in corresponding LSMEANS statements. This is the Type effect. The second LSMEANS statement uses the PLOTS= option of the LSMEANS statement to change the default ABS option to NOABS.

The Type least-squares means are shown in Figure 18. Note that the order in which the levels appear corresponds to the order in which they were read from the data set. This was accomplished with the ORDER=DATA option of the PROC GLIMMIX statement.

The GLIMMIX Procedure					
Type Least Squares Means					
Type	Estimate	Standard Error	DF	t Value	Pr >  t
Clarion	32.1667	0.7405	12	43.44	<.0001
Clinton	30.3000	0.7405	12	40.92	<.0001
Knox	34.9000	0.7405	12	47.13	<.0001
ONeill	33.8000	0.7405	12	45.64	<.0001
Compost	29.6667	0.7405	12	40.06	<.0001
Wabash	35.9667	0.7405	12	48.57	<.0001
Webster	31.1000	0.7405	12	42.00	<.0001

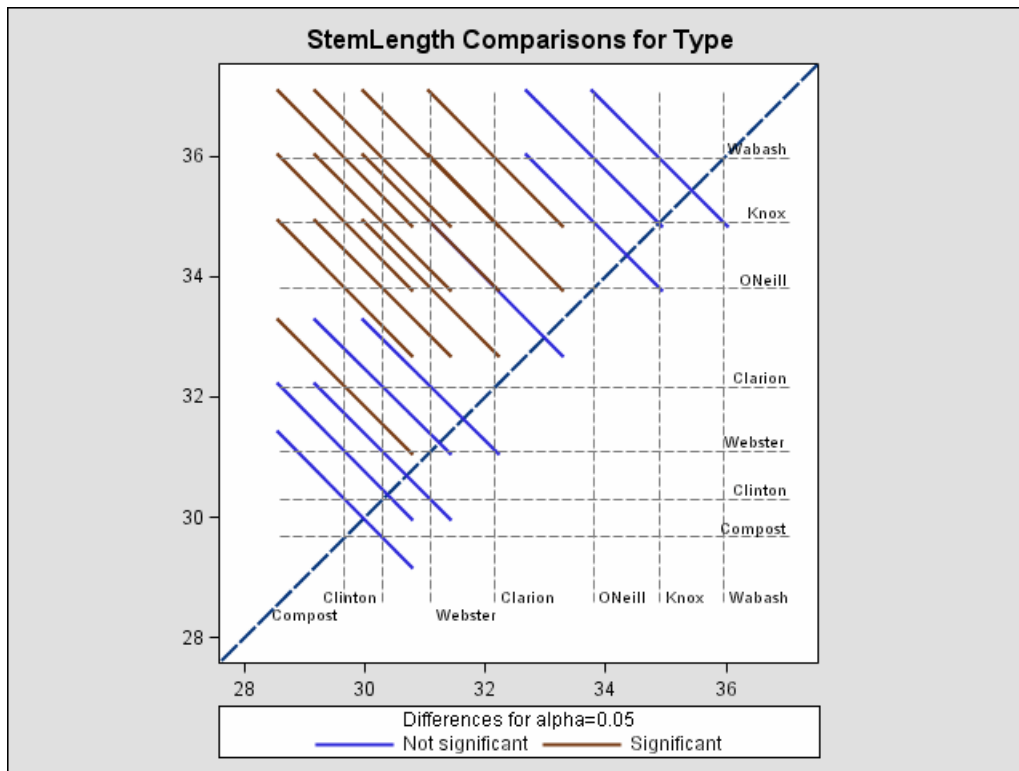
**Figure 18.** Least-squares Means for Type Effect

Because there are seven levels of Type in this analysis, there are  $7(6-1)/2 = 21$  pairwise comparisons among the least-squares means. The comparisons are performed in the following fashion: The first level of Type is compared against levels 2 through 7; the second level of Type is compared against levels 3 through 7; and so forth.

The default difference plot for these data is shown in Figure 19. The display is also known as a “mean-mean scatter plot” when it is based on arithmetic means (Hsu 1996; Hsu and Peruggia 1994). It contains 21 lines rotated by 45 degrees counter clockwise, and a reference line (dashed 45 degree line). The  $(x, y)$  coordinate for the center of each line corresponds to the two least-squares means being compared. Suppose that  $\hat{\eta}_i$  and  $\hat{\eta}_j$  denote the  $i$ th and  $j$ th least-squares mean for

the effect in question, where  $i < j$  according to the ordering of the effect levels. If the ABS option is in effect, which is the default, the line segment is centered at  $(\min\{\hat{\eta}_i, \hat{\eta}_j\}, \max\{\hat{\eta}_i, \hat{\eta}_j\})$ . Take, for example, the comparison of types “Clarion” and “Compost.” The respective estimates of their LS-means are  $\hat{\eta}_{1.1} = 32.1667$  and  $\hat{\eta}_{1.5} = 29.6667$ . The center of the line segment for  $H_0: \eta_{1.1} = \eta_{1.5}$  is placed at  $(29.6667, 32.1667)$ .

The length of the line segment for the comparison between means  $i$  and  $j$  corresponds to the width of the confidence interval for the difference  $\eta_i - \eta_j$ . This length is adjusted for the rotation in the plot. As a consequence, comparisons whose confidence interval covers zero cross the 45 degree reference line. These are the nonsignificant comparisons. Lines associated with significant comparisons do not touch or cross the reference line. Since these data are balanced, the estimated standard errors of all pairwise comparisons are identical, and the widths of the line segments are the same.



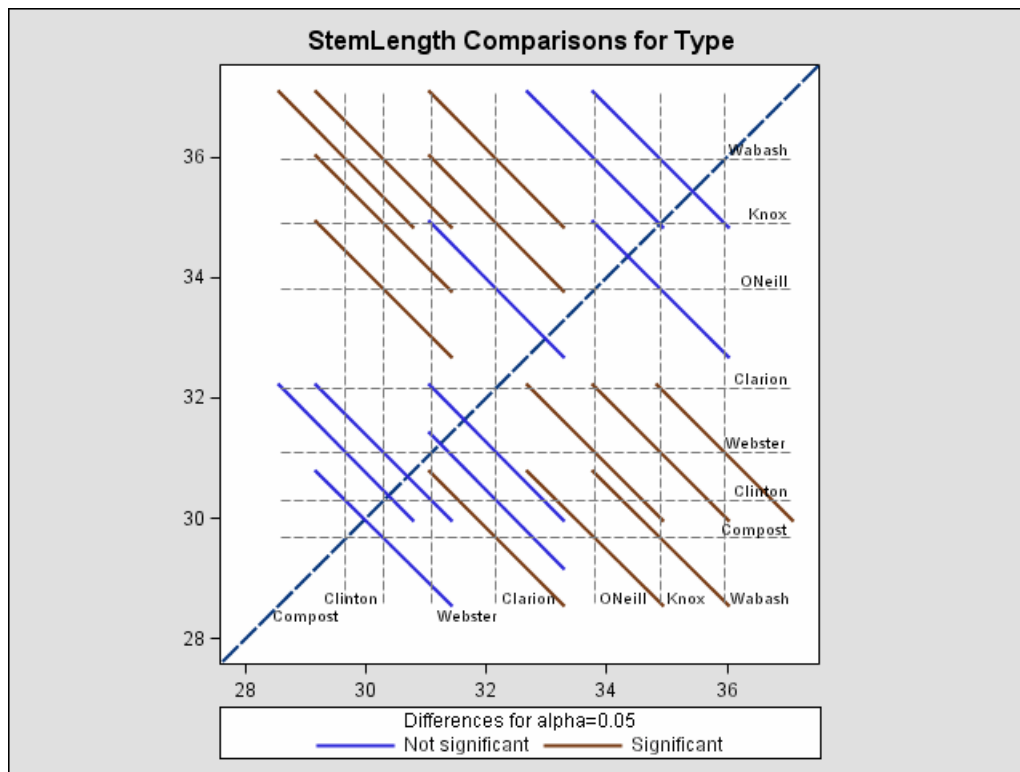
**Figure 19.** LS-means Plot of Pairwise Differences (Experimental)

The background grid of the difference plot is drawn at the values of the least-squares means for the seven *type* levels. These grid lines are used to find a particular comparison by intersection. Also, the labels of the grid lines indicate the ordering of the least-squares means.

The NOABS option of the difference plot changes the way in which the GLIMMIX procedure places the line segments (Figure 20). If the *specific-plot-option* NOABS is in effect, the line segment is centered at the point  $(\hat{\eta}_i, \hat{\eta}_j)$ ,  $i < j$ . For example, the center of the line segment for a comparison of “Clarion” and “Compost” types is centered at  $(\hat{\eta}_{1.1}, \hat{\eta}_{1.5}) = (32.1667, 29.6667)$ . Whether a line segment appears above

or below the reference line depends on the magnitude of the least-squares means and the order of their appearance in the “Least Squares Means” table.

Since the ABS option places lines on the same side of the 45 degree reference, it can help to visually discover groups of significant and nonsignificant differences. On the other hand, when the number of levels in the effect is large, the display can get crowded. The NOABS option can then provide a more accessible resolution.



**Figure 20.** LS-means Plot of Pairwise Differences with NOABS Option (Experimental)

### Least-Squares Mean Control Plot

The following SAS statements create the same data set as before, except that one observation for Type="Knox" has been removed for illustrative purposes.

```
data plants;
  input Type $ @;
  do Block = 1 to 3;
    input StemLength @;
    output;
  end;
  datalines;
  Clarion 32.7 32.3 31.5
  Clinton 32.1 29.7 29.1
  Knox    35.7 35.9 .
  O'Neill 36.0 34.2 31.2
  Compost 31.8 28.0 29.2
  Wabash 38.2 37.8 31.9
```

```
Webster 32.5 31.1 29.7
;
```

The following code requests ControlPlots for effects in LSMEANS statements with compatible option.

```
ods html;
ods graphics on;
ods select DiffS ControlPlot;

proc glimmix data=plants order=data plots=ControlPlot;
  class Block Type;
  model StemLength = Block Type;
  lsmeans Type / diff=control('Clarion') adjust=dunnett;
run;

ods graphics off;
ods html close;
```

The LSMEANS statement for the `Type` effect is compatible; it requests comparisons of `Type` levels against “Clarion,” adjusted for multiplicity with Dunnett’s method. Since “Clarion” is the first level of the effect, the LSMEANS statement is equivalent to

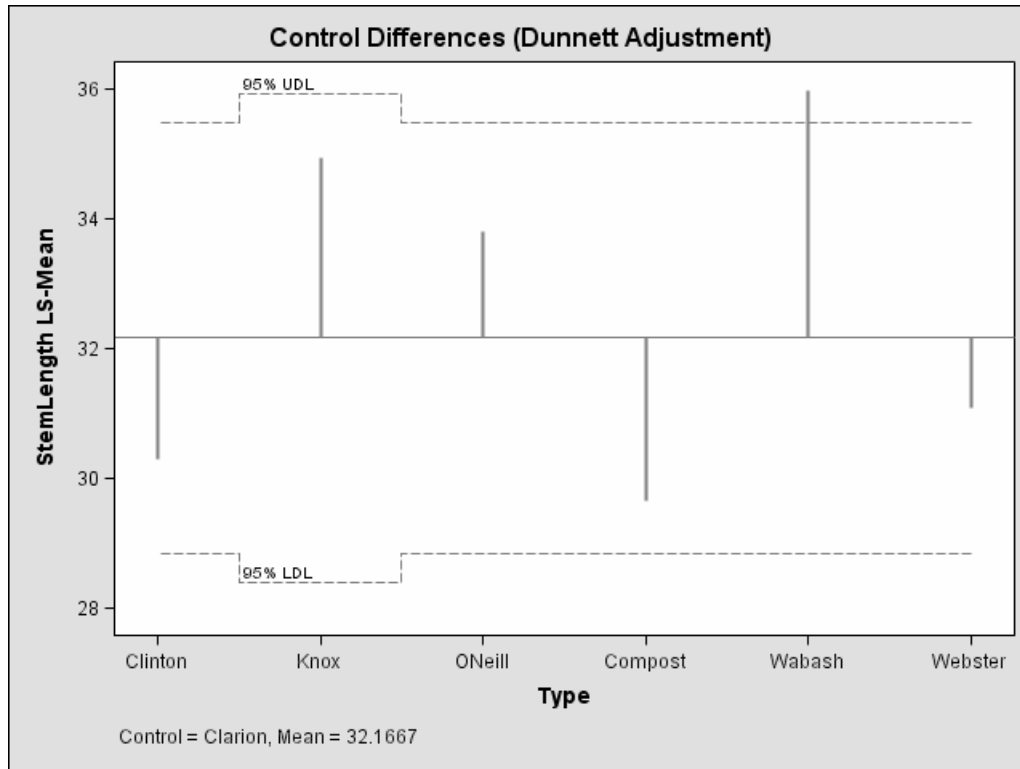
```
lsmeans type / diff=control adjust=dunnett;
```

The “Differences of Type Least Squares Means” table shows the six comparisons between `Type` levels and the control level.

The GLIMMIX Procedure							
Differences of Type Least Squares Means							
Type	_Type	Estimate	Standard Error	DF	t Value	Pr >  t	Adj P
Clinton	Clarion	-1.8667	1.0937	11	-1.71	0.1159	0.3936
Knox	Clarion	2.7667	1.2430	11	2.23	0.0479	0.1854
ONeill	Clarion	1.6333	1.0937	11	1.49	0.1635	0.5144
Compost	Clarion	-2.5000	1.0937	11	-2.29	0.0431	0.1688
Wabash	Clarion	3.8000	1.0937	11	3.47	0.0052	0.0236
Webster	Clarion	-1.0667	1.0937	11	-0.98	0.3504	0.8359

The two rightmost columns of the table give the unadjusted and multiplicity adjusted  $p$ -values. At the 5% significance level, both “Knox” and “Wabash” differ significantly from “Clarion” according to the unadjusted tests. After adjusting for multiplicity, only “Wabash” has a least-squares mean significantly different from the control mean. Note that the standard error for the comparison involving “Knox” is larger than that for other comparisons because of the reduced sample size in that group.

In the plot of control differences a horizontal line is drawn at the value of the “Clarion” least-squares mean. Vertical lines emanating from this reference line terminate in the least-squares means for the other levels (Figure 21).



**Figure 21.** LS-means Plot of Differences Against a Control (Experimental)

The dashed upper and lower horizontal reference lines are the upper and lower decision limits for tests against the control level. If a vertical line crosses the upper or lower decision limit, the corresponding least-squares mean is significantly different from the LS-mean in the control group. If the data had been balanced, the UDL and LDL would be straight lines, since all estimates  $\hat{\eta}_i - \hat{\eta}_j$  would have had the same standard error. The limits for the comparison between “Knox” and “Clarion” are wider than for other comparisons, because of the reduced sample size in the “Knox” group.

The significance level of the decision limits is determined from the ALPHA= level of the LSMEANS statement. The default are 95% limits. If you choose one-sided comparisons with DIFF=CONTROLL or DIFF=CONTROLU in the LSMEANS statement, only one of the decision limits is drawn.

### ODS Graph Names

Each statistical graphic created by PROC GLIMMIX has a name associated with it, and you can reference the graph when using ODS statements. These names are listed in Table 20.

**Table 20.** ODS Graphics Produced by PROC GLIMMIX

<b>ODS Graph Name</b>	<b>Plot Description</b>	<b>Option</b>
AnomPlot	Plot of LS-mean differences against the average LS-mean	PLOTS=ANOMPLOT LSMEANS / PLOTS=ANOMPLOT
ControlPlot	Plot of LS-mean differences against a control level	PLOTS=CONTROLPLOT LSMEANS / PLOTS=CONTROLPLOT
DiffPlot	Plot of LS-mean pairwise differences	PLOTS=DIFFPLOT LSMEANS / PLOTS=DIFFPLOT
MeanPlot	Plot of least-squares means	PLOTS=MEANPLOT LSMEANS / PLOTS=MEANPLOT
PearsonBoxPlot	Box plot of Pearson residuals	PLOTS=PEARSONPANEL(UNPACK)
PearsonbyPredicted	Pearson residuals vs. mean	PLOTS=PEARSONPANEL(UNPACK)
PearsonHistogram	Histogram of Pearson residuals	PLOTS=PEARSONPANEL(UNPACK)
PearsonPanel	Panel of Pearson residuals	PLOTS=PEARSONPANEL
PearsonQQ	<i>Q-Q</i> plot of Pearson residuals	PLOTS=PEARSONPANEL(UNPACK)
ResidualBoxPlot	Box plot of (raw) residuals;	PLOTS=RESIDUALPANEL(UNPACK)
ResidualbyPredicted	Residuals vs. mean or linear predictor	PLOTS=RESIDUALPANEL(UNPACK)
ResidualHistogram	Histogram of (raw) residuals	PLOTS=RESIDUALPANEL(UNPACK)
ResidualPanel	Panel of (raw) residuals	PLOTS=RESIDUALPANEL
ResidualQQ	<i>Q-Q</i> plot of (raw) residuals	PLOTS=RESIDUALPANEL(UNPACK)
StudentBoxPlot	Box plot of studentized residuals	PLOTS=STUDENTPANEL(UNPACK)
StudentbyPredicted	Studentized residuals vs. mean or linear predictor	PLOTS=STUDENTPANEL(UNPACK)
StudentHistogram	Histogram of studentized residuals	PLOTS=STUDENTPANEL(UNPACK)
StudentPanel	Panel of studentized residuals	PLOTS=STUDENTPANEL
StudentQQ	<i>Q-Q</i> plot of studentized residuals	PLOTS=STUDENTPANEL(UNPACK)

---

## Examples

---

### Example 1. Binomial Counts in Randomized Blocks

---

In the context of spatial prediction in generalized linear models, Gotway and Stroup (1997) analyze data from an agronomic field trial. Researchers studied sixteen varieties (entries) of wheat for their resistance to infestation with the Hessian fly. They arranged the varieties in a randomized complete block design on an  $8 \times 8$  grid. Each  $4 \times 4$  quadrant of that arrangement constitutes a block.

The outcome of interest was the number of damaged plants ( $Y_{ij}$ ) out of the total number of plants growing on the unit ( $n_{ij}$ ). The two subscripts identify the block ( $i = 1, \dots, 4$ ) and the entry ( $j = 1, \dots, 16$ ). The following SAS statements create the data set. The variables `lat` and `lng` denote the coordinate of an experimental unit on the  $8 \times 8$  grid.

```

data HessianFly;
  label Y = 'No. of damaged plants'
        n = 'No. of plants';
  input block entry lat lng n Y @@;
  datalines;
1 14 1 1 8 2      1 16 1 2 9 1
1 7 1 3 13 9     1 6 1 4 9 9
1 13 2 1 9 2     1 15 2 2 14 7
1 8 2 3 8 6     1 5 2 4 11 8
1 11 3 1 12 7   1 12 3 2 11 8
1 2 3 3 10 8    1 3 3 4 12 5
1 10 4 1 9 7    1 9 4 2 15 8
1 4 4 3 19 6    1 1 4 4 8 7
2 15 5 1 15 6   2 3 5 2 11 9
2 10 5 3 12 5   2 2 5 4 9 9
2 11 6 1 20 10  2 7 6 2 10 8
2 14 6 3 12 4   2 6 6 4 10 7
2 5 7 1 8 8     2 13 7 2 6 0
2 12 7 3 9 2    2 16 7 4 9 0
2 9 8 1 14 9    2 1 8 2 13 12
2 8 8 3 12 3    2 4 8 4 14 7
3 7 1 5 7 7     3 13 1 6 7 0
3 8 1 7 13 3    3 14 1 8 9 0
3 4 2 5 15 11   3 10 2 6 9 7
3 3 2 7 15 11   3 9 2 8 13 5
3 6 3 5 16 9    3 1 3 6 8 8
3 15 3 7 7 0    3 12 3 8 12 8
3 11 4 5 8 1    3 16 4 6 15 1
3 5 4 7 12 7    3 2 4 8 16 12
4 9 5 5 15 8    4 4 5 6 10 6
4 12 5 7 13 5   4 1 5 8 15 9
4 15 6 5 17 6   4 6 6 6 8 2
4 14 6 7 12 5   4 7 6 8 15 8
4 13 7 5 13 2   4 8 7 6 13 9
4 3 7 7 9 9     4 10 7 8 6 6
4 2 8 5 12 8    4 11 8 6 9 7

```



```

4 5 8 7 11 10 4 16 8 8 15 7
;

```

### Analysis as a GLM

If infestations are independent among experimental units, and all plants within a unit have the same propensity of infestation, then the  $Y_{ij}$  are binomial random variables. The first model considered is a standard generalized linear model for independent binomial counts.

```

proc glimmix data=HessianFly;
  class block entry;
  model y/n = block entry / solution;
run;

```

The **PROC GLIMMIX** statement invokes the procedure. The **CLASS** statement instructs the GLIMMIX procedure to treat both **block** and **entry** as classification variables. The **MODEL** statement specifies the response variable and the fixed effects in the model. PROC GLIMMIX constructs the **X** matrix of the model from the terms in the right-hand side of the **MODEL** statement. The GLIMMIX procedure supports two kinds of syntax for the response variable. This example uses the *events/trials* syntax. The variable **y** represents the number of successes (*events*) out of **n** Bernoulli trials. When the *events/trials* syntax is used, the GLIMMIX procedure automatically selects the binomial distribution as the response distribution. Once the distribution is determined, the procedure selects the link function for the model. The default link for binomial data is the logit link. The preceding statements are thus equivalent to the following statements.

```

proc glimmix data=HessianFly;
  class block entry;
  model y/n = block entry / dist=binomial link=logit solution;
run;

```

The **SOLUTION** option in the **MODEL** statement requests that solutions for the fixed effects (parameter estimates) be displayed.

The “Model Information” table in [Output 1.1](#) describes the model and methods used in fitting the statistical model.

**Output 1.1.** GLM Analysis

The GLIMMIX Procedure	
Model Information	
Data Set	WORK.HESSIANFLY
Response Variable (Events)	Y
Response Variable (Trials)	n
Response Distribution	Binomial
Link Function	Logit
Variance Function	Default
Variance Matrix	Diagonal
Estimation Technique	Maximum Likelihood
Degrees of Freedom Method	Residual

The GLIMMIX procedure recognizes that this is a model for uncorrelated data (variance matrix is diagonal) and that parameters can be estimated by maximum likelihood. The default degrees of freedom method to denominator degrees of freedom for  $F$  tests and  $t$  tests is the RESIDUAL method. This corresponds to choosing  $f - \text{rank}(\mathbf{X})$  as the degrees of freedom, where  $f$  is the sum of the frequencies used in the analysis. You can change the degrees of freedom method with the **DDFM=** option of the **MODEL** statement.

The “Class Level Information” table in [Output 1.2](#) lists the levels of the variables specified in the **CLASS** statement and the ordering of the levels.

**Output 1.2.** GLM Analysis (continued)

Class Level Information		
Class	Levels	Values
block	4	1 2 3 4
entry	16	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

The “Number of Observations” table displays the number of observations read and used in the analysis ([Output 1.3](#)).

**Output 1.3.** GLM Analysis (continued)

Number of Observations Read	64
Number of Observations Used	64
Number of Events	396
Number of Trials	736

The “Dimensions” table lists the size of relevant matrices ([Output 1.4](#)).

**Output 1.4.** GLM Analysis (continued)

Dimensions	
Columns in X	21
Columns in Z	0
Subjects (Blocks in V)	1
Max Obs per Subject	64

Because of the absence of G-side random effects in this model, there are no columns in the **Z** matrix. The 21 columns in the **X** matrix comprise the intercept, 4 columns for the **block** effect and 16 columns for the **entry** effect. Because no **RANDOM** statement with a **SUBJECT=** option was specified, the **GLIMMIX** procedure does not process the data by subjects (see the “[Processing by Subjects](#)” section on page 129 for details on subject processing).

The “[Optimization Information](#)” table provides information about the methods and size of the optimization problem ([Output 1.5](#)).

**Output 1.5.** GLM Analysis (continued)

Optimization Information	
Optimization Technique	Newton-Raphson
Parameters in Optimization	19
Lower Boundaries	0
Upper Boundaries	0
Fixed Effects	Not Profiled

With few exceptions, models fit with the **GLIMMIX** procedure require numerical methods for parameter estimation. The default optimization method for (overdispersed) **GLM** models is the Newton-Raphson algorithm. In this example, the optimization involves 19 parameters, corresponding to the number of linearly independent columns of the  $X'X$  matrix.

The “[Iteration History](#)” table shows that the procedure converged after 3 iterations and 13 function evaluations ([Output 1.6](#)).

**Output 1.6.** GLM Analysis (continued)

Iteration History					
Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	134.13393738	.	4.899609
1	0	3	132.85058236	1.28335502	0.206204
2	0	3	132.84724263	0.00333973	0.000698
3	0	3	132.84724254	0.00000009	3.029E-8

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

The Change column measures the change in the objective function between iterations; however, this is not the convergence criterion monitored. The GLIMMIX procedure monitors several features simultaneously to determine whether to stop an optimization.

The “Fit Statistics” table in [Output 1.7](#) lists information about the fitted model.

**Output 1.7.** GLM Analysis (continued)

Fit Statistics	
-2 Log Likelihood	265.69
AIC (smaller is better)	303.69
AICC (smaller is better)	320.97
BIC (smaller is better)	344.71
CAIC (smaller is better)	363.71
HQIC (smaller is better)	319.85
Pearson Chi-Square	106.74
Pearson Chi-Square / DF	2.37

The -2 Log Likelihood values are useful for comparing nested models, and the information criteria AIC, AICC, BIC, CAIC, and HQIC are useful for comparing nonnested models. On average, the ratio between the Pearson statistic and its degrees of freedom should equal one in GLMs. Values larger than one are indicative of overdispersion. With a ratio of 2.37, these data appear to exhibit more dispersion than expected under a binomial model with block and varietal effects.

In [Output 1.8](#), the “Parameter Estimates” table displays the maximum likelihood estimates (Estimate), standard errors, and *t* tests for the hypothesis that the estimate is zero.

**Output 1.8.** GLM Analysis (continued)

Parameter Estimates							
Effect	block	entry	Estimate	Standard Error	DF	t Value	Pr >  t
Intercept			-1.2936	0.3908	45	-3.31	0.0018
block	1		-0.05776	0.2332	45	-0.25	0.8055
block	2		-0.1838	0.2303	45	-0.80	0.4289
block	3		-0.4420	0.2328	45	-1.90	0.0640
block	4		0	.	.	.	.
entry		1	2.9509	0.5397	45	5.47	<.0001
entry		2	2.8098	0.5158	45	5.45	<.0001
entry		3	2.4608	0.4956	45	4.97	<.0001
entry		4	1.5404	0.4564	45	3.38	0.0015
entry		5	2.7784	0.5293	45	5.25	<.0001
entry		6	2.0403	0.4889	45	4.17	0.0001
entry		7	2.3253	0.4966	45	4.68	<.0001
entry		8	1.3006	0.4754	45	2.74	0.0089
entry		9	1.5605	0.4569	45	3.42	0.0014
entry		10	2.3058	0.5203	45	4.43	<.0001
entry		11	1.4957	0.4710	45	3.18	0.0027
entry		12	1.5068	0.4767	45	3.16	0.0028
entry		13	-0.6296	0.6488	45	-0.97	0.3370
entry		14	0.4460	0.5126	45	0.87	0.3889
entry		15	0.8342	0.4698	45	1.78	0.0826
entry		16	0	.	.	.	.

The Gradient column displays the (projected) gradient of the parameter estimate at convergence of the algorithm. These gradients should be small.

The “Type III Tests of Fixed Effect” table displays significance tests for the two fixed effects in the model (Output 1.9).

**Output 1.9.** GLM Analysis (continued)

Type III Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
block	3	45	1.42	0.2503
entry	15	45	6.96	<.0001

These tests are Wald-type tests, not likelihood ratio tests. The entry effect is clearly significant in this model with a  $p$ -value of  $< 0.0001$ , indicating that the 16 wheat varieties are not equally susceptible to damage by the Hessian fly.

### Analysis with Random Block Effects

There are several possible reasons for the overdispersion noted in Output 1.7 (Pearson ratio = 2.37). The data may not follow a binomial distribution, one or more important effects may have not been accounted for in the model, or the data are positively

correlated. If important fixed effects have been omitted, then you might need to consider adding them to the model. Since this is a designed experiment, it is reasonable not to expect further effects apart from the block and entry effects that represent the treatment and error control design structure. The reasons for the overdispersion must lie elsewhere.

If overdispersion stems from correlations among the observations, then the model should be appropriately adjusted. The correlation can have multiple sources. First, it may not be the case that the plants within an experimental unit responded independently. If the probability of infestation of a particular plant is altered by the infestation of a neighboring plant within the same unit, the infestation counts are not binomial and a different probability model should be used. A second possible source of correlations is the lack of independence of experimental units. Even if treatments were assigned to units at random, they may not respond independently. Shared spatial soil effects, for example, may be the underlying factor. The following analyses take these spatial effects into account.

First, assume that the environmental effects operate at the scale of the blocks. By making the block effects random, the marginal responses will be correlated due to the fact that observations within a block share the same random effects. Observations from different blocks will remain uncorrelated, in the spirit of separate randomizations among the blocks. The next set of SAS statements fits a generalized linear mixed model (GLMM) with random block effects.

```
proc glimmix data=HessianFly;
  class block entry;
  model y/n = entry / solution;
  random block;
run;
```

Because the conditional distribution—conditional on the block effects—is binomial, the marginal distribution will be overdispersed relative to the binomial distribution. In contrast to adding a multiplicative scale parameter to the variance function, treating the block effects as random changes the estimates compared to a model with fixed block effects.

**Output 1.10.** Random Effects Analysis

The GLIMMIX Procedure	
Model Information	
Data Set	WORK.HESSIANFLY
Response Variable (Events)	Y
Response Variable (Trials)	n
Response Distribution	Binomial
Link Function	Logit
Variance Function	Default
Variance Matrix	Not blocked
Estimation Technique	Residual PL
Degrees of Freedom Method	Containment

In the presence of random effects and a conditional binomial distribution, PROC GLIMMIX does not use maximum likelihood for estimation. Instead, the GLIMMIX procedure applies a restricted (residual) pseudo-likelihood algorithm (Output 1.10). The “restricted” attribute derives from the same rationale by which restricted (residual) maximum likelihood methods for linear mixed models attain their name; the likelihood equations are adjusted for the presence of fixed effects in the model to reduce bias in covariance parameter estimates.

The “Class Level Information” and “Number of Observations” tables in Output 1.11 and Output 1.12 are as before.

**Output 1.11.** Random Effects Analysis (continued)

Class Level Information		
Class	Levels	Values
block	4	1 2 3 4
entry	16	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

**Output 1.12.** Random Effects Analysis (continued)

Number of Observations Read	64
Number of Observations Used	64
Number of Events	396
Number of Trials	736

The “Dimensions” table indicates that there is a single G-side parameter, the variance of the random block effect.

**Output 1.13.** Random Effects Analysis (continued)

Dimensions	
G-side Cov. Parameters	1
Columns in X	17
Columns in Z	4
Subjects (Blocks in V)	1
Max Obs per Subject	64

The “Dimensions” table in [Output 1.13](#) has changed from the previous model (compare to [Output 1.4](#)). Note that although the block effect has four levels, only a single variance component is estimated. The **Z** matrix has four columns, however, corresponding to the four levels of the block effect. Because no SUBJECT= option is used in the **RANDOM** statement, the GLIMMIX procedure treats these data as having arisen from a single subject with 64 observations.

In [Output 1.14](#), the “Optimization Information” table indicates that a Quasi-Newton method is used to solve the optimization problem. This is the default method for GLMM models.

**Output 1.14.** Random Effects Analysis (continued)

Optimization Information	
Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	1
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Profiled
Starting From	Data

In contrast to the Newton-Raphson method, the Quasi-Newton method does not require second derivatives. Because the covariance parameters are not unbounded in this example, the procedure enforces a lower boundary constraint (zero) for the variance of the block effect, and the optimization method is changed to a dual quasi-Newton method. The fixed effects are profiled from the likelihood equations in this model. The resulting optimization problem involves only the covariance parameters.

The “Iteration History” table appears to indicate that the procedure converged after four iterations.



**Output 1.15.** Random Effects Analysis (continued)

Iteration History						
Iteration	Restarts	Subiterations	Objective Function	Change	Max Gradient	
0	0	4	173.28473428	0.81019251	0.000197	
1	0	3	181.66726674	0.17550228	0.000739	
2	0	2	182.20789493	0.00614874	7.018E-6	
3	0	1	182.21315596	0.00004386	1.213E-8	
4	0	0	182.21317662	0.00000000	3.349E-6	

Convergence criterion (PCONV=1.11022E-8) satisfied.

Notice, however, that the “Iteration History” table in [Output 1.15](#) has changed slightly from the previous analysis (see [Output 1.6](#)). The Evaluations column has been replaced by the Subiterations column, since the GLIMMIX procedure applied a doubly iterative fitting algorithm. The entire process consisted of five optimizations, each of which was iterative. The initial optimization required four iterations, the next one three iterations, and so on.

In [Output 1.16](#), the “Fit Statistics” table shows information about the fit of the GLMM.

**Output 1.16.** Random Effects Analysis (continued)

Fit Statistics	
-2 Res Log Pseudo-Likelihood	182.21
Generalized Chi-Square	107.96
Gener. Chi-Square / DF	2.25

The log likelihood reported in the table is not the residual log likelihood of the data. It is the residual log likelihood for an approximated model. The generalized chi-square statistic measures the residual sum of squares in the final model and the ratio with its degrees of freedom is a measure of variability of the observation about the mean model.

**Output 1.17.** Random Effects Analysis (continued)

Covariance Parameter Estimates		
Cov Parm	Estimate	Standard Error
block	0.01116	0.03116

The variance of the random block effects is rather small ([Output 1.17](#)). If the environmental effects operate on a spatial scale smaller than the block size, the random block model does not provide a suitable adjustment. From the coarse layout of the experimental area, it is not surprising that random block effects alone do not account for the overdispersion in the data.

Adding a random component to a generalized linear model is different from adding a multiplicative overdispersion component, for example, via the PSCALE option in PROC GENMOD or a

```
random _residual_;
```

statement in PROC GLIMMIX. Such overdispersion components do not affect the parameter estimates, only their standard errors. A genuine random effect, on the other hand, affects both the parameter estimates and their standard errors (compare [Output 1.18](#) to [Output 1.8](#)).

**Output 1.18.** Random Effects Analysis (continued)

Solutions for Fixed Effects						
Effect	entry	Estimate	Standard Error	DF	t Value	Pr >  t
Intercept		-1.4637	0.3738	3	-3.92	0.0296
entry	1	2.9609	0.5384	45	5.50	<.0001
entry	2	2.7807	0.5138	45	5.41	<.0001
entry	3	2.4339	0.4934	45	4.93	<.0001
entry	4	1.5347	0.4542	45	3.38	0.0015
entry	5	2.7653	0.5276	45	5.24	<.0001
entry	6	2.0014	0.4865	45	4.11	0.0002
entry	7	2.3518	0.4952	45	4.75	<.0001
entry	8	1.2927	0.4739	45	2.73	0.0091
entry	9	1.5663	0.4554	45	3.44	0.0013
entry	10	2.2896	0.5179	45	4.42	<.0001
entry	11	1.5018	0.4682	45	3.21	0.0025
entry	12	1.5075	0.4752	45	3.17	0.0027
entry	13	-0.5955	0.6475	45	-0.92	0.3626
entry	14	0.4573	0.5111	45	0.89	0.3758
entry	15	0.8683	0.4682	45	1.85	0.0702
entry	16	0	.	.	.	.

Because the block variance component is small, the Type III test for the variety effect in [Output 1.19](#) is affected only very little compared to the GLM ([Output 1.9](#)).

**Output 1.19.** Random Effects Analysis (continued)

Type III Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
entry	15	45	6.90	<.0001

**Analysis with Smooth Spatial Trends**

You can also consider these data in an observational sense, where the covariation of the observations is subject to modeling. Rather than deriving model components from the experimental design alone, environmental effects can be modeled by adjusting the mean and/or correlation structure. Gotway and Stroup (1997) and Schabenberger and Pierce (2002) supplant the coarse block effects with smooth-scale spatial components.

The model considered by Gotway and Stroup (1997) is a marginal model in that the correlation structure is modeled through residual-side (R-side) random components. This exponential covariance model is fit with the statements

```
proc glimmix data=HessianFly;
  class entry;
  model y/n = entry / solution ddfm=contain;
  random _residual_ / subject=intercept type=sp(exp) (lng lat);
run;
```

Note that the block effects have been removed from the statements. The keyword `_RESIDUAL_` in the `RANDOM` statement instructs the GLIMMIX procedure to model the  $\mathbf{R}$  matrix. Here,  $\mathbf{R}$  is to be modeled as an exponential covariance structure matrix. The `SUBJECT=INTERCEPT` option means that all observations are considered correlated.

**Output 1.20.** Marginal Spatial Analysis

The GLIMMIX Procedure	
Dimensions	
R-side Cov. Parameters	2
Columns in X	17
Columns in Z per Subject	0
Subjects (Blocks in V)	1
Max Obs per Subject	64

Because the random effects are residual-type effects, there are no columns in the  $\mathbf{Z}$  matrix for this model (Output 1.20).

**Output 1.21.** Marginal Spatial Analysis (continued)

Optimization Information	
Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	1
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Profiled
Residual Variance	Profiled
Starting From	Data

In addition to the fixed effects, the GLIMMIX procedure now profiles one of the covariance parameters, the variance of the exponential covariance model ([Output 1.21](#)). This reduces the size of the optimization problem. Only a single parameter is part of the optimization, the “range” (SP(EXP)) of the spatial process.

**Output 1.22.** Marginal Spatial Analysis (continued)

Covariance Parameter Estimates			
Cov Parm	Subject	Estimate	Standard Error
SP (EXP)	Intercept	0.9052	0.4404
Residual		2.5315	0.6974

The practical range of a spatial process is that distance at which the correlation between data points has decreased to at most 0.05. The parameter reported by the GLIMMIX procedure as SP(EXP) in [Output 1.22](#) corresponds to one third of the practical range. The practical range in this process is  $3 \times 0.9052 = 2.7156$ . Correlations extend beyond a single experimental unit, but they do not appear to exist on the scale of the block size.

The sill of the spatial process, the variance of the underlying residual effect, is estimated as 2.5315.

**Output 1.23.** Marginal Spatial Analysis (continued)

Type III Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
entry	15	48	3.60	0.0004

The  $F$  value for the entry effect in [Output 1.23](#) has been sharply reduced compared to the previous analyses. The smooth spatial variation accounts for some of the variation

among the varieties.

In this example three models were considered for the analysis of a randomized block design with binomial outcomes. If data are correlated, a standard generalized linear model often will indicate overdispersion relative to the binomial distribution. Two courses of action are considered in this example to address this overdispersion. The inclusion of G-side random effects models the correlation indirectly; it is induced through the sharing of random effects among responses from the same block. The R-side spatial covariance structure models covariation directly. In generalized linear (mixed) models the two modeling approaches can lead to different inferences, because the models have different interpretation. The random block effects are modeled on the linked (logit) scale, and the spatial effects were modeled on the mean scale. Only in a linear mixed model are the two scales identical.

## Example 2. Mating Experiment with Crossed Random Effects

McCullagh and Nelder (1989, Ch. 14.5) describe a mating experiment—conducted by S. Arnold and P. Verell at the University of Chicago, Department of Ecology and Evolution—involving two geographically isolated populations of mountain dusky salamanders. One goal of the experiment was to determine whether barriers to interbreeding have evolved in light of the geographical isolation of the populations. In this case, matings within a population should be more successful than matings between the populations. The experiment conducted in the summer 1986 involved 40 animals, 20 rough butt (R) and 20 whiteside (W) salamanders, with equal numbers of males and females. The animals were grouped into two sets of R males, two sets of R females, two sets of W males, and two sets of W females, so that each set comprised five salamanders. Each set was mated against one rough butt and one whiteside set, creating eight crossings. Within the pairings of sets, each female was paired to three male animals. The salamander mating data have been used by a number of authors; refer, for example, to McCullagh and Nelder (1989), Schall (1991), Karim and Zeger (1992), Breslow and Clayton (1993), Wolfinger and O’Connell (1993), and Shun (1997).

The following DATA step creates the data set for the analysis.

```

data salamander;
  input day fpop$ fnum mpop$ mnum mating @@;
  datalines;
4  rb  1  rb  1  1  4  rb  2  rb  5  1
4  rb  3  rb  2  1  4  rb  4  rb  4  1
4  rb  5  rb  3  1  4  rb  6  ws  9  1
4  rb  7  ws  8  0  4  rb  8  ws  6  0
4  rb  9  ws 10  0  4  rb 10  ws  7  0
4  ws  1  rb  9  0  4  ws  2  rb  7  0
4  ws  3  rb  8  0  4  ws  4  rb 10  0
4  ws  5  rb  6  0  4  ws  6  ws  5  0
4  ws  7  ws  4  1  4  ws  8  ws  1  1
4  ws  9  ws  3  1  4  ws 10  ws  2  1
8  rb  1  ws  4  1  8  rb  2  ws  5  1
8  rb  3  ws  1  0  8  rb  4  ws  2  1

```

```

8  rb  5  ws  3  1  8  rb  6  rb  9  1
8  rb  7  rb  8  0  8  rb  8  rb  6  1
8  rb  9  rb  7  0  8  rb 10  rb 10  0
8  ws  1  ws  9  1  8  ws  2  ws  6  0
8  ws  3  ws  7  0  8  ws  4  ws 10  1
8  ws  5  ws  8  1  8  ws  6  rb  2  0
8  ws  7  rb  1  1  8  ws  8  rb  4  0
8  ws  9  rb  3  1  8  ws 10  rb  5  0
12 rb  1  rb  5  1 12  rb  2  rb  3  1
12 rb  3  rb  1  1 12  rb  4  rb  2  1
12 rb  5  rb  4  1 12  rb  6  ws 10  1
12 rb  7  ws  9  0 12  rb  8  ws  7  0
12 rb  9  ws  8  1 12  rb 10  ws  6  1
12 ws  1  rb  7  1 12  ws  2  rb  9  0
12 ws  3  rb  6  0 12  ws  4  rb  8  1
12 ws  5  rb 10  0 12  ws  6  ws  3  1
12 ws  7  ws  5  1 12  ws  8  ws  2  1
12 ws  9  ws  1  1 12  ws 10  ws  4  0
16 rb  1  ws  1  0 16  rb  2  ws  3  1
16 rb  3  ws  4  1 16  rb  4  ws  5  0
16 rb  5  ws  2  1 16  rb  6  rb  7  0
16 rb  7  rb  9  1 16  rb  8  rb 10  0
16 rb  9  rb  6  1 16  rb 10  rb  8  0
16 ws  1  ws 10  1 16  ws  2  ws  7  1
16 ws  3  ws  9  0 16  ws  4  ws  8  1
16 ws  5  ws  6  0 16  ws  6  rb  4  0
16 ws  7  rb  2  0 16  ws  8  rb  5  0
16 ws  9  rb  1  1 16  ws 10  rb  3  1
20 rb  1  rb  4  1 20  rb  2  rb  1  1
20 rb  3  rb  3  1 20  rb  4  rb  5  1
20 rb  5  rb  2  1 20  rb  6  ws  6  1
20 rb  7  ws  7  0 20  rb  8  ws 10  1
20 rb  9  ws  9  1 20  rb 10  ws  8  1
20 ws  1  rb 10  0 20  ws  2  rb  6  0
20 ws  3  rb  7  0 20  ws  4  rb  9  0
20 ws  5  rb  8  0 20  ws  6  ws  2  0
20 ws  7  ws  1  1 20  ws  8  ws  5  1
20 ws  9  ws  4  1 20  ws 10  ws  3  1
24 rb  1  ws  5  1 24  rb  2  ws  2  1
24 rb  3  ws  3  1 24  rb  4  ws  4  1
24 rb  5  ws  1  1 24  rb  6  rb  8  1
24 rb  7  rb  6  0 24  rb  8  rb  9  1
24 rb  9  rb 10  1 24  rb 10  rb  7  0
24 ws  1  ws  8  1 24  ws  2  ws 10  0
24 ws  3  ws  6  1 24  ws  4  ws  9  1
24 ws  5  ws  7  0 24  ws  6  rb  1  0
24 ws  7  rb  5  1 24  ws  8  rb  3  0
24 ws  9  rb  4  0 24  ws 10  rb  2  0
;

```

The first observation, for example, indicates that rough butt female 1 was paired in the laboratory on day 4 of the experiment with rough butt male 1 and mated. On the same day rough butt female 7 was paired with whiteside male 8, the pairing did not

result in mating of the animals.

The model adopted by many authors for these data comprises fixed effects for gender and population, their interaction, and male and female random effects. Specifically, let  $\pi_{RR}$ ,  $\pi_{RW}$ ,  $\pi_{WR}$ , and  $\pi_{WW}$  denote the mating probabilities between the populations, where the first subscript identifies the female partner of the pair. Then, we model

$$\log \left\{ \frac{\pi_{kl}}{1 - \pi_{kl}} \right\} = \tau_{kl} + \gamma_f + \gamma_m \quad k, l \in \{R, W\}$$

where  $\gamma_f$  and  $\gamma_m$  are independent random variables representing female and male random effects (20 each), and  $\tau_{kl}$  denotes the average logit of mating between females of population  $k$  and males of population  $l$ .

The following statements fit this model by pseudo-likelihood.

```
proc glimmix data=salamander;
  class fpop fnum mpop mnum;
  model mating(event='1') = fpop|mpop / dist=binary;
  random fpop*fnum mpop*mnum;
  lsmeans fpop*mpop / ilink;
run;
```

The response variable is the two-level variable `mating`. Since it is coded as zeros and ones, and since PROC GLIMMIX models by default the probability of the first level according to the response level ordering, the `EVENT='1'` option instructs PROC GLIMMIX to model the probability of a successful mating. The distribution of the mating variable, conditional on the random effects, is binary.

The `fpop*fnum` effect in the `RANDOM` statement creates a random intercept for each female animal. Because `fpop` and `fnum` are `CLASS` variables, the effect has 20 levels (10 rb and 10 ws females). Similarly, the `mpop*mnum` effect creates the random intercepts for the male animals. Because no `TYPE=` is specified in the `RANDOM` statement, the covariance structure defaults to `TYPE=VC`. The random effects and their levels are independent, and each effect has its own variance component. Since the conditional distribution of the data, conditioned on the random effects, is binary, no extra scale parameter ( $\phi$ ) is added.

The `LSMEANS` statement requests least-squares means for the four levels of the `fpop*mpop` effect, which are estimates of the cell means in the  $2 \times 2$  classification of female and male populations. The `ILINK` option of the `LSMEANS` statement requests that the estimated means and standard errors are also reported on the scale of the data. This yields estimates of the four mating probabilities,  $\pi_{RR}$ ,  $\pi_{RW}$ ,  $\pi_{WR}$ , and  $\pi_{WW}$ .

The “Model Information” table in [Output 2.1](#) displays general information about the model being fit.

**Output 2.1.** Analysis of Mating Experiment with Crossed Random Effects

The GLIMMIX Procedure	
Model Information	
Data Set	WORK.SALAMANDER
Response Variable	mating
Response Distribution	Binary
Link Function	Logit
Variance Function	Default
Variance Matrix	Not blocked
Estimation Technique	Residual PL
Degrees of Freedom Method	Containment

The response variable `mating` follows a binary distribution (conditional on the random effects). Hence, the mean of the data is an event probability,  $\pi$ , and the logit of this probability is linearly related to the linear predictor of the model. The variance function is the default function that is implied by the distribution,  $a(\pi) = \pi(1 - \pi)$ . The variance matrix is not blocked, since the GLIMMIX procedure does not process the data by subjects (see the “[Processing by Subjects](#)” section for details). The estimation technique is the default methods for GLMMs, residual pseudo-likelihood (METHOD=RSPL), and degrees of freedom for tests and confidence intervals are determined by the containment method.

The “Class Level Information” table in [Output 2.2](#) lists the levels of the variables listed in the `CLASS` statement, as well as the order of the levels.

**Output 2.2.** Analysis of Mating Experiment (continued)

Class Level Information		
Class	Levels	Values
<code>fpop</code>	2	rb ws
<code>fnum</code>	10	1 2 3 4 5 6 7 8 9 10
<code>mpop</code>	2	rb ws
<code>mnum</code>	10	1 2 3 4 5 6 7 8 9 10
Number of Observations Read		120
Number of Observations Used		120

Note that there are two female populations and two male populations; also, the variables `fnum` and `mnum` have 10 levels each. As a consequence, the effects `fpop*fnum` and `mpop*mnum` identify the 20 females and males, respectively. The effect `fpop*mpop` identifies the four mating types.



**Output 2.3.** Analysis of Mating Experiment (continued)

Response Profile		
Ordered Value	mating	Total Frequency
1	0	50
2	1	70

The GLIMMIX procedure is modeling the probability that mating='1'.

The “Response Profile Table” in [Output 2.3](#), which is displayed for binary or multinomial data, lists the levels of the response variable and their order. With binary data, the table also provides information about which level of the response variable defines the event. Because of the EVENT='1' response variable option in the [MODEL](#) statement, the probability being modeled is that of the higher ordered value.

**Output 2.4.** Analysis of Mating Experiment (continued)

Dimensions	
G-side Cov. Parameters	2
Columns in X	9
Columns in Z	40
Subjects (Blocks in V)	1
Max Obs per Subject	120

There are two covariance parameters in this model, the variance of the fpop\*fnum effect and the variance of the mpop\*mnum effect ([Output 2.4](#)). Both parameters are modeled as G-side parameters. The nine columns in the **X** matrix comprise the intercept, two columns each for the levels of the fpop and mpop effects, and four columns for their interaction. The **Z** matrix has forty columns, one for each animal. Because the data are not processed by subjects, PROC GLIMMIX assumes the data consist of a single subject (a single block in **V**).

In [Output 2.5](#), the “Optimization Information” table displays basic information about the optimization.

**Output 2.5.** Analysis of Mating Experiment (continued)

Optimization Information	
Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	2
Lower Boundaries	2
Upper Boundaries	0
Fixed Effects	Profiled
Starting From	Data

The default technique for GLMMs is the Quasi-Newton method. There are two parameters in the optimization, which correspond to the two variance components. The 17 fixed effects parameters are not part of the optimization. The initial optimization computes pseudo-data based on the response values in the data set rather than from estimates of a generalized linear model fit.

**Output 2.6.** Analysis of Mating Experiment (continued)

Iteration History						
Iteration	Restarts	Subiterations	Objective Function	Change	Max Gradient	
0	0	6	537.09165908	2.00000000	0.000026	
1	0	5	544.12515308	0.66319387	0.000023	
2	0	3	545.89134774	0.13547016	0.000125	
3	0	3	546.10499422	0.01695914	0.000433	
4	0	2	546.13052996	0.00322192	0.000905	
5	0	2	546.13420974	0.00056273	0.000146	
6	0	1	546.13416244	0.00012776	0.000033	
7	0	1	546.13420573	0.00002626	8.177E-6	
8	0	0	546.13418118	0.00000000	5.177E-6	

Convergence criterion (PCONV=1.11022E-8) satisfied.

The GLIMMIX procedure performs eight optimizations after the initial optimization (Output 2.6). That is, following the initial pseudo-data creation, the pseudo-data were updated eight more times and a total of nine linear mixed models were estimated.

**Output 2.7.** Analysis of Mating Experiment (continued)

Covariance Parameter Estimates		
Cov Parm	Estimate	Standard Error
fpop*fnum	1.4099	0.8871
mpop*mnum	0.08963	0.4102

The Covariance Parameter Estimates table in Output 2.7 lists the estimates for the two variance components and their estimated standard errors. The heterogeneity (in the logit of the mating probabilities) among the females is considerably larger than the heterogeneity among the males.

**Output 2.8.** Analysis of Mating Experiment (continued)

Type III Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
fpop	1	18	2.86	0.1081
mpop	1	17	4.71	0.0444
fpop*mpop	1	81	9.61	0.0027

The “Type III Tests of Fixed Effects” table indicates a significant interaction between the male and female populations (Output 2.8). A comparison in the logits of mating success in pairs with R-females and W-females depends on whether the male partner in the pair is the same species. The “mating Least Squares Means” table in Output 2.9 shows this effect more clearly.

**Output 2.9.** Analysis of Mating Experiment (continued)

fpop*mpop Least Squares Means								
fpop	mpop	Estimate	Standard Error	DF	t Value	Pr >  t	Mean	Standard Error Mean
rb	rb	1.1629	0.5961	81	1.95	0.0545	0.7619	0.1081
rb	ws	0.7839	0.5729	81	1.37	0.1750	0.6865	0.1233
ws	rb	-1.4119	0.6143	81	-2.30	0.0241	0.1959	0.09678
ws	ws	1.0151	0.5871	81	1.73	0.0876	0.7340	0.1146

In a pairing with a male rough butt salamander, the logit drops sharply from 1.1629 to -1.4119 when the male is paired with a whiteside female instead of a female from its own population. The corresponding estimated probabilities of mating success are  $\hat{\pi}_{RR} = 0.7619$  and  $\hat{\pi}_{WR} = 0.1959$ . If the same comparisons are made in pairs with whiteside males, then you also notice a drop in the logit if the female comes from a different population, 1.0151 versus 0.7839. The change is considerably less, though, corresponding to mating probabilities of  $\hat{\pi}_{WW} = 0.7340$  and  $\hat{\pi}_{RW} = 0.6865$ . Whiteside females appear to be successful with their own population. Whiteside males appear to succeed equally well with female partners of the two populations.

This insight into the factor level comparisons can be amplified by graphing the least-squares mean comparisons and by subsetting the differences of least-squares means. This is accomplished with the following statements.

```
ods html;
ods graphics on;
ods select DiffPlot SliceDiffs;

proc glimmix data=salamander;
  class fpop fnum mpop mnum;
  model mating(event='1') = fpop|mpop / dist=binary;
  random fpop*fnum mpop*mnum;
  lsmeans fpop*mpop / plots=(diffplot);
  lsmeans fpop*mpop / slicediff=(mpop fpop);
run;

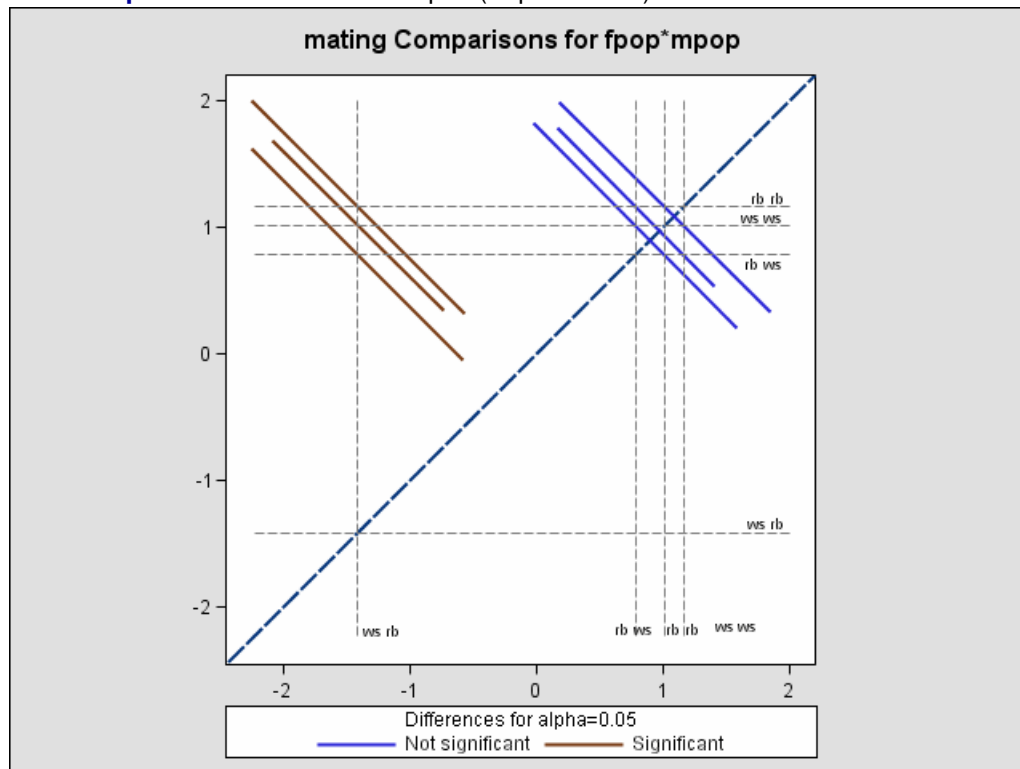
ods graphics off;
ods html close;
```

The `PLOTS=DIFFPLOT` option of the first `LSMEANS` statement requests a comparison plot that displays the result of all pairwise comparisons (Output 2.10). The `SLICEDIFF=(mpop fpop)` option requests differences of simple effects.

The comparison plot in Output 2.10 is also known as a mean-mean scatter plot (Hsu 1996). Each solid line in the plot corresponds to one of the possible  $4 \times 3/2 = 6$  unique pairwise comparisons. The line is centered at the intersection of two least-squares means and the length of the line segments corresponds to the width of a 95% confidence interval for the difference between the two least-squares means. The length of the segment is adjusted for the rotation. If a line segment crosses the dashed 45 degree line, the comparison between the two factor levels is not significant; otherwise, it is significant. The horizontal and vertical axes of the plot are drawn in least-squares means units and the grid lines are placed at the values of the least-squares means.

The six pairs of least-squares means comparisons separate into two sets of three pairs. Comparisons in the first set are significant; comparisons in the second set are not significant. For the significant set, the female partner in one of the pairs is a whiteside salamander. For the nonsignificant comparisons, the male partner in one of the pairs is a whiteside salamander.

**Output 2.10.** LS-means Diffplot (Experimental)



In [Output 2.11](#), the “Simple Effect Comparisons” tables show the results of the `SLICEDIFF=` option in the second `LSMEANS` statement.

**Output 2.11.** Analysis of Mating Experiment (continued)

```

The GLIMMIX Procedure

Simple Effect Comparisons of fpop*mpop Least Squares Means By mpop

Simple
Effect
Level   fpop   _fpop   Estimate   Standard
Error     DF     t Value   Pr > |t|
mpop rb   rb     ws       2.5748    0.8458    81      3.04    0.0031
mpop ws   rb     ws      -0.2312    0.8092    81     -0.29    0.7758

Simple Effect Comparisons of fpop*mpop Least Squares Means By fpop

Simple
Effect
Level   mpop   _mpop   Estimate   Standard
Error     DF     t Value   Pr > |t|
fpop rb   rb     ws       0.3790    0.6268    81      0.60    0.5471
fpop ws   rb     ws      -2.4270    0.6793    81     -3.57    0.0006
    
```

The first table of simple effects comparisons holds fixed the level of the `mpop` factor and compares the levels of the `fpop` factor. Since there is only one possible compari-

son for each male population, there are two entries in the table. The first compares the logits of mating probabilities when the male partner is a rough butt, and the second entry applies when the male partner is from the whiteside population. The second table of simple effects comparisons applies the same logic, but holds fixed the level of the female partner in the pair. Note that these four comparisons are a subset of all six possible comparisons, eliminating those where both factors are varied at the same time. The simple effect comparisons show that there is no difference in mating probabilities if the male partner is a whiteside salamander, or if the female partner is a rough butt. Rough butt females also appear to mate indiscriminately.

### Example 3. Smoothing Disease Rates; Standardized Mortality Ratios

Clayton and Kaldor (1987, Table 1) present data on observed and expected cases of lip cancer in the 56 counties of Scotland between 1975 and 1980. The expected number of cases was determined by a separate multiplicative model that accounted for the age distribution in the counties. The goal of the analysis is to estimate the county-specific log-relative risks, also known as standardized mortality ratios (SMR).

If  $Y_i$  is the number of incident cases in county  $i$  and  $E_i$  is the expected number of incident cases, then the ratio of observed to expected counts,  $Y_i/E_i$ , is the standardized mortality ratio. Clayton and Kaldor (1987) assume that there exists a relative risk  $\lambda_i$  that is specific to each county and is a random variable. Conditional on  $\lambda_i$ , the observed counts are independent Poisson variables with mean  $E_i\lambda_i$ .

An elementary mixed model for  $\lambda_i$  specifies only a random intercept for each county, in addition to a fixed intercept. Breslow and Clayton (1993), in their analysis of these data, also provide a covariate that measures the percentage of employees in agriculture, fishing, and forestry. The expanded model for the region-specific relative risk in Breslow and Clayton (1993) is

$$\lambda_i = \exp\{\beta_0 + \beta_1 x_i/10 + \gamma_i\}, \quad i = 1, \dots, 56$$

where  $\beta_0$  and  $\beta_1$  are fixed effects, and the  $\gamma_i$  are county random effects.

The following DATA step creates the data set `lipcancer`. The expected number of cases is based on the observed standardized mortality ratio for counties with lip cancer cases, and on the expected counts reported by Clayton and Kaldor (1987, Table 1) for the counties without cases. The sum of the expected counts then equals the sum of the observed counts.

```
data lipcancer;
  input county observed expected employment SMR;
  if (observed > 0) then expCount = 100*observed/SMR;
  else expCount = expected;
  datalines;
1 9 1.4 16 652.2
2 39 8.7 16 450.3
3 11 3.0 10 361.8
4 9 2.5 24 355.7
```

5	15	4.3	10	352.1
6	8	2.4	24	333.3
7	26	8.1	10	320.6
8	7	2.3	7	304.3
9	6	2.0	7	303.0
10	20	6.6	16	301.7
11	13	4.4	7	295.5
12	5	1.8	16	279.3
13	3	1.1	10	277.8
14	8	3.3	24	241.7
15	17	7.8	7	216.8
16	9	4.6	16	197.8
17	2	1.1	10	186.9
18	7	4.2	7	167.5
19	9	5.5	7	162.7
20	7	4.4	10	157.7
21	16	10.5	7	153.0
22	31	22.7	16	136.7
23	11	8.8	10	125.4
24	7	5.6	7	124.6
25	19	15.5	1	122.8
26	15	12.5	1	120.1
27	7	6.0	7	115.9
28	10	9.0	7	111.6
29	16	14.4	10	111.3
30	11	10.2	10	107.8
31	5	4.8	7	105.3
32	3	2.9	24	104.2
33	7	7.0	10	99.6
34	8	8.5	7	93.8
35	11	12.3	7	89.3
36	9	10.1	0	89.1
37	11	12.7	10	86.8
38	8	9.4	1	85.6
39	6	7.2	16	83.3
40	4	5.3	0	75.9
41	10	18.8	1	53.3
42	8	15.8	16	50.7
43	2	4.3	16	46.3
44	6	14.6	0	41.0
45	19	50.7	1	37.5
46	3	8.2	7	36.6
47	2	5.6	1	35.8
48	3	9.3	1	32.1
49	28	88.7	0	31.6
50	6	19.6	1	30.6
51	1	3.4	1	29.1
52	1	3.6	0	27.6
53	1	5.7	1	17.4
54	1	7.0	1	14.2
55	0	4.2	16	0.0
56	0	1.8	10	0.0

;

Since the mean of the Poisson variates, conditional on the random effects, is  $\mu_i = E_i \lambda_i$ , applying a log link yields

$$\log\{\mu_i\} = \log\{E_i\} + \beta_0 + \beta_1 x_i/10 + \gamma_i$$

The term  $\log\{E_i\}$  is an offset, a regressor variable whose coefficient is known to be one. Note that it is assumed that the  $E_i$  are known, they are not treated as random variables.

The following statements fit this model by residual pseudo-likelihood.

```
proc glimmix data=lipcancer;
  class county;
  x    = employment / 10;
  logn = log(expCount);
  model observed = x / dist=poisson offset=logn
                    solution ddfm=none;

  random county;
  SMR_pred = 100*exp(_zgamma_ + _xbeta_);
  id employment SMR SMR_pred;
  output out=glimmixout;
run;
```

The offset is created with the assignment statement

```
logn = log(expCount);
```

and associated with the linear predictor through the OFFSET= option of the MODEL statement. The statement

```
x = employment / 10;
```

transforms the covariate measuring percentage of employment in agriculture, fisheries, and forestry to agree with the analysis of Breslow and Clayton (1993). The DDFM=NONE option in the MODEL statement requests chi-square tests and  $z$  tests instead of the default  $F$  tests and  $t$  tests by setting the denominator degrees of freedom in tests of fixed effects to  $\infty$ .

The statement

```
SMR_pred = 100*exp(_zgamma_ + _xbeta_);
```

calculates the fitted standardized mortality rate. Note that the offset variable does not contribute to the term being exponentiated.

The OUTPUT statement saves results of the calculations to the output data set glimmixout. The ID statement specifies that only the variables listed in are written to the output data set.



**Output 3.1.** Scottish Lip Cancer Analysis

The GLIMMIX Procedure	
Model Information	
Data Set	WORK.LIPCANCER
Response Variable	observed
Response Distribution	Poisson
Link Function	Log
Variance Function	Default
Offset Variable	logn = log(expCount);
Variance Matrix	Not blocked
Estimation Technique	Residual PL
Degrees of Freedom Method	None

The GLIMMIX procedure displays in the “Model Information” table that the offset variable was computed with programming statements and the final assignment statement from your GLIMMIX statements (Output 3.1).

**Output 3.2.** Scottish Lipcancer Analysis (continued)

Class Level Information		
Class	Levels	Values
county	56	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
	Number of Observations Read	56
	Number of Observations Used	56

**Output 3.3.** Scottish Lipcancer Analysis (continued)

Dimensions	
G-side Cov. Parameters	1
Columns in X	2
Columns in Z	56
Subjects (Blocks in V)	1
Max Obs per Subject	56

There are two columns in the **X** matrix, corresponding to the intercept and the regressor  $x/10$ . There are 56 columns in the **Z** matrix, however, one for each observation in the data set (Output 3.3).

**Output 3.4.** Scottish Lipcancer Analysis (continued)

Optimization Information	
Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	1
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Profiled
Starting From	Data

The optimization involves only a single covariance parameter, the variance of the county effect ([Output 3.4](#)). Because this parameter is a variance, the GLIMMIX procedure imposes a lower boundary constraint; the solution for the variance is bounded by zero from below.

**Output 3.5.** Scottish Lipcancer Analysis (continued)

Iteration History					
Iteration	Restarts	Subiterations	Objective Function	Change	Max Gradient
0	0	4	123.64113992	0.20997891	3.848E-8
1	0	3	127.05866018	0.03393332	0.000048
2	0	2	127.48839749	0.00223427	5.753E-6
3	0	1	127.50502469	0.00006946	1.892E-7
4	0	1	127.50528068	0.00000118	2.911E-8
5	0	0	127.50528481	0.00000000	1.378E-6

Convergence criterion (PCONV=1.11022E-8) satisfied.

Following the initial creation of pseudo-data and the fit of a linear mixed model, the procedure goes through five more updates of the pseudo-data, each associated with a separate optimization ([Output 3.5](#)). Although the objective function in each optimization is the negative of twice the restricted maximum likelihood for that pseudo-data, there is no guarantee that across the outer iterations the objective function decreases in subsequent optimizations. In this example, minus twice the residual maximum likelihood at convergence takes on its smallest value at the initial optimization and increases in subsequent optimizations.

**Output 3.6.** Scottish Lipcancer Analysis (continued)

Covariance Parameter Estimates		
Cov Parm	Estimate	Standard Error
county	0.3567	0.09869

The “Covariance Parameter Estimates” table in [Output 3.6](#) shows the estimate of the variance of the region-specific log-relative risks. There is significant county-to-county heterogeneity in risks. If the covariate were removed from the analysis, as in Clayton and Kaldor (1987), the heterogeneity in county-specific risks would increase. (The fitted SMRs in Table 6 of Breslow and Clayton (1993) were obtained without the covariate  $x$  in the model.)

**Output 3.7.** Scottish Lipcancer Analysis (continued)

Solutions for Fixed Effects					
Effect	Estimate	Standard Error	DF	t Value	Pr >  t
Intercept	-0.4406	0.1572	Infty	-2.80	0.0051
x	0.6799	0.1409	Infty	4.82	<.0001

The “Solutions for Fixed Effects” table in [Output 3.7](#) displays the estimates of  $\beta_0$  and  $\beta_1$  along with their standard errors and test statistics. Because of the DDFM=NONE option in the **MODEL** statement, PROC GLIMMIX assumes that the degrees of freedom for the  $t$  tests of  $H_0: \beta_j = 0$  are infinite. The  $p$ -values correspond to probabilities under a standard normal distribution. The covariate measuring employment percentages in agriculture, fisheries, and forestry is significant. This covariate may be a surrogate for the exposure to sunlight, an important risk factor for lip cancer.

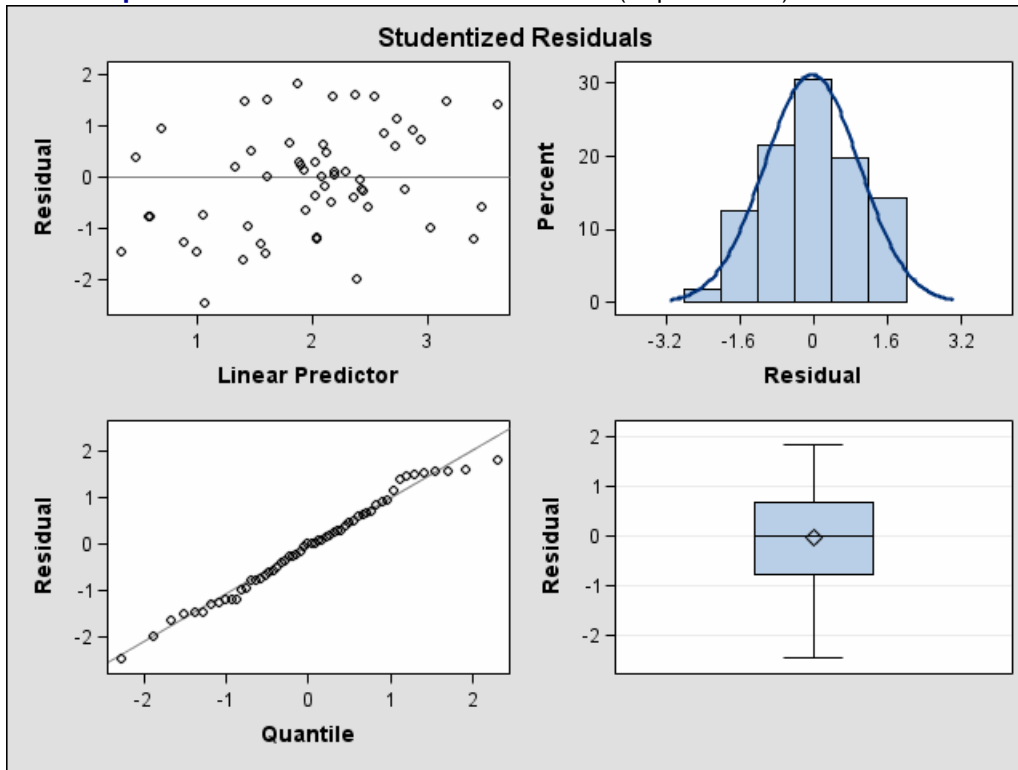
You can examine the quality of the fit of this model with various residual plots. A panel of studentized residuals is requested with the following statements.

```
ods html;
ods graphics on;
ods select StudentPanel;

proc glimmix data=lipcancer plots=studentpanel;
  class county;
  x = employment / 10;
  logn = log(expCount);
  model observed = x / dist=poisson offset=logn s ddfm=none;
  random county;
run;

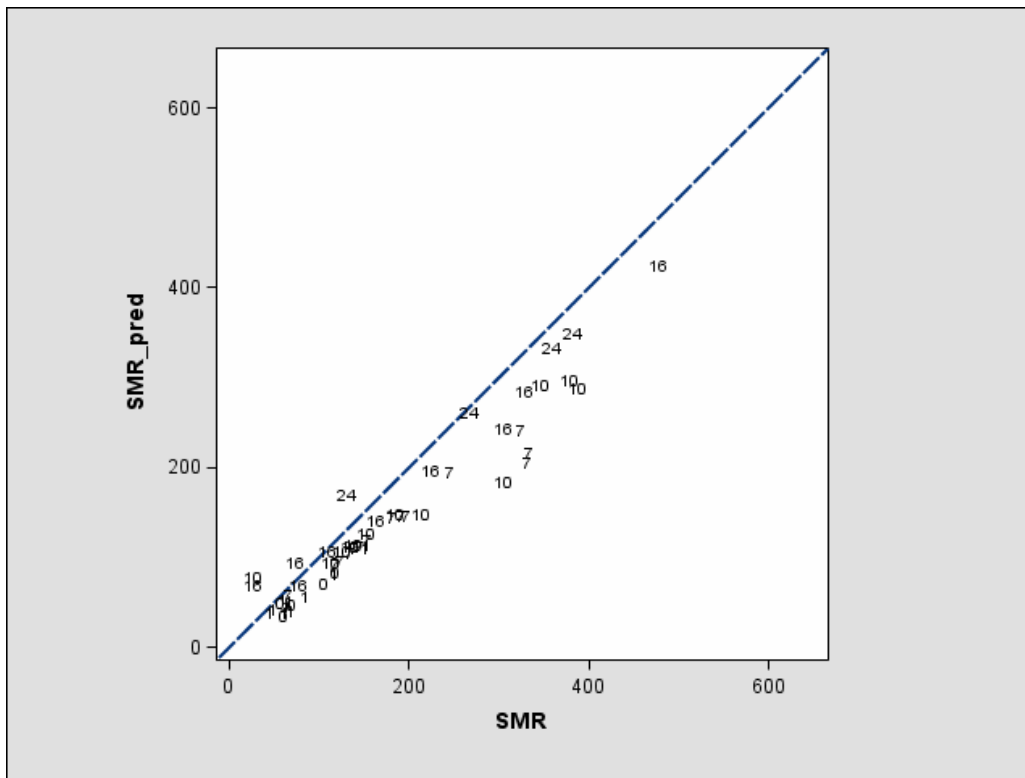
ods graphics off;
ods html close;
```

The graph in the upper left corner of the panel displays studentized residuals against the linear predictor ([Output 3.8](#)). The default of the GLIMMIX procedure is to use the estimated BLUPs in the construction of the residuals and to present them on the linear scale, which in this case is the logarithmic scale. You can change the type of the computed residual with the TYPE= suboptions of each paneled display. For example, the option **PLOTS=STUDENTPANEL(TYPE=NOBLUP)** would request a paneled display of the marginal residuals on the linear scale.

**Output 3.8.** Panel of Studentized Residuals (Experimental)

The graph in the upper right-hand corner of the panel shows a histogram with overlaid normal density. A Q-Q plot and a box plot are shown in the lower cells of the panel.

Figure 22 displays the observed and predicted standardized mortality ratios. Fitted SMRs tend to be larger than the observed SMRs for counties with small observed SMR and smaller than observed for counties with high observed SMR.



**Figure 22.** Observed and Predicted SMRs. Datalabels Indicate Covariate Values

To demonstrate the impact of the random effects adjustment to the log-relative risks, the following statements fit a Poisson regression model (a GLM) by maximum likelihood. The SMRs are county specific to the extent that risks vary according to the value of the covariate, but risks are no longer adjusted based on county-to-county heterogeneity in the observed incidence count.

```
proc glimmix data=lipcancer;
  x      = employment / 10;
  logn = log(expCount);
  model observed = x / dist=poisson offset=logn
                    solution ddfm=none;
  SMR_pred = 100*exp(_zgamma_ + _xbeta_);
  id employment SMR SMR_pred;
  output out=glimmixout;
run;
```

**Output 3.9.** Analysis as a Poisson GLM

The GLIMMIX Procedure	
Model Information	
Data Set	WORK.LIPCANCER
Response Variable	observed
Response Distribution	Poisson
Link Function	Log
Variance Function	Default
Offset Variable	logn = log(expCount);
Variance Matrix	Diagonal
Estimation Technique	Maximum Likelihood
Degrees of Freedom Method	None

Because of the absence of random effects, the GLIMMIX procedure recognizes the model as a generalized linear model and fits it by maximum likelihood ([Output 3.9](#)). The variance matrix is diagonal, because the observations are uncorrelated.

**Output 3.10.** GLM Analysis (continued)

Dimensions	
Columns in X	2
Columns in Z	0
Subjects (Blocks in V)	1
Max Obs per Subject	56

In [Output 3.10](#), the “Dimensions” table shows that there are no G-side random effects in this model, and no R-side scale parameter either.

**Output 3.11.** GLM Analysis (continued)

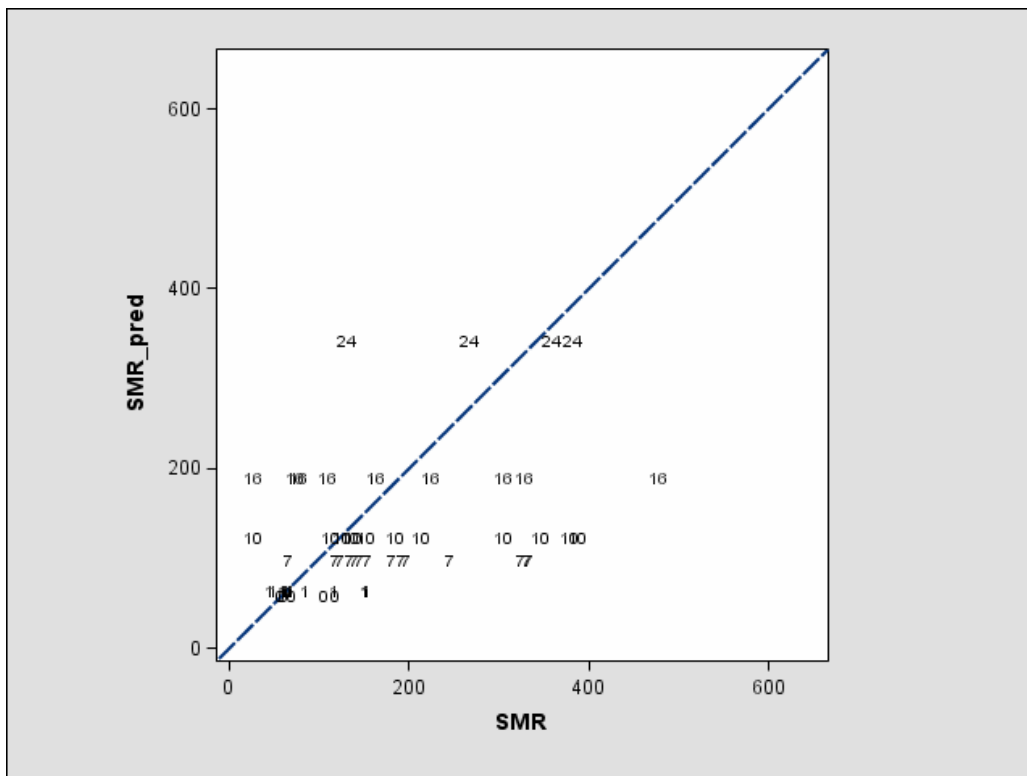
Optimization Information	
Optimization Technique	Newton-Raphson
Parameters in Optimization	2
Lower Boundaries	0
Upper Boundaries	0
Fixed Effects	Not Profiled

Because this is a GLM, the GLIMMIX procedure defaults to the Newton-Raphson algorithm and the fixed effects (intercept and slope) comprise the parameters in the optimization ([Output 3.11](#)). The default optimization technique for a GLM is Newton-Raphson.

**Output 3.12.** GLM Analysis (continued)

Parameter Estimates					
Effect	Estimate	Standard Error	DF	t Value	Pr >  t
Intercept	-0.5419	0.06951	Infty	-7.80	<.0001
x	0.7374	0.05954	Infty	12.38	<.0001

The estimates of  $\beta_0$  and  $\beta_1$  have changed from the previous analysis. In the GLMM, the estimates were  $\hat{\beta}_0 = -0.4406$  and  $\hat{\beta}_1 = 0.6799$  (Output 3.12). More importantly, without the county-specific adjustments through the best linear unbiased predictors of the random effects, the predicted SMRs are the same for all counties with the same percentage of employees in agriculture, fisheries, and forestry (Figure 23).



**Figure 23.** Observed and Predicted SMRs in Poisson GLIM

### Example 4. Quasi-Likelihood Estimation for Proportions with Unknown Distribution

Wedderburn (1974) analyzes data on the incidence of leaf blotch (*Rhynchosporium secalis*) on barley. The data represent the percentage of leaf area affected in a two-way layout with 10 barley varieties at 9 sites. The following DATA step converts these data to proportions, as were analyzed in McCullagh and Nelder (1989, Ch. 9.2.4). The

purpose of the analysis is to make comparisons among the varieties, adjusted for site effects.

```

data blotch;
  array p{9} pct1-pct9;
  input variety pct1-pct9;
  do site = 1 to 9;
    prop = p{site}/100;
    output;
  end;
  drop pct1-pct9;
  datalines;
1 0.05 0.00 1.25 2.50 5.50 1.00 5.00 5.00 17.50
2 0.00 0.05 1.25 0.50 1.00 5.00 0.10 10.00 25.00
3 0.00 0.05 2.50 0.01 6.00 5.00 5.00 5.00 42.50
4 0.10 0.30 16.60 3.00 1.10 5.00 5.00 5.00 50.00
5 0.25 0.75 2.50 2.50 2.50 5.00 50.00 25.00 37.50
6 0.05 0.30 2.50 0.01 8.00 5.00 10.00 75.00 95.00
7 0.50 3.00 0.00 25.00 16.50 10.00 50.00 50.00 62.50
8 1.30 7.50 20.00 55.00 29.50 5.00 25.00 75.00 95.00
9 1.50 1.00 37.50 5.00 20.00 50.00 50.00 75.00 95.00
10 1.50 12.70 26.25 40.00 43.50 75.00 75.00 75.00 95.00
;

```

Little is known about the distribution of the leaf area proportions. The outcomes are not binomial proportions, since they do not represent the ratio of a count over a total number of Bernoulli trials. However, since the mean proportion  $\mu_{ij}$  for variety  $j$  on site  $i$  must lie in the interval  $[0, 1]$ , you can commence the analysis with a model that treats Prop as a “pseudo-binomial” variable:

$$\begin{aligned}
 E[Prop_{ij}] &= \mu_{ij} \\
 \mu_{ij} &= 1/(1 + \exp\{-\eta_{ij}\}) \\
 \eta_{ij} &= \beta_0 + \alpha_i + \tau_j \\
 \text{var}[Prop_{ij}] &= \phi\mu_{ij}(1 - \mu_{ij})
 \end{aligned}$$

Here,  $\eta_{ij}$  is the linear predictor for variety  $j$  on site  $i$ ,  $\alpha_i$  denotes the  $i$ th site effect and  $\tau_j$  denotes the  $j$ th barley variety effect. The logit of the expected leaf area proportions is linearly related to these effects. The variance function of the model is that of a binomial( $n, \mu_{ij}$ ) variable, and  $\phi$  is an overdispersion parameter. This model is fit with the GLIMMIX procedure with the following statements.

```

proc glimmix data=blotch;
  class site variety;
  model prop = site variety / link=logit dist=binomial;
  random _residual_;
  lsmeans variety / diff=control('1');
run;

```



The **MODEL** statement specifies the distribution as binomial and the logit link. Since the variance function of the binomial distribution is  $a(\mu) = \mu(1 - \mu)$ , you use the

```
random _residual_;
```

statement to specify the scale parameter  $\phi$ . The **LSMEANS** statement requests estimates of the least-squares means for the barley variety. The **DIFF=CONTROL('1')** option requests tests of least-squares means differences against the first variety.

The “Model Information” table in [Output 4.1](#) describes the model and methods used in fitting the statistical model. It is assumed here that the data are binomial proportions.

**Output 4.1.** Pseudo-Binomial Analysis

The GLIMMIX Procedure	
Model Information	
Data Set	WORK.BLOTCH
Response Variable	prop
Response Distribution	Binomial
Link Function	Logit
Variance Function	Default
Variance Matrix	Diagonal
Estimation Technique	Maximum Likelihood
Degrees of Freedom Method	Residual

The “Class Level Information” table in [Output 4.2](#) lists the number of levels of the **Site** and **Variety** effects and their values. All 90 observations read from the data are used in the analysis.

**Output 4.2.** Pseudo-Binomial Analysis (continued)

Class Level Information		
Class	Levels	Values
site	9	1 2 3 4 5 6 7 8 9
variety	10	1 2 3 4 5 6 7 8 9 10
Number of Observations Read		90
Number of Observations Used		90

In [Output 4.3](#), the “Dimensions” table shows that the model does not contain G-side random effects. There is a single covariance parameter, which corresponds to  $\phi$ . The “Optimization Information” table in [Output 4.4](#) shows that the optimization comprises 18 parameters. These correspond to the 18 nonsingular columns of the  $X'X$  matrix.

**Output 4.3.** Pseudo-Binomial Analysis (continued)

Dimensions	
Covariance Parameters	1
Columns in X	20
Columns in Z	0
Subjects (Blocks in V)	1
Max Obs per Subject	90

**Output 4.4.** Pseudo-Binomial Analysis (continued)

Optimization Information	
Optimization Technique	Newton-Raphson
Parameters in Optimization	18
Lower Boundaries	0
Upper Boundaries	0
Fixed Effects	Not Profiled

**Output 4.5.** Pseudo-Binomial Analysis (continued)

Fit Statistics	
-2 Log Likelihood	57.15
AIC (smaller is better)	93.15
AICC (smaller is better)	102.79
BIC (smaller is better)	138.15
CAIC (smaller is better)	156.15
HQIC (smaller is better)	111.30
Pearson Chi-Square	6.39
Pearson Chi-Square / DF	0.09

There are significant site and variety effects in this model based on the approximate Type III  $F$  tests (Output 4.6).

**Output 4.6.** Pseudo-Binomial Analysis (continued)

Type III Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
site	8	72	18.25	<.0001
variety	9	72	13.85	<.0001

Output 4.7 displays the least-squares means for this analysis. These are obtained by averaging

$$\text{logit}(\hat{\mu}_{ij}) = \hat{\eta}_{ij}$$

across the sites. In other words, LS-means are computed on the linked scale where the model effects are additive. Note that the least-squares means are ordered by variety. The estimate of the expected proportion of infected leaf area for the first variety is

$$\hat{\mu}_{.,1} = \frac{1}{1 + \exp\{4.38\}} = 0.0124$$

and that for the last variety is

$$\hat{\mu}_{.,10} = \frac{1}{1 + \exp\{0.127\}} = 0.468$$

**Output 4.7.** Pseudo-Binomial Analysis (continued)

variety Least Squares Means					
variety	Estimate	Standard Error	DF	t Value	Pr >  t
1	-4.3800	0.5643	72	-7.76	<.0001
2	-4.2300	0.5383	72	-7.86	<.0001
3	-3.6906	0.4623	72	-7.98	<.0001
4	-3.3319	0.4239	72	-7.86	<.0001
5	-2.7653	0.3768	72	-7.34	<.0001
6	-2.0089	0.3320	72	-6.05	<.0001
7	-1.8095	0.3228	72	-5.61	<.0001
8	-1.0380	0.2960	72	-3.51	0.0008
9	-0.8800	0.2921	72	-3.01	0.0036
10	-0.1270	0.2808	72	-0.45	0.6523

Because of the ordering of the least-squares means, the differences against the first variety are also ordered from smallest to largest (Output 4.8).

**Output 4.8.** Pseudo-Binomial Analysis (continued)

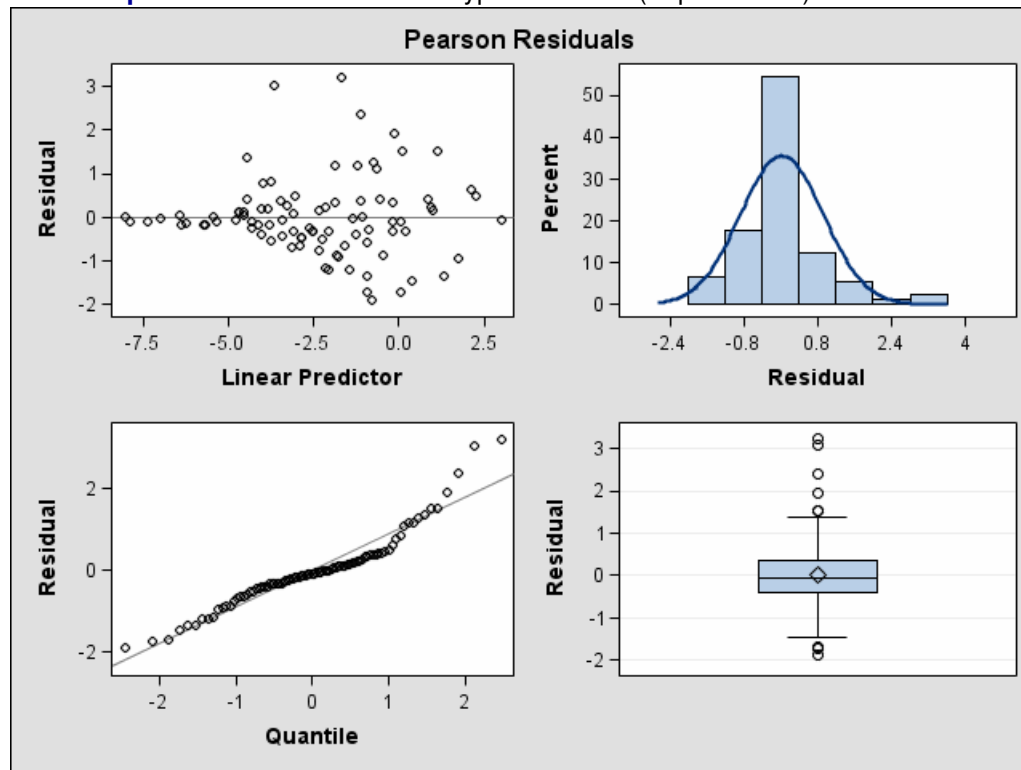
Differences of variety Least Squares Means						
variety	_variety	Estimate	Standard Error	DF	t Value	Pr >  t
2	1	0.1501	0.7237	72	0.21	0.8363
3	1	0.6895	0.6724	72	1.03	0.3086
4	1	1.0482	0.6494	72	1.61	0.1109
5	1	1.6147	0.6257	72	2.58	0.0119
6	1	2.3712	0.6090	72	3.89	0.0002
7	1	2.5705	0.6065	72	4.24	<.0001
8	1	3.3420	0.6015	72	5.56	<.0001
9	1	3.5000	0.6013	72	5.82	<.0001
10	1	4.2530	0.6042	72	7.04	<.0001

This analysis depends on your choice for the variance function which was implied by the binomial distribution. You can diagnose the distributional assumption by examining various graphical diagnostics measures. The following statements request a panel display of the Pearson-type residuals.

```
ods html;
ods graphics on;
ods select PearsonPanel;

proc glimmix data=blotch plots=pearsonpanel;
  class site variety;
  model prop = site variety / link=logit dist=binomial;
  random _residual_;
run;

ods graphics off;
ods html close;
```

**Output 4.9.** Panel of Pearson-type Residuals (Experimental)

Output 4.9 clearly indicates that the chosen variance function is not appropriate for these data. As  $\mu$  approaches zero or one, the variability in the residuals is less than that implied by the binomial variance function. To remedy this situation, McCullagh and Nelder (1989) consider instead the variance function

$$\text{var}[Prop_{ij}] = \mu_{ij}^2(1 - \mu_{ij})^2$$

Imagine two varieties with  $\mu_i = 0.1$  and  $\mu_k = 0.5$ . Under the binomial variance function, the variance of the proportion for variety  $k$  is 2.77 times larger than that for variety  $i$ . Under the revised model this ratio increases to  $2.77^2 = 7.67$ .

The analysis of the revised model is obtained with the next set of GLIMMIX statements. Since you need to model a variance function that does not correspond to any of the built-in distributions, you need to supply a function with an assignment to the automatic variable `_VARIANCE_`. The GLIMMIX procedure then considers the distribution of the data as unknown. The corresponding estimation technique is quasi-likelihood. Since this model does not include an extra scale parameter, you can drop the `RANDOM _RESIDUAL_` statement from the analysis.

```
ods html;
ods graphics on;
ods select ModelInfo FitStatistics LSMeans Diffs PearsonPanel;

proc glimmix data=blotch plots=pearsonpanel;
  class site variety;
```

```

    _variance_ = _mu_**2 * (1-_mu_)**2;
    model prop = site variety / link=logit;
    lsmeans variety / diff=control('1');
run;

ods graphics off;
ods html close;

```

The “Model Information” table in [Output 4.10](#) now displays the distribution as “Unknown”, because of the assignment made in the GLIMMIX statements to `_VARIANCE_`. The table also shows the expression evaluated as the variance function.

#### Output 4.10. Quasi-Likelihood Analysis

The GLIMMIX Procedure	
Model Information	
Data Set	WORK.BLOTCH
Response Variable	prop
Response Distribution	Unknown
Link Function	Logit
Variance Function	<code>_mu_**2 * (1-_mu_)**2</code>
Variance Matrix	Diagonal
Estimation Technique	Quasi-Likelihood
Degrees of Freedom Method	Residual

The fit statistics of the model are now expressed in terms of the log quasi-likelihood. It is computed as

$$\sum_{i=1}^9 \sum_{j=1}^{10} \int_{y_{ij}}^{\mu_{ij}} \frac{y_{ij} - t}{t^2(1-t)^2} dt$$

Twice the negative of this sum equals  $-85.74$ , which is displayed in the “Fit Statistics” table ([Output 4.11](#)).

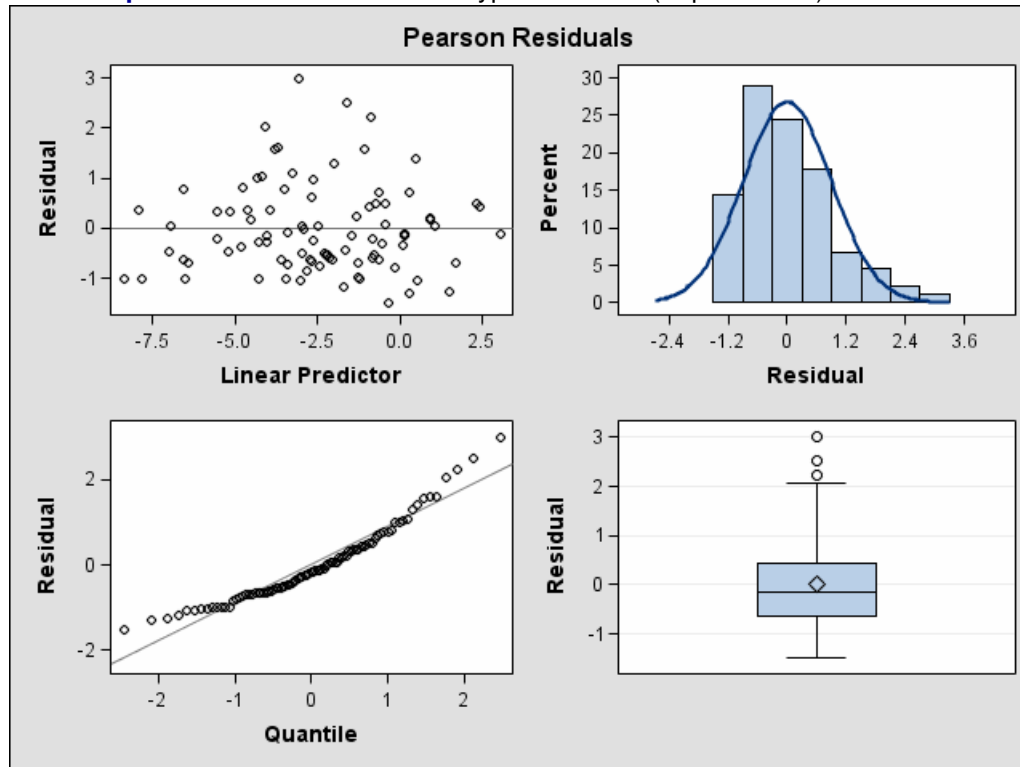
The scaled Pearson statistic is now 0.99. Inclusion of an extra scale parameter  $\phi$  would have little or no effect on the results.

#### Output 4.11. Quasi-Likelihood Analysis (continued)

Fit Statistics	
-2 Log Quasi-Likelihood	-85.74
Quasi-AIC (smaller is better)	-49.74
Quasi-AICC (smaller is better)	-40.11
Quasi-BIC (smaller is better)	-4.75
Quasi-CAIC (smaller is better)	13.25
Quasi-HQIC (smaller is better)	-31.60
Pearson Chi-Square	71.17
Pearson Chi-Square / DF	0.99

The panel of Pearson-type residuals now shows a much more adequate distribution for the residuals and a reduction in the number of outlying residuals (Output 4.12).

**Output 4.12.** Panel of Pearson-type Residuals (Experimental)



The least-squares means are no longer ordered in size by variety (Output 4.13). For example,  $\text{logit}(\hat{\mu}_{.1}) > \text{logit}(\hat{\mu}_{.2})$ . Under the revised model, the second variety has a greater percentage of its leaf area covered by blotch, compared to the first variety. Varieties 5 and 6 and varieties 8 and 9 show similar reversal in ranking.

**Output 4.13.** Quasi-Likelihood Analysis (continued)

variety Least Squares Means					
variety	Estimate	Standard Error	DF	t Value	Pr >  t
1	-4.0453	0.3333	72	-12.14	<.0001
2	-4.5126	0.3333	72	-13.54	<.0001
3	-3.9664	0.3333	72	-11.90	<.0001
4	-3.0912	0.3333	72	-9.27	<.0001
5	-2.6927	0.3333	72	-8.08	<.0001
6	-2.7167	0.3333	72	-8.15	<.0001
7	-1.7052	0.3333	72	-5.12	<.0001
8	-0.7827	0.3333	72	-2.35	0.0216
9	-0.9098	0.3333	72	-2.73	0.0080
10	-0.1580	0.3333	72	-0.47	0.6369

Interestingly, the standard errors are constant among the LS-means (Output 4.13), and among the LS-means differences (Output 4.14). This is due to the fact that for the logit link

$$\frac{\partial \mu}{\partial \eta} = \mu(1 - \mu)$$

which cancels with the square root of the variance function in the estimating equations. The analysis is thus orthogonal.

**Output 4.14.** Quasi-Likelihood Analysis (continued)

Differences of variety Least Squares Means						
variety	_variety	Estimate	Standard Error	DF	t Value	Pr >  t
2	1	-0.4673	0.4714	72	-0.99	0.3249
3	1	0.07885	0.4714	72	0.17	0.8676
4	1	0.9541	0.4714	72	2.02	0.0467
5	1	1.3526	0.4714	72	2.87	0.0054
6	1	1.3286	0.4714	72	2.82	0.0062
7	1	2.3401	0.4714	72	4.96	<.0001
8	1	3.2626	0.4714	72	6.92	<.0001
9	1	3.1355	0.4714	72	6.65	<.0001
10	1	3.8873	0.4714	72	8.25	<.0001

## Example 5. Joint Modeling of Binary and Count Data

Clustered data arise when multiple observations are collected on the same sampling or experimental unit. Often, these multiple observations refer to the same attribute measured at different points in time or space. This leads to repeated measures, longitudinal, and spatial data, which are special forms of multivariate data. A different class of multivariate data arises when the multiple observations refer to different attributes.

The data set `hernio`, created in the DATA step below, provides an example of a bivariate outcome variable. It reflects the condition and length of hospital stay for 32 herniorrhaphy patients. These data are based on data given by Mosteller and Tukey (1977) and reproduced in Hand et al. (1994, p. 390, 391). The data set below does not contain all the covariates given in these sources. The response variables are `leave` and `los`; these denote the condition of the patient upon leaving the operating room and the length of hospital stay after the operation (in days). The variable `leave` takes on the value one if a patient experiences a routine recovery, and the value zero if post-operative intensive care was required. The binary variable `OKstatus` distinguishes patients based on their post-operative physical status (“1” implies better status).

```
data hernio;
  input patient age gender$ OKstatus leave los;
  datalines;
```



```

1    78  m   1   0   9
2    60  m   1   0   4
3    68  m   1   1   7
4    62  m   0   1  35
5    76  m   0   0   9
6    76  m   1   1   7
7    64  m   1   1   5
8    74  f   1   1  16
9    68  m   0   1   7
10   79  f   1   0  11
11   80  f   0   1   4
12   48  m   1   1   9
13   35  f   1   1   2
14   58  m   1   1   4
15   40  m   1   1   3
16   19  m   1   1   4
17   79  m   0   0   3
18   51  m   1   1   5
19   57  m   1   1   8
20   51  m   0   1   8
21   48  m   1   1   3
22   48  m   1   1   5
23   66  m   1   1   8
24   71  m   1   0   2
25   75  f   0   0   7
26    2  f   1   1   0
27   65  f   1   0  16
28   42  f   1   0   3
29   54  m   1   0   2
30   43  m   1   1   3
31    4  m   1   1   3
32   52  m   1   1   8
;

```

While the response variable `los` is a Poisson count variable, the response variable `leave` is a binary variable. You can perform separate analysis for the two outcomes, for example, by fitting a logistic model for the operating room exit condition and a Poisson regression model for the length of hospital stay. This, however, would ignore the correlation between the two outcomes. Intuitively, you would expect that the length of post-operative hospital stay is longer for those patients who had more tenuous exit conditions.

The following DATA step converts the data set `hernio` from the multivariate form to the univariate form. In the multivariate form the responses are stored in separate variables. The GLIMMIX procedure requires the univariate data structure.

```

data hernio_uv;
  length dist $7;
  set hernio;
  response = (leave=1);
  dist     = "Binary";
  output;

```

```

response = los;
dist      = "Poisson";
output;
keep patient age OKstatus response dist;
run;

```

This DATA step expands the 32 observations in data set `hernio` into 64 observations, stacking two observations per patient. The character variable `dist` identifies the distribution that is assumed for the respective observations within a patient. The first observation for each patient corresponds to the binary response.

The following GLIMMIX statements fit a logistic regression model with two regressors (`age` and `OKstatus`) to the binary observations.

```

proc glimmix data=hernio_uv(where=(dist="Binary"));
  model response(event='1') = age OKstatus / s dist=binary;
run;

```

The `EVENT=('1')` response option requests that PROC GLIMMIX model the probability  $\Pr(\text{leave} = 1)$ , that is, the probability of routine recovery. The fit statistics and parameter estimates for this univariate analysis are shown in [Output 5.1](#). The coefficient for the `age` effect is negative ( $-0.07725$ ) and marginally significant at the 5% level ( $p = 0.0491$ ). The negative sign indicates that the probability of routine recovery decreases with age. The coefficient for the `OKstatus` variable is also negative. Its large standard error and the  $p$ -value of 0.7341 indicate, however, that this regressor is not significant.

### Output 5.1. Univariate Logistic Regression

The GLIMMIX Procedure					
Fit Statistics					
	-2 Log Likelihood				32.77
	AIC (smaller is better)				38.77
	AICC (smaller is better)				39.63
	BIC (smaller is better)				43.17
	CAIC (smaller is better)				46.17
	HQIC (smaller is better)				40.23
	Pearson Chi-Square				30.37
	Pearson Chi-Square / DF				1.05
Parameter Estimates					
Effect	Estimate	Standard Error	DF	t Value	Pr >  t
Intercept	5.7694	2.8245	29	2.04	0.0503
age	-0.07725	0.03761	29	-2.05	0.0491
OKstatus	-0.3516	1.0253	29	-0.34	0.7341

Based on the univariate logistic regression analysis, you would probably want to revisit the model, examine other regressor variables, test for gender effects and interactions, and so forth. For this example, we are content with the two-regressor model. It will be illustrative to trace the relative importance of the two regressors through various types of models.

The next statements fit the same regressors to the count data.

```
proc glimmix data=hernio_uv(where=(dist="Poisson"));
  model response = age OKStatus / s dist=Poisson;
run;
```

### Output 5.2. Univariate Poisson Regression

The GLIMMIX Procedure					
Fit Statistics					
-2 Log Likelihood					215.52
AIC (smaller is better)					221.52
AICC (smaller is better)					222.38
BIC (smaller is better)					225.92
CAIC (smaller is better)					228.92
HQIC (smaller is better)					222.98
Pearson Chi-Square					129.98
Pearson Chi-Square / DF					4.48
Parameter Estimates					
Effect	Estimate	Standard Error	DF	t Value	Pr >  t
Intercept	1.2640	0.3393	29	3.72	0.0008
age	0.01525	0.004454	29	3.42	0.0019
OKstatus	-0.3301	0.1562	29	-2.11	0.0433

For this response, both regressors appear to make significant contributions at the 5% significance level (Output 5.2). The sign of the coefficient seems appropriate, the length of hospital stay should increase with patient age and be shorter for patients with better pre-operative health. The magnitude of the scaled Pearson statistic (4.48) indicates, however, that there is considerable overdispersion in this model. This could be due to omitted variables or an improper distributional assumption. The importance of pre-operative health status, for example, can change with a patient's age, which could call for an interaction term.

You can also model both responses jointly. The following statements request a multivariate analysis.

```
proc glimmix data=hernio_uv;
  class dist;
  model response(event='1') = dist dist*age dist*OKstatus /
    noint s dist=byobs(dist);
run;
```

The `DIST=BYOBS` option in the `MODEL` statement instructs the GLIMMIX procedure to examine the variable `dist` in order to identify the distribution of an observation. The variable can be character or numeric. See the `DIST=` option of the `MODEL` statement for a list of the numeric codes for the various distributions that are compatible with the `DIST=BYOBS` formulation. Since no `LINK=` option is specified, the link functions are chosen as the default links that correspond to the respective distributions. In this case, the LOGIT link is applied to the binary observations and the LOG link is applied to the Poisson outcomes. The `dist` variable is also listed in the `CLASS` statement, which enables you to use interaction terms in the `MODEL` statement to vary the regression coefficients by response distribution. The `NOINT` option is used here so that the parameter estimates of the joint model are directly comparable to those in [Output 5.1](#) and [Output 5.2](#).

The “Fit Statistics” and “Parameter Estimates” tables of this bivariate estimation process are shown in [Output 5.3](#).

### Output 5.3. Bivariate Analysis – Independence

The GLIMMIX Procedure						
Fit Statistics						
Description	Binary	Poisson	Total			
-2 Log Likelihood	32.77	215.52	248.29			
AIC (smaller is better)	44.77	227.52	260.29			
AICC (smaller is better)	48.13	230.88	261.77			
BIC (smaller is better)	53.56	236.32	273.25			
CAIC (smaller is better)	59.56	242.32	279.25			
HQIC (smaller is better)	47.68	230.44	265.40			
Pearson Chi-Square	30.37	129.98	160.35			
Pearson Chi-Square / DF	1.05	4.48	2.76			
Parameter Estimates						
Effect	dist	Estimate	Standard Error	DF	t Value	Pr >  t
dist	Binary	5.7694	2.8245	58	2.04	0.0456
dist	Poisson	1.2640	0.3393	58	3.72	0.0004
age*dist	Binary	-0.07725	0.03761	58	-2.05	0.0445
age*dist	Poisson	0.01525	0.004454	58	3.42	0.0011
OKstatus*dist	Binary	-0.3516	1.0253	58	-0.34	0.7329
OKstatus*dist	Poisson	-0.3301	0.1562	58	-2.11	0.0389

The “Fit Statistics” table now contains a separate column for each response distribution, as well as an overall contribution. Since the model does not specify any random effects or R-side correlations, the log likelihoods are additive. The parameter estimates and their standard errors in this joint model are identical to those in [Output 5.1](#) and [Output 5.2](#). The  $p$ -values reflect the larger “sample size” in the joint analysis. Note that the coefficients would be different from the separate analyses, if the `dist` variable had not been used to form interactions with the model effects.

There are two ways in which the correlations between the two responses for the same patient can be incorporated. You can induce them through shared random effects or model the dependency directly. The following statements fit a model that induces correlation: You add the patient variable to the CLASS statement and add a RANDOM statement.

```
proc glimmix data=hernio_uv;
  class patient dist;
  model response(event='1') = dist dist*age dist*OKstatus /
    noint s dist=byobs(dist);
  random int / subject=patient;
run;
```

**Output 5.4.** Bivariate Analysis – Mixed Model

The GLIMMIX Procedure						
Fit Statistics						
-2 Res Log Pseudo-Likelihood						226.71
Generalized Chi-Square						52.25
Gener. Chi-Square / DF						0.90
Covariance Parameter Estimates						
Cov Parm	Subject	Estimate	Standard Error			
Intercept	patient	0.2990	0.1116			
Solutions for Fixed Effects						
Effect	dist	Estimate	Standard Error	DF	t Value	Pr >  t
dist	Binary	5.7783	2.9048	29	1.99	0.0562
dist	Poisson	0.8410	0.5696	29	1.48	0.1506
age*dist	Binary	-0.07572	0.03791	29	-2.00	0.0552
age*dist	Poisson	0.01875	0.007383	29	2.54	0.0167
OKstatus*dist	Binary	-0.4697	1.1251	29	-0.42	0.6794
OKstatus*dist	Poisson	-0.1856	0.3020	29	-0.61	0.5435

The “Fit Statistics” table in [Output 5.4](#) no longer has separate columns for each response distribution, since the data are not independent. The log (pseudo-)likelihood does not factor into additive component that correspond to distributions. Instead, it factors into components associated with subjects.

The estimate of the variance of the random patient intercept is 0.2990, and the estimated standard error of this variance component estimate is 0.1116. There appears to be significant patient-to-patient variation in the intercepts. The estimates of the fixed effects as well as their estimated standard errors have changed from the bivariate-independence analysis (see [Output 5.3](#)). When the length of hospital stay and the post-operative condition are modeled jointly, the pre-operative health status (variable

OKStatus) no longer appears significant. Compare this result to [Output 5.3](#); in the separate analyses the initial health status was a significant predictor of the length of hospital stay. A further joint analysis of these data would probably remove this predictor from the model entirely.

A joint model of the second kind, where correlations are modeled directly, is fit with the following GLIMMIX statements.

```
proc glimmix data=hernio_uv;
  class patient dist;
  model response(event='1') = dist dist*age dist*OKstatus /
    noint s dist=byobs(dist);
  random _residual_ / subject=patient type=chol;
run;
```

Instead of a shared G-side random effect, an R-side covariance structure is used to model the correlations. It is important to note that this is a marginal model that models covariation on the scale of the data. The previous model involves the  $Z\gamma$  random components inside the linear predictor.

The `_RESIDUAL_` keyword instructs PROC GLIMMIX to model the R-side correlations. Because of the `SUBJECT=PATIENT` option, data from different patients are independent, and data from a single patient follow the covariance model specified with the `TYPE=` option. In this case, a generally unstructured  $2 \times 2$  covariance matrix is modeled, but in its Cholesky parameterization. This ensures that the resulting covariance matrix is at least positive semidefinite and stabilizes the numerical optimizations.

**Output 5.5.** Bivariate Analysis – Marginal Correlated Error Model

The GLIMMIX Procedure						
Fit Statistics						
-2 Res Log Pseudo-Likelihood		240.98				
Generalized Chi-Square		58.00				
Gener. Chi-Square / DF		1.00				
Covariance Parameter Estimates						
Cov Parm	Subject	Estimate	Standard Error			
CHOL(1,1)	patient	1.0162	0.1334			
CHOL(2,1)	patient	0.3942	0.3893			
CHOL(2,2)	patient	2.0819	0.2734			
Solutions for Fixed Effects						
Effect	dist	Estimate	Standard Error	DF	t Value	Pr >  t
dist	Binary	5.6514	2.8283	26	2.00	0.0563
dist	Poisson	1.2463	0.7189	26	1.73	0.0948
age*dist	Binary	-0.07568	0.03765	26	-2.01	0.0549
age*dist	Poisson	0.01548	0.009432	26	1.64	0.1128
OKstatus*dist	Binary	-0.3421	1.0384	26	-0.33	0.7445
OKstatus*dist	Poisson	-0.3253	0.3310	26	-0.98	0.3349

The “Covariance Parameter Estimates” table in [Output 5.5](#) contains three entries for this model, corresponding to a  $(2 \times 2)$  covariance matrix for each patient. The Cholesky root of the  $\mathbf{R}$  matrix is

$$\mathbf{L} = \begin{bmatrix} 1.0162 & 0 \\ 0.3942 & 2.0819 \end{bmatrix}$$

so that the covariance matrix can be obtained as

$$\mathbf{LL}' = \begin{bmatrix} 1.0162 & 0 \\ 0.3942 & 2.0819 \end{bmatrix} \begin{bmatrix} 1.0162 & 0.3942 \\ 0 & 2.0819 \end{bmatrix} = \begin{bmatrix} 1.0326 & 0.4005 \\ 0.4005 & 4.4897 \end{bmatrix}$$

This is not the covariance matrix of the data, however, since the variance functions need to be accounted for.

The  $p$ -values in the “Solutions for Fixed Effects” table indicate the same pattern of significance and nonsignificance as in the conditional model with random patient intercepts.

## Example 6. Radial Smoothing of Repeated Measures Data

This example of a repeated measures study is taken from Diggle, Liang, and Zeger (1994, p. 100). The data consist of body weights of 27 cows, measured at 23 unequally spaced time points over a period of approximately 22 months. Following Diggle, Liang, and Zeger (1994), one animal is removed from the analysis, one observation is removed according to their Figure 5.7, and the time is shifted to start at 0 and is measured in 10-day increments. The design is a  $2 \times 2$  factorial, and the factors are the infection of an animal with *M. paratuberculosis* and whether the animal is receiving iron dosing.

The following DATA steps create the data and arrange it in univariate format.

```

data times;
  input time1-time23;
  datalines;
122 150 166 179 219 247 276 296 324 354 380 445
478 508 536 569 599 627 655 668 723 751 781
;

data cows;
  if _n_ = 1 then merge times;
  array t{23} time1 - time23;
  array w{23} weight1 - weight23;
  input cow iron infection weight1-weight23 @@;
  do i=1 to 23;
    weight = w{i};
    tpoint = (t{i}-t{1})/10;
    output;
  end;
  keep cow iron infection tpoint weight;
  datalines;
1 0 0 4.7 4.905 5.011 5.075 5.136 5.165 5.298 5.323
5.416 5.438 5.541 5.652 5.687 5.737 5.814 5.799
5.784 5.844 5.886 5.914 5.979 5.927 5.94
2 0 0 4.868 5.075 5.193 5.22 5.298 5.416 5.481 5.521
5.617 5.635 5.687 5.768 5.799 5.872 5.886 5.872
5.914 5.966 5.991 6.016 6.087 6.098 6.153
3 0 0 4.868 5.011 5.136 5.193 5.273 5.323 5.416 5.46
5.521 5.58 5.617 5.687 5.72 5.753 5.784 5.784
5.784 5.814 5.829 5.872 5.927 5.9 5.991
4 0 0 4.828 5.011 5.136 5.193 5.273 5.347 5.438 5.561
5.541 5.598 5.67 . 5.737 5.844 5.858 5.872
5.886 5.927 5.94 5.979 6.052 6.028 6.12
5 1 0 4.787 4.977 5.043 5.136 5.106 5.298 5.298 5.371
5.438 5.501 5.561 5.652 5.67 5.737 5.784 5.768
5.784 5.784 5.829 5.858 5.914 5.9 5.94
6 1 0 4.745 4.868 5.043 5.106 5.22 5.298 5.347 5.347
5.416 5.501 5.561 5.58 5.687 5.72 5.737 5.72
5.737 5.753 5.768 5.784 5.844 5.844 5.9
7 1 0 4.745 4.905 5.011 5.106 5.165 5.273 5.371 5.416
5.416 5.521 5.541 5.635 5.687 5.704 5.784 5.768
5.768 5.814 5.829 5.858 5.94 5.94 6.004
8 0 1 4.942 5.106 5.136 5.193 5.298 5.347 5.46 5.521
5.561 5.58 5.635 5.704 5.784 5.823 5.858 5.9
5.94 5.991 6.016 6.064 6.052 6.016 5.979
9 0 1 4.605 4.745 4.868 4.905 4.977 5.22 5.165 5.22

```



			5.22	5.247	5.298	5.416	5.501	5.521	5.58	5.58
			5.635	5.67	5.72	5.753	5.799	5.829	5.858	
10	0	1	4.7	4.868	4.905	4.977	5.011	5.106	5.165	5.22
			5.22	5.22	5.273	5.384	5.438	5.438	5.501	5.501
			5.541	5.598	5.58	5.635	5.687	5.72	5.704	
11	0	1	4.828	5.011	5.075	5.165	5.247	5.323	5.394	5.46
			5.46	5.501	5.541	5.609	5.687	5.704	5.72	5.704
			5.704	5.72	5.737	5.768	5.858	5.9	5.94	
12	0	1	4.7	4.828	4.905	5.011	5.075	5.165	5.247	5.298
			5.298	5.323	5.416	5.505	5.561	5.58	5.561	5.635
			5.687	5.72	5.72	5.737	5.784	5.814	5.799	
13	0	1	4.828	5.011	5.075	5.136	5.22	5.273	5.347	5.416
			5.438	5.416	5.521	5.628	5.67	5.687	5.72	5.72
			5.799	5.858	5.872	5.914	5.94	5.991	6.016	
14	0	1	4.828	4.942	5.011	5.075	5.075	5.22	5.273	5.298
			5.323	5.298	5.394	5.489	5.541	5.58	5.617	5.67
			5.704	5.753	5.768	5.814	5.872	5.927	5.927	
15	0	1	4.745	4.905	4.977	5.075	5.193	5.22	5.298	5.323
			5.394	5.394	5.438	5.583	5.617	5.652	5.687	5.72
			5.753	5.768	5.814	5.844	5.886	5.886	5.886	
16	0	1	4.7	4.868	5.011	5.043	5.106	5.165	5.247	5.298
			5.347	5.371	5.438	5.455	5.617	5.635	5.704	5.737
			5.784	5.768	5.814	5.844	5.886	5.94	5.927	
17	1	1	4.605	4.787	4.828	4.942	5.011	5.136	5.22	5.247
			5.273	5.247	5.347	5.366	5.416	5.46	5.541	5.481
			5.501	5.635	5.652	5.598	5.635	5.635	5.598	
18	1	1	4.828	4.977	5.011	5.136	5.273	5.298	5.371	5.46
			5.416	5.416	5.438	5.557	5.617	5.67	5.72	5.72
			5.799	5.858	5.886	5.914	5.979	6.004	6.028	
19	1	1	4.7	4.905	4.942	5.011	5.043	5.136	5.193	5.193
			5.247	5.22	5.323	5.338	5.371	5.394	5.438	5.416
			5.501	5.561	5.541	5.58	5.652	5.67	5.704	
20	1	1	4.745	4.905	4.977	5.043	5.136	5.273	5.347	5.394
			5.416	5.394	5.521	5.617	5.617	5.617	5.67	5.635
			5.652	5.687	5.652	5.617	5.687	5.768	5.814	
21	1	1	4.787	4.942	4.977	5.106	5.165	5.247	5.323	5.416
			5.394	5.371	5.438	5.521	5.521	5.561	5.635	5.617
			5.687	5.72	5.737	5.737	5.768	5.768	5.704	
22	1	1	4.605	4.828	4.828	4.977	5.043	5.165	5.22	5.273
			5.247	5.22	5.298	5.375	5.371	5.416	5.501	5.501
			5.521	5.561	5.617	5.635	5.72	5.737	5.768	
23	1	1	4.7	4.905	5.011	5.075	5.106	5.22	5.22	5.298
			5.323	5.347	5.416	5.472	5.501	5.541	5.598	5.598
			5.598	5.652	5.67	5.704	5.737	5.768	5.784	
24	1	1	4.745	4.942	5.011	5.075	5.106	5.247	5.273	5.323
			5.347	5.371	5.416	5.481	5.501	5.541	5.598	5.598
			5.635	5.687	5.704	5.72	5.829	5.844	5.9	
25	1	1	4.654	4.828	4.828	4.977	4.977	5.043	5.136	5.165
			5.165	5.165	5.193	5.204	5.22	5.273	5.371	5.347
			5.46	5.58	5.635	5.67	5.753	5.799	5.844	
26	1	1	4.828	4.977	5.011	5.106	5.165	5.22	5.273	5.323
			5.371	5.394	5.46	5.576	5.652	5.617	5.687	5.67
			5.72	5.784	5.784	5.784	5.829	5.814	5.844	

;

The mean response profiles of the cows are not of particular interest; what matters are inferences about the Iron effect, the Infection effect, and their interaction. Nevertheless, the body weight of the cows changes over the 22-month period, and

you need to account for these changes in the analysis. A reasonable approach is to apply the approximate low-rank smoother to capture the trends over time. This approach frees you from having to stipulate a parametric model for the response trajectories over time. In addition, you can test hypotheses about the smoothing parameter; for example, whether it should be varied by treatment.

The following statements fit a model with a  $2 \times 2$  factorial treatment structure and smooth trends over time, choosing the Newton-Raphson algorithm with ridging for the optimization.

```
proc glimmix data=cows;
  t2 = tpoint / 100;
  class cow iron infection;
  model weight = iron infection iron*infection tpoint;
  random t2 / type=rsmooth subject=cow
             knotmethod=kdtree(bucket=100 knotinfo);
  output out=gmxout pred(blup)=mu;
  nloptions tech=newrap;
run;
```

The continuous time effect appears in both the **MODEL** (tpoint) and the **RANDOM** statement (t2). Because the variance of the radial smoothing component is dependent on the temporal metric, the time scale was rescaled for the **RANDOM** effect to move the parameter estimate away from the boundary. The knots of the radial smoother are selected as the vertices of a  $k$ - $d$  tree. Specifying **BUCKET=100** sets the bucket size of the tree to  $b = 100$ . Since measurements at each time point are available for 26 (or 25) cows, this groups approximately four time points in a single bucket. The **KNOTINFO** keyword of the **KNOTMETHOD=** option requests a printout of the knot locations for the radial smoother. The **OUTPUT** statement saves the predictions of the mean of each observations to the data set **gmxout**. Finally, the **TECH=NEWRAP** option in the **NLOPTIONS** statement specifies the Newton-Raphson algorithm for the optimization technique.

#### Output 6.1. Repeated Measures Analysis

The GLIMMIX Procedure	
Model Information	
Data Set	WORK.COWS
Response Variable	weight
Response Distribution	Gaussian
Link Function	Identity
Variance Function	Default
Variance Matrix Blocked By	cow
Estimation Technique	Restricted Maximum Likelihood
Degrees of Freedom Method	Containment

The “Class Level Information” table in [Output 6.2](#) lists the number of levels of the Cow, Iron, and Infection effects.

**Output 6.2.** Repeated Measures Analysis (continued)

Class Level Information		
Class	Levels	Values
cow	26	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
iron	2	0 1
infection	2	0 1

In [Output 6.3](#), the “Radial Smoother Knots for RSmooth(t2)” table displays the knots computed from the vertices of the t2 *k-d* tree. Notice that knots are spaced unequally and that the extreme time points are among the knot locations. The “Number of Observations” table shows that one observation was not used in the analysis. The twelfth observation for cow 4 has a missing value.

**Output 6.3.** Repeated Measures Analysis (continued)

Radial Smoother Knots for RSmooth(t2)	
Knot Number	t2
1	0
2	0.04400
3	0.1250
4	0.2020
5	0.3230
6	0.4140
7	0.5050
8	0.6010
9	0.6590
Number of Observations Read 598	
Number of Observations Used 597	

The “Dimensions” table in [Output 6.4](#) shows that the model contains only two covariance parameters, the G-side variance of the spline coefficients ( $\sigma^2$ ) and the R-side scale parameter ( $\phi$ ). For each subject (cow), there are nine columns in the **Z** matrix, one per knot location. The GLIMMIX procedure processes these data by subjects (cows).

**Output 6.4.** Repeated Measures Analysis (continued)

Dimensions	
G-side Cov. Parameters	1
R-side Cov. Parameters	1
Columns in X	10
Columns in Z per Subject	9
Subjects (Blocks in V)	26
Max Obs per Subject	23

The “Optimization Information” table displays information about the optimization process. Because fixed effects and the residual scale parameter can be profiled from the optimization, the iterative algorithm involves only a single covariance parameter, the variance of the spline coefficients ([Output 6.5](#)).

**Output 6.5.** Repeated Measures Analysis (continued)

Optimization Information	
Optimization Technique	Newton-Raphson
Parameters in Optimization	1
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Profiled
Residual Variance	Profiled
Starting From	Data

After 11 iterations, the optimization process terminates ([Output 6.6](#)). In this case, the absolute gradient convergence criterion was met.

**Output 6.6.** Repeated Measures Analysis (continued)

Iteration History					
Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	-1302.549272	.	20.33682
1	0	3	-1451.587367	149.03809501	9.940495
2	0	3	-1585.640946	134.05357887	4.71531
3	0	3	-1694.516203	108.87525722	2.176741
4	0	3	-1775.290458	80.77425512	0.978577
5	0	3	-1829.966584	54.67612585	0.425724
6	0	3	-1862.878184	32.91160012	0.175992
7	0	3	-1879.329133	16.45094874	0.066061
8	0	3	-1885.175082	5.84594887	0.020137
9	0	3	-1886.238032	1.06295072	0.00372
10	0	3	-1886.288519	0.05048659	0.000198
11	0	3	-1886.288673	0.00015425	6.364E-7

Convergence criterion (ABSGCONV=0.00001) satisfied.

The generalized chi-square statistic in the “Fit Statistics” table is small for this model (Output 6.7). There is very little residual variation. The radial smoother is associated with 433.55 residual degrees of freedom, computed as 597 minus the trace of the smoother matrix.

**Output 6.7.** Repeated Measures Analysis (continued)

Fit Statistics	
-2 Res Log Likelihood	-1886.29
AIC (smaller is better)	-1882.29
AICC (smaller is better)	-1882.27
BIC (smaller is better)	-1879.77
CAIC (smaller is better)	-1877.77
HQIC (smaller is better)	-1881.56
Generalized Chi-Square	0.47
Gener. Chi-Square / DF	0.00
Radial Smoother df(res)	433.55

The “Covariance Parameter Estimates” table in Output 6.8 displays the solutions of the covariance parameters. The variance of the random spline coefficients is estimated as  $\hat{\sigma}^2 = 0.5961$ , and the scale parameter (=residual variance) estimate is  $\hat{\phi} = 0.0008$ .

**Output 6.8.** Repeated Measures Analysis (continued)

Covariance Parameter Estimates			
Cov Parm	Subject	Estimate	Standard Error
Var[RSmooth(t2)]	cow	0.5961	0.08144
Residual		0.000800	0.000059

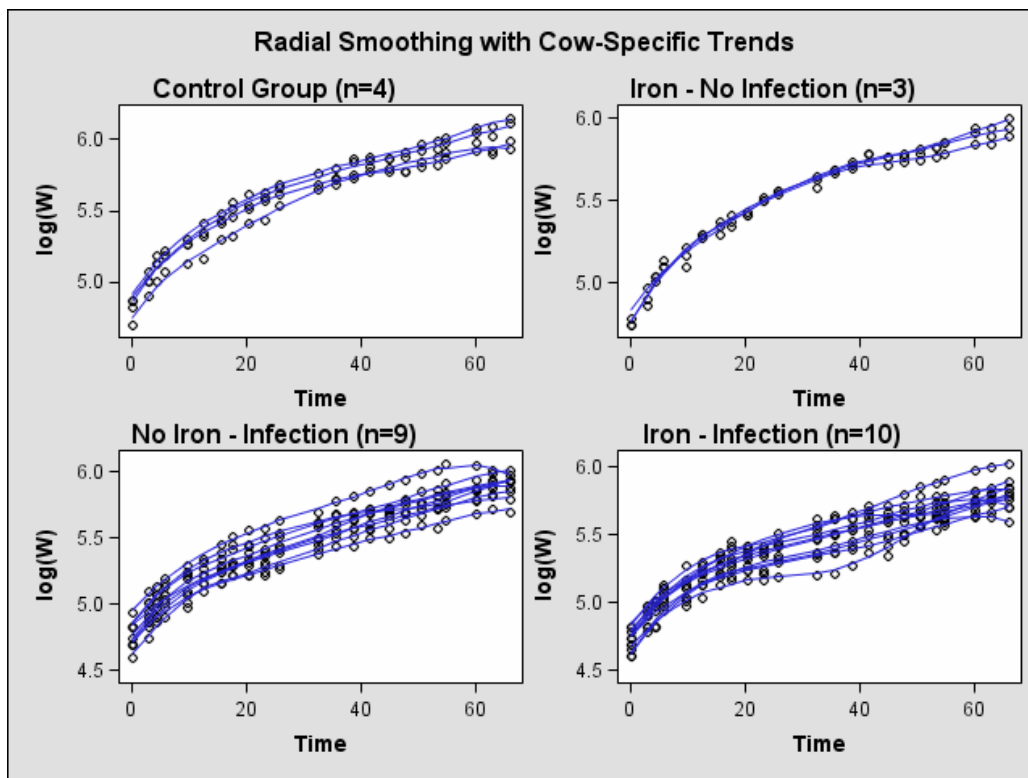
The “Type III Tests of Fixed Effects” table in Output 6.9 displays  $F$  tests for the fixed effects in the MODEL statement.

**Output 6.9.** Repeated Measures Analysis (continued)

Type III Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
iron	1	358	3.59	0.0588
infection	1	358	21.16	<.0001
iron*infection	1	358	0.09	0.7637
tpoint	1	358	53.88	<.0001

There is a strong infection effect as well as the absence of an interaction between infection with *M. paratuberculosis* and iron dosing.

The graph of the observed data and fitted profiles in the four groups is shown in Figure 24. The trends are quite smooth, and you can see how the radial smoother adapts to the cow-specific profile. This is the reason for the small scale parameter estimate,  $\hat{\phi} = 0.008$ . Comparing the panels on top to the panels at the bottom of Figure 24 reveals the effect of Infection. A comparison of the panels on the left to those on the right indicates the weak Iron effect.



**Figure 24.** Observed and Predicted Profiles

The smoothing parameter in this analysis is related to the covariance parameter estimates. Because there is only one radial smoothing variance component, the amount of smoothing is the same in all four treatment groups. To test whether the smoothing parameter should be varied by group, you can refine the analysis of the previous model. The following statements fit the same general model, but vary the covariance parameters by the levels of the Iron\*Infection interaction. This is accomplished with the GROUP= option of the **RANDOM** statement.

```
ods select OptInfo FitStatistics CovParms;
proc glimmix data=cows;
  t2 = tpoint / 100;
  class cow iron infection;
  model weight = iron infection iron*infection tpoint;
  random t2 / type=rsmooth
```

```

subject=cow
group=iron*infection
knotmethod=kdtree(bucket=100);
nloptions tech=newrap;
run;

```

All observations that have the same value combination of the Iron and Infection effects share the same covariance parameter. As a consequence, you obtain different smoothing parameters result in the four groups.

In [Output 6.10](#), the “Optimization Information” table shows that there are now four covariance parameters in the optimization, one spline coefficient variance for each group.

**Output 6.10.** Analysis with Group-Specific Smoothing Parameter

The GLIMMIX Procedure	
Optimization Information	
Optimization Technique	Newton-Raphson
Parameters in Optimization	4
Lower Boundaries	4
Upper Boundaries	0
Fixed Effects	Profiled
Residual Variance	Profiled
Starting From	Data

Varying this variance component by groups has changed the  $-2$  Res Log Likelihood from  $-1886.29$  to  $-1887.95$  ([Output 6.11](#)). The difference, 1.66, can be viewed (asymptotically) as the realization of a chi-square random variable with 3 degrees of freedom. The difference is not significant ( $p = 0.64586$ ).

**Output 6.11.** Analysis with Group-Specific Smoothing Parameter (continued)

Fit Statistics	
-2 Res Log Likelihood	-1887.95
AIC (smaller is better)	-1877.95
AICC (smaller is better)	-1877.85
BIC (smaller is better)	-1871.66
CAIC (smaller is better)	-1866.66
HQIC (smaller is better)	-1876.14
Generalized Chi-Square	0.48
Gener. Chi-Square / DF	0.00
Radial Smoother df(res)	434.72

The “Covariance Parameter Estimates” table in [Output 6.12](#) confirms that the estimates of the spline coefficient variance are quite similar in the four groups, ranging from 0.4788 to 0.7105.

**Output 6.12.** Analysis with Group-Specific Smoothing Parameter (continued)

Covariance Parameter Estimates				
Cov Parm	Subject	Group	Estimate	Standard Error
Var[RSmooth(t2)]	cow	iron*infection 0 0	0.4788	0.1922
Var[RSmooth(t2)]	cow	iron*infection 0 1	0.5152	0.1182
Var[RSmooth(t2)]	cow	iron*infection 1 0	0.4904	0.2195
Var[RSmooth(t2)]	cow	iron*infection 1 1	0.7105	0.1409
Residual			0.000807	0.000060

Finally, you can apply a different technique for varying the temporal trends among the cows. From [Figure 24](#) it appears that an assumption of parallel trends within groups might be reasonable. In other words, you can fit a model in which the “overall” trend over time in each group is modeled nonparametrically, and this trend is shifted up or down to capture the behavior of the individual cow. You can accomplish this with the following statements.

```
ods select OptInfo FitStatistics CovParms;
proc glimmix data=cows;
  t2 = tpoint / 100;
  class cow iron infection;
  model weight = iron infection iron*infection tpoint;
  random t2 / type=rsmooth
            subject=iron*infection
            knotmethod=kdtree(bucket=100);
  random intercept / subject=cow;
  output out=gmxout pred(blup)=mu;
  nloptions tech=newrap;
run;
```

There are now two subject effects in this analysis. The first **RANDOM** statement applies the radial smoothing and identifies the experimental conditions as the subject. For each condition, a separate realization of the random spline coefficients is obtained. The second **RANDOM** statement adds a random intercept to the trend for each cow. This random intercept results in the parallel shift of the trends over time.

The “Fit Statistics” table is shown in [Output 6.13](#). Because the parallel shift model is not nested within either one of the previous models, the models cannot be compared with a likelihood ratio test. However, you can draw on the other fit statistics.



**Output 6.13.** Analysis with Parallel Shifts

The GLIMMIX Procedure	
Fit Statistics	
-2 Res Log Likelihood	-1788.52
AIC (smaller is better)	-1782.52
AICC (smaller is better)	-1782.48
BIC (smaller is better)	-1788.52
CAIC (smaller is better)	-1785.52
HQIC (smaller is better)	-1788.52
Generalized Chi-Square	1.17
Gener. Chi-Square / DF	0.00
Radial Smoother df(res)	547.21

All statistics indicate that this model does not fit the data as well as the initial model that varies the spline coefficients by cow. The Pearson chi-square statistic is more than twice as large as in the previous model, indicating much more residual variation in the fit. On the other hand, this model generates only four sets of spline coefficients, one for each treatment group, and thus retains more residual degrees of freedom.

The “Covariance Parameter Estimates” table in [Output 6.14](#) displays the solutions for the covariance parameters. The estimate of the variance of the spline coefficients is not that different from the estimate obtained in the first model (0.5961). The residual variance, however, has more than doubled.

**Output 6.14.** Analysis with Parallel Shifts (continued)

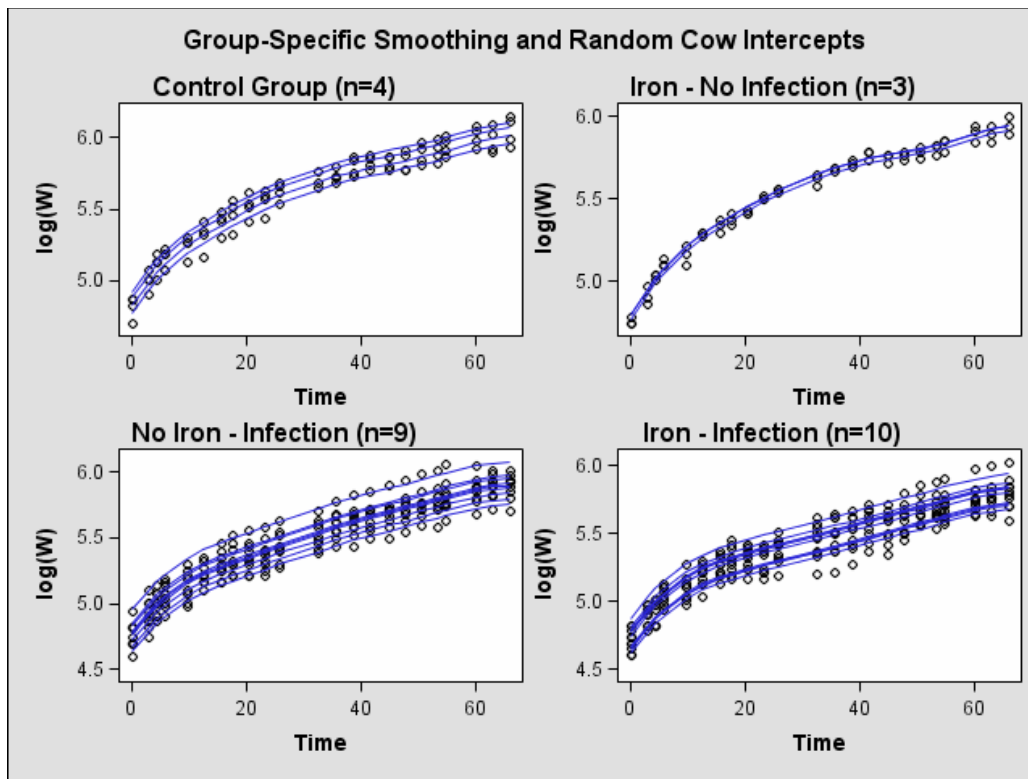
Covariance Parameter Estimates			
Cov Parm	Subject	Estimate	Standard Error
Var[RSmooth(t2)]	iron*infection	0.5398	0.1940
Intercept	cow	0.007122	0.002173
Residual		0.001976	0.000121

The Infection effect continues to be significant in this model; however, *p*-values are larger for all fixed effects as compared to the initial model ([Output 6.15](#)).

**Output 6.15.** Analysis with Parallel Shifts (continued)

Type III Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
iron	1	534	0.63	0.4288
infection	1	534	4.39	0.0365
iron*infection	1	534	0.02	0.8828
tpoint	1	534	8.34	0.0040

You can clearly see the parallel shifts of the nonparametric smooths in [Figure 25](#). In the groups receiving only iron or only an infection, the parallel lines assumption holds quite well. In the control group and the group receiving iron and the infection, the parallel shift assumption does not hold as well. Two of the profiles in the iron-only group are nearly indistinguishable.



**Figure 25.** Observed and Predicted Profiles

This example demonstrates that mixed-model smoothing techniques can be applied not only to achieve scatterplot smoothing, but also to longitudinal or repeated measures data. You can then use the `SUBJECT=` option in the `RANDOM` statement to obtain independent sets of spline coefficients for different subjects, and the `GROUP=` option in the `RANDOM` statement to vary the degree of smoothing across groups. Also, radial smoothers can be combined with other random effects. For the data con-

sidered here, the appropriate model is one with a single smoothing parameter for all treatment group and cow-specific spline coefficients.

## Example 7. Isotonic Contrasts for Ordered Alternatives

Dose response studies often focus on testing for monotone increasing or decreasing behavior in the mean values of the dependent variable. Hirotsu and Srivastava (2000) demonstrate one approach using data that originally appeared in Moriguchi (1976). The data consists of ferrite cores subjected to four increasing temperatures. The response variable is the magnetic force of each core.

```
data FerriteCores;
  do Temp = 1 to 4;
    do rep = 1 to 5; drop rep;
      input MagneticForce @@;
      output;
    end;
  end;
datalines;
10.8  9.9 10.7 10.4  9.7
10.7 10.6 11.0 10.8 10.9
11.9 11.2 11.0 11.1 11.3
11.4 10.7 10.9 11.3 11.7
;
```

It is of interest to test whether the magnetic force of the cores rises monotonically with temperature. The approach of Hirotsu and Srivastava (2000) depends on the lower confidence limits of the *isotonic contrasts* of the force means at each temperature, adjusted for multiplicity. The corresponding isotonic contrast compares the average a particular group and the preceding groups with the average of the succeeding groups. You can compute adjusted confidence intervals for isotonic contrasts using the `LSMESTIMATE` statement.

The following statements request an analysis of the `FerriteCores` data as a one-way design and multiplicity adjusted lower confidence limits for the isotonic contrasts. For the multiplicity adjustment, the `LSMESTIMATE` statement employs simulation, which provides adjusted *p*-values and lower confidence limits that are exact up to Monte Carlo error.

```
proc glimmix data=FerriteCores;
  class Temp;
  model MagneticForce = Temp;
  lsmestimate Temp
    'avg(1:1)<avg(2:4)' -3 1 1 1 divisor=3,
    'avg(1:2)<avg(3:4)' -1 -1 1 1 divisor=2,
    'avg(1:3)<avg(4:4)' -1 -1 -1 3 divisor=3
    / adjust=simulate(seed=1) cl upper;
  ods select LSMestimates;
run;
```

The results are shown in [Output 7.1](#)

**Output 7.1.** Analysis of LS-Means with Isotonic Contrasts

The GLIMMIX Procedure							
Least Squares Means Estimates							
Effect	Label	Estimate	Standard Error	DF	t Value	Pr > t	Adj P
Temp	avg(1:1) < avg(2:4)	0.8	0.191	16	4.20	0.0003	0.0010
Temp	avg(1:2) < avg(3:4)	0.7	0.165	16	4.24	0.0003	0.0009
Temp	avg(1:3) < avg(4:4)	0.4	0.191	16	2.10	0.0260	0.0625
Least Squares Means Estimates							
Effect	Label	Alpha	Lower	Upper	Adj Lower	Adj Upper	
Temp	avg(1:1) < avg(2:4)	0.05	0.4672	Infty	0.3771	Infty	
Temp	avg(1:2) < avg(3:4)	0.05	0.4118	Infty	0.3337	Infty	
Temp	avg(1:3) < avg(4:4)	0.05	0.06721	Infty	-0.02291	Infty	

With an adjusted  $p$ -value of 0.001, the magnetic force at the first temperature is significantly less than the average of the other temperatures. Likewise, the average of the first two temperatures is significantly less than the average of the last two ( $p = 0.0009$ ). However, the force at the last temperature is not significantly greater than the average of the others ( $p = 0.0625$ ). These results indicate a significant monotone increase over the first three temperatures, but not across all four temperatures.

## References

- Akaike, H. (1974), "A New Look at the Statistical Model Identification," *IEEE Transaction on Automatic Control*, AC-19, 716–723.
- Bahadur, R. R. (1961), "A Representation of the Joint Distribution of Responses to  $n$  Dichotomous Items," in *Studies in Item Analysis and Prediction*, ed. H. Solomon, Stanford, CA: Stanford University Press.
- Beale, E. M. L. (1972), "A Derivation of Conjugate Gradients," in *Numerical Methods for Nonlinear Optimization*, ed. F.A. Lootsma, London: Academic Press.
- Bell, R. M. and McCaffrey, D. F. (2002), "Bias Reduction in Standard Errors for Linear Regression with Multi-Stage Samples," *Survey Methodology*, 28, 169–181.
- Bozdogan, H. (1987), "Model Selection and Akaike's Information Criterion (AIC): The General Theory and Its Analytical Extensions," *Psychometrika*, 52, 345–370.
- Breslow, N. E. and Clayton, D. G. (1993), "Approximate Inference in Generalized Linear Mixed Models," *Journal of the American Statistical Association*, 88, 9–25.

- Brinkman, N. D. (1981), "Ethanol Fuel —A Single Engine Study of Efficiency and Exhaust Emission," *SAE Transactions*, 90, No. 810345, 1410–1424.
- Brown, H. and Prescott. R. (1999), *Applied Mixed Models in Medicine*, New York: John Wiley & Sons.
- Burnham, K. P. and Anderson, D. R. (1998), *Model Selection and Inference: A Practical Information-Theoretic Approach*, New York: Springer-Verlag.
- Clayton, D. and Kaldor, J. (1987) "Empirical Bayes Estimates of Age-standardized Relative Risks for Use in Disease Mapping," *Biometrics*, 43:671–681.
- Cleveland, W. S. and Grosse, E. (1991) "Computational Methods for Local Regression," *Statistics and Computing*, 1, 47–62.
- 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," *J. Optim. Theory Appl.*, 28, 453–482.
- Dennis, J. E. and Schnabel, R. B. (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Englewood, NJ: Prentice-Hall.
- Diggle, P. J., Liang, K.-Y., and Zeger, S. L. (1994), *Analysis of Longitudinal Data*, Oxford, UK: Oxford University Press.
- Dunnett, 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.
- Eilers, P. H. C. and Marx, B. D. (1996), "Flexible Smoothing with *B*-splines and Penalties," *Statistical Science*, 11, 89–121.
- Eskow, E. and Schnabel, R. B. (1991), "Algorithm 695: Software for a New Modified Cholesky Factorization," *Transactions on Mathematical Software*, 17(3), 306–312.
- Fai, A. H. T. and Cornelius, P. L. (1996), "Approximate *F*-tests of Multiple Degree of Freedom Hypotheses in Generalized Least Squares Analyses of Unbalanced Split-plot Experiments," *Journal of Statistical Computation and Simulation*, 54, 363–378.
- Fay, M. P. and Graubard, B. I. (2001), "Small-Sample Adjustments for Wald-Type Tests Using Sandwich Estimators," *Biometrics*, 57, 1198–1206.
- Ferrari. S. L. P. and Cribari-Neto, F. (2004), "Beta Regression for Modelling Rates and Proportions," *Journal of Applied Statistics*, 31, 799–815.
- Fletcher, R. (1987), *Practical Methods of Optimization*, Second Edition, Chichester: John Wiley & Sons, Inc.
- Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977) "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Transactions on Mathematical Software*, 3, 209–226.

- Fuller, W. A. (1976), *Introduction to Statistical Time Series*, New York: John Wiley & Sons, Inc.
- 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.
- Giesbrecht, F. G. and Burns, J. C. (1985), "Two-Stage Analysis Based on a Mixed Model: Large-sample Asymptotic Theory and Small-Sample Simulation Results," *Biometrics*, 41, 477–486.
- Gilliland, D. and Schabenberger, O. (2001), "Limits on Pairwise Association for Equi-Correlated Binary Variables," *Journal of Applied Statistical Sciences*, 10, 279–285.
- Goodnight, J. H. (1978), SAS Technical Report R-101, *Tests of Hypotheses in Fixed-Effects Linear Models*, Cary, NC: SAS Institute Inc.
- Goodnight, J. H. (1978), SAS Technical Report R-105, *Computing MIVQUE0 Estimates of Variance Components*, Cary, NC: SAS Institute Inc.
- Goodnight, J. H. (1979), "A Tutorial on the Sweep Operator," *American Statistician*, 33, 149–158.
- Goodnight, J. H. and Hemmerle, W. J. (1979), "A Simplified Algorithm for the W-Transformation in Variance Component Estimation," *Technometrics*, 21, 265–268.
- Gotway, C. A. and Stroup, W. W. (1997), "A Generalized Linear Model Approach to Spatial Data and Prediction," *Journal of Agricultural, Biological, and Environmental Statistics*, 2, 157–187.
- 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.
- Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J., and Ostrowski, E. (1994), *A Handbook of Small Data Sets*, London: Chapman and Hall.
- Hannan, E. J. and Quinn, B. G. (1979), "The Determination of the Order of an Autoregression," *Journal of the Royal Statistical Society, Series B*, 41, 190–195.
- Harville, D. A. and Jeske, D. R. (1992), "Mean Squared Error of Estimation or Prediction Under a General Linear Model," *Journal of the American Statistical Association*, 87, 724–731.
- Hemmerle, W. J. and Hartley, H. O. (1973), "Computing Maximum Likelihood Estimates for the Mixed AOV Model Using the W-Transformation," *Technometrics*, 15, 819–831.
- Henderson, C. R. (1984), *Applications of Linear Models in Animal Breeding*, University of Guelph.

- Hinkley, D. V. (1977), "Jackknifing in Unbalanced Situations," *Technometrics*, 19, 285–292.
- Hirotsu, C. and Srivastava, M. (2000), "Simultaneous Confidence Intervals Based on One-sided Max t Test," *Statistics and Probability Letters*, 49, 25–37.
- 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.
- Huber, P. J. (1967), "The Behavior of Maximum Likelihood Estimates Under Nonstandard Conditions," *Proc. Fifth Berkeley Symp. Math. Statist. Prob.*, 1, 221–233.
- Hurvich, C. M. and Tsai, C.-L. (1989), "Regression and Time Series Model Selection in Small Samples," *Biometrika*, 76, 297–307.
- Kackar, R. N. and Harville, D. A. (1984), "Approximations for Standard Errors of Estimators of Fixed and Random Effects in Mixed Linear Models," *Journal of the American Statistical Association*, 79, 853–862.
- Karim, M. Z. and Zeger, S. L. (1992), "Generalized Linear Models with Random Effects; Salamander Mating Revisited," *Biometrics*, 48:631–644.
- Kass, R. E. and Steffey, D. (1989), "Approximate Bayesian Inference in Conditionally Independent Hierarchical Models (Parametric Empirical Bayes Models)," *Journal of the American Statistical Association*, 84, 717–726.
- Kauermann, G. and Carroll, R. J. (2001), "A Note on the Efficiency of Sandwich Covariance Estimation," *Journal of the American Statistical Association*, 96, 1387–1396.
- 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.
- Liang, K. Y. and Zeger, S. L. (1986), "Longitudinal Data Analysis Using Generalized Linear Models," *Biometrika*, 73, 13–22.
- Long, J. S., and Ervin, L. H. (2000), "Using Heteroscedasticity Consistent Standard Errors in the Linear Regression Model," *The American Statistician*, 217–224.
- MacKinnon, J. G. and White, H. (1985), "Some Heteroskedasticity-Consistent Covariance Matrix Estimators With Improved Finite Sample Properties," *Journal of Econometrics*, 29, 305–325.

- Mancil, L. A. and DeRouen, T. A. (2001), "A Covariance Estimator for GEE with Improved Small-Sample Properties," *Biometrics*, 57, 126–134.
- McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, Second Edition, London: Chapman and Hall.
- McLean, R. A. and Sanders, W. L. (1988), "Approximating Degrees of Freedom for Standard Errors in Mixed Linear Models," *Proceedings of the Statistical Computing Section*, American Statistical Association, New Orleans, 50–59.
- McLean, R. A., Sanders, W. L., and Stroup, W. W. (1991), "A Unified Approach to Mixed Linear Models," *The American Statistician*, 45, 54–64.
- 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.
- Moriguchi, S. (ed.) (1976), *Statistical Method for Quality Control*, (in Japanese), Tokyo: Japan Standards Association.
- Mosteller, F. and Tukey, J. W. (1977), *Data Analysis and Regression*, Reading, MA: Addison-Wesley.
- Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society A*, 135, 370–384.
- 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  $\rho_{ij} = -\alpha_i\alpha_j$ ," *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," *Math. Prog.*, 12, 241–254.
- Prasad, N. G. N. and Rao, J. N. K. (1990), "The Estimation of Mean Squared Error of Small-Area Estimators," *Journal of the American Statistical Association*, 85, 163–171.
- Ruppert, D., Wand, M. P., and Carroll, R. J. (2003), *Semiparametric Regression*, Cambridge: Cambridge University Press.
- Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons, Inc.



- Schabenberger, O. and Gregoire, T. G. (1996), “Population-averaged and Subject-specific Approaches for Clustered Categorical Data,” *Journal of Statistical Computation and Simulation*, 54, 231–253.
- Schabenberger, O. Gregoire, T. G., and Kong, F. (2000), “Collections of Simple Effects and Their Relationship to Main Effects and Interactions in Factorials,” *The American Statistician*, 54, 210–214.
- Schabenberger, O. and Pierce, F. J. (2002), *Contemporary Statistical Models for the Plant and Soil Sciences*, Boca Raton, FL: CRC Press.
- Schall, R. (1991), “Estimation in Generalized Linear Models with Random Effects,” *Biometrika*, 78:719–727.
- Schluchter, M. D. and Elashoff, J. D. (1990), “Small-Sample Adjustments to Tests with Unbalanced Repeated Measures Assuming Several Covariance Structures,” *Journal of Statistical Computation and Simulation*, 37, 69–87.
- Schwarz, G. (1978), “Estimating the Dimension of a Model,” *Annals of Statistics*, 6, 461–464.
- Shaffer, J. P. (1986), “Modified Sequentially Rejective Multiple Test Procedures,” *Journal of the American Statistical Association*, 81:329–335.
- Shun, Z. (1997), “Another Look at the Salamander Mating Data: A Modified Laplace Approximation Approach,” *Journal of the American Statistical Association*, 92:341–349.
- Stenstrom, F. H. (1940), “The Growth of Snapdragons, Stocks, Cinerarias and Carnations on Six Iowa Soils,” Master’s thesis, Iowa State College.
- 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.
- Verbeke, G. and Molenberghs, G. (2000), *Linear Mixed Models for Longitudinal Data*, Berlin-Heidelberg-New York: Springer-Verlag.
- Vonesh, E.F. and Chinchilli, V. M. (1997) *Linear and Nonlinear Models for the Analysis of Repeated Measurements*, New York: Marcel Dekker.
- Wedderburn, R. W. M. (1947) “Quasilikelihood Functions, Generalized Linear Models and the Gauss-Newton Method,” *Biometrika*, 61, 439–447.
- 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. J. and Young, S. S. (1993), *Resampling-based Multiple Testing*, New York: John Wiley & Sons, Inc.
- Westfall, P. H., Tobias, R. D., Rom, D., Wolfinger, R. D., and Hochberg, Y. (1999), *Multiple Comparisons and Multiple Tests Using the SAS System*, SAS Institute Inc., Cary, NC.
- White, H. (1980), “A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity,” *Econometrica*, 48, 817–838.

- Winer, B. J. (1971), *Statistical Principles in Experimental Design*, Second Edition, New York: McGraw-Hill, Inc.
- Wolfinger, R. and O'Connell, M. (1993), "Generalized Linear Mixed Models: A Pseudo-Likelihood Approach", *Journal of Statistical Computation and Simulation*, 4, 233–243.
- Wolfinger, R., Tobias, R., and Sall, J. (1994) "Computing Gaussian Likelihoods and Their Derivatives for General Linear Mixed Models," *SIAM Journal on Scientific Computing*, 15(6), 1294–1310.
- Zeger, S. L. and Liang, K.-Y. (1986), "Longitudinal Data Analysis for Discrete and Continuous Outcomes," *Biometrics*, 42, 121–130.