

SAS/STAT[®] 15.1

User's Guide

The BGLIMM Procedure

This document is an individual chapter from *SAS/STAT® 15.1 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS/STAT® 15.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 15.1 User's Guide

Copyright © 2018, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

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

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

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

November 2018

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

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

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to <http://support.sas.com/thirdpartylicenses>.

Chapter 28

The BGLIMM Procedure

Contents

| | |
|-----------------------------------------------------------|-------------|
| Overview: BGLIMM Procedure | 1206 |
| Basic Features | 1207 |
| Notation for the Generalized Linear Mixed Model | 1207 |
| PROC BGLIMM Compared with Other SAS Procedures | 1209 |
| Getting Started: BGLIMM Procedure | 1210 |
| Logistic Regression with Random Intercepts | 1210 |
| Syntax: BGLIMM Procedure | 1218 |
| PROC BGLIMM Statement | 1218 |
| BY Statement | 1227 |
| CLASS Statement | 1227 |
| ESTIMATE Statement | 1229 |
| MODEL Statement | 1231 |
| RANDOM Statement | 1238 |
| REPEATED Statement | 1248 |
| Details: BGLIMM Procedure | 1251 |
| Generalized Linear Mixed Models | 1251 |
| Response Probability Distributions | 1253 |
| Likelihood | 1254 |
| Scale and Dispersion Parameters | 1256 |
| How PROC BGLIMM Works | 1256 |
| GLM Parameters | 1257 |
| GLMM with Random Effects | 1257 |
| Models with Missing Values | 1258 |
| Sampling Methods | 1258 |
| Conjugate Sampling | 1258 |
| Gamerman Algorithm | 1259 |
| Hamiltonian Monte Carlo Sampler | 1260 |
| Prior Distributions | 1261 |
| Prior for the Fixed-Effects Coefficients | 1261 |
| Prior for the Random-Effects Coefficients | 1262 |
| Prior for the G-Side Covariance | 1262 |
| Prior for the Scale Parameter | 1264 |
| Prior for the R-Side Covariance | 1264 |
| Treatment of Subjects in the RANDOM Statement | 1264 |
| Initial Values of the Markov Chains | 1265 |
| Missing Data | 1266 |

| | |
|----------------------------------------------------------------------------|-------------|
| Autocall Macros for Postprocessing | 1266 |
| Regenerating Diagnostics Plots | 1268 |
| Displayed Output | 1269 |
| Model- and Data-Related ODS Tables | 1269 |
| Sampling-Related ODS Tables | 1270 |
| ODS Tables Related to Posterior Statistics | 1270 |
| ODS Tables Related to Convergence Diagnostics | 1271 |
| ODS Table Names | 1272 |
| ODS Graphics | 1274 |
| Examples: BGLIMM Procedure | 1275 |
| Example 28.1: Normal Regression with Repeated Measurements | 1275 |
| Example 28.2: Mating Experiment with Crossed Random Effects | 1282 |
| Example 28.3: Poisson Regression with Random Effects | 1290 |
| Example 28.4: Repeated Growth Measurements with Group Difference | 1296 |
| References | 1305 |

Overview: BGLIMM Procedure

The BGLIMM procedure is a high-performance, sampling-based procedure that provides Bayesian inference for generalized linear mixed models (GLMMs).

GLMMs are hierarchical models that combine a generalized linear model with normally distributed random effects. Conditional on the random effects, data have distributions in the exponential family (binary, binomial, Poisson, normal, gamma, and so on). GLMMs are widely used in practice and are especially useful in applications where the data consist of collections of units and are hierarchically structured. Variations of GLMMs offer modeling flexibility that enables you to capture the complex nature of real-world data.

In a classical (frequentist) approach, the fixed-effects parameters are considered fixed with an unknown mean and are the primary focus of inference. The random effects are treated as unobserved latent variables. Estimation is achieved by maximizing the marginal likelihood of the fixed-effects parameters while integrating out the random effects (Davidian and Giltinan 1995; Vonesh, Chinchilli, and Pu 1996). Typically, asymptotic normality is assumed in inference.

In contrast, the Bayesian approach estimates the joint posterior distribution of all parameters in a model, including all fixed- and random-effects parameters. The Monte Carlo method numerically integrates out the random effects and propagates the uncertainties to the marginal posterior of the fixed-effects parameters. PROC BGLIMM uses efficient Markov chain Monte Carlo (MCMC) sampling tools to estimate the posterior marginal distributions and use them for further inference.

For a short introduction to Bayesian analysis and related basic concepts, see Chapter 7, “[Introduction to Bayesian Analysis Procedures](#).” For discussions of the relative advantages and disadvantages of the Bayesian paradigm, see the section “[Bayesian Analysis: Advantages and Disadvantages](#)” on page 136 in Chapter 7, “[Introduction to Bayesian Analysis Procedures](#).” For a guide to Bayesian textbooks of varying degrees of difficulty, see the section “[A Bayesian Reading List](#)” on page 161 in Chapter 7, “[Introduction to Bayesian Analysis Procedures](#).”

PROC BGLIMM uses syntax similar to that of PROC MIXED and PROC GLIMMIX in specifying a GLMM. You use the **MODEL** statement to specify the distribution and link function, the **RANDOM** statement to specify the random effects, the **CLASS** statement to specify categorical variables, the **REPEATED** statement to specify the correlation of longitudinal responses, and the **ESTIMATE** statements for inferences. PROC BGLIMM draws samples from the target distributions, computes summary and diagnostic statistics, and saves the posterior samples in an output data set that you can use for further analysis.

Basic Features

The basic features of PROC BGLIMM include the following:

- GLMMs with univariate or multivariate dimensional random effects
- nested or non-nested hierarchical models
- repeated measures models (balanced or unbalanced data) with normal data
- suite of covariance structures for random effects and residuals, including variance components, compound symmetry, unstructured, AR(1), Toeplitz, autoregressive, and many more
- built-in prior distributions for regression coefficients and covariance parameters
- the ability to model heterogeneity in covariance structures
- the ability to produce estimates and credible intervals for estimable linear combination of effects
- support for missing completely at random (MCAR) and missing at random (MAR) approaches in modeling missing data
- multithreading of optimal sampling algorithms for fast performance
- the ability to save posterior samples to an output data set for use in further inferences

PROC BGLIMM uses the Output Delivery System (ODS) to display and control the output. ODS enables you to convert any of the output from PROC BGLIMM to a SAS data set. For more information, see the section “[ODS Table Names](#)” on page 1272.

PROC BGLIMM uses ODS Graphics to create graphs as part of its output. For specific information about the statistical graphics available with the BGLIMM procedure, see the [PLOTS](#) options in [PROC BGLIMM](#).

Notation for the Generalized Linear Mixed Model

This section introduces the mathematical notation that the chapter uses to describe the generalized linear mixed model. For a description of the statistical details and sampling algorithms, see the section “[Details: BGLIMM Procedure](#)” on page 1251.

First consider the simple normal linear model. The quantity of primary interest, y_i , is called the response or outcome variable for the i th individual. The variable \mathbf{x}_i is the $1 \times p$ covariate vector for the fixed effects. The distribution of y_i given \mathbf{x}_i is normal with a mean that is a linear function of \mathbf{x}_i ,

$$y_i = \mathbf{x}_i \boldsymbol{\beta} + \epsilon_i, \quad i = 1, \dots, I$$

$$\epsilon_i \sim N(0, \sigma^2)$$

where $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients (also known as fixed effects) and ϵ_i is the noise with a variance σ^2 .

The normal linear model can be expanded to include random effects, and the model becomes a normal linear mixed model,

$$y_i = \mathbf{x}_i \boldsymbol{\beta} + \mathbf{z}_i \boldsymbol{\gamma}_i + \epsilon_i$$

$$\boldsymbol{\gamma}_i \sim N(\mathbf{0}, \mathbf{G}_i)$$

$$\epsilon_i \sim N(0, \sigma^2)$$

where $\boldsymbol{\gamma}_i$ is a $q \times 1$ vector of random effects, \mathbf{z}_i is a $1 \times q$ matrix of covariates for the $\boldsymbol{\gamma}_i$, and \mathbf{G}_i is the covariance matrix of the random effects $\boldsymbol{\gamma}_i$ (\mathbf{G} is a block diagonal matrix where each block is \mathbf{G}_i).

When an individual i has n_i repeated measurements, the random-effects model for outcome vector \mathbf{y}_i is given by

$$\mathbf{y}_i = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{Z}_i \boldsymbol{\gamma}_i + \boldsymbol{\epsilon}_i, \quad i = 1, \dots, I$$

where \mathbf{y}_i is $n_i \times 1$, \mathbf{X}_i is an $n_i \times p$ matrix of fixed covariates, $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients (also known as fixed effects), $\boldsymbol{\gamma}_i$ is a $q \times 1$ vector of random effects, \mathbf{Z}_i is an $n_i \times q$ matrix of covariates for the $\boldsymbol{\gamma}_i$, and $\boldsymbol{\epsilon}_i$ is an $n_i \times 1$ vector of random errors.

It is further assumed that

$$\boldsymbol{\gamma}_i \sim N(\mathbf{0}, \mathbf{G}_i)$$

$$\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \mathbf{R}_i)$$

where \mathbf{G}_i is the covariance matrix of $\boldsymbol{\gamma}_i$ (\mathbf{G} is a block diagonal matrix where each block is \mathbf{G}_i) and \mathbf{R}_i is the covariance matrix of the residual errors for the i th subject (\mathbf{R} is a block diagonal matrix where each block is \mathbf{R}_i).

There are cases where the relationship between the design matrix (\mathbf{X} and \mathbf{Z}) and the expectation of the response is not linear, or where the distribution for the response is far from normal, even after transformation of the data. The class of generalized linear mixed models unifies the approaches that you need in order to analyze data in those cases. Let \mathbf{y} be the collection of all \mathbf{y}_i ; and let \mathbf{X} and \mathbf{Z} be the collection of all \mathbf{X}_i and \mathbf{Z}_i , respectively. A generalized linear mixed model consists of the following:

- the linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$
- the link function $g(\cdot)$ that relates the linear predictor to the mean of the outcome via a monotone link function,

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

where $g(\cdot)$ is a differentiable monotone link function and $g^{-1}(\cdot)$ is its inverse

- a response distribution in the exponential family of distributions. The distribution can also depend on a scale parameter, ϕ .

The conditional distribution of the response variable, given \boldsymbol{y} , is a member of the exponential family of distributions, including the normal distribution. You specify the distribution by using the **DIST=** option in the **MODEL** statement and specify the link function $g(\cdot)$ by using the **LINK=** option.

The BGLIMM procedure distinguishes two types of covariance structure: the “G-side” and the “R-side.” The G-side matrix is the covariance matrix of the random effects; the R-side matrix is the covariance matrix of the residuals. Models without G-side effects are also known as marginal (or population-averaged) models.

The columns of **X** are constructed from effects that are listed on the right side in the **MODEL** statement. Columns of **Z** and the G-side covariance matrix **G** are constructed from the **RANDOM** statement. The R-side covariance matrix **R** is constructed from the **REPEATED** statement, or from the **RANDOM** statement with the **RESIDUAL** option.

By default, the **R** matrix is the scaled identity matrix, $\mathbf{R} = \phi \mathbf{I}$. The scale parameter ϕ is set to 1 if the distribution does not have a scale parameter, such as in the case of the binary, binomial, Poisson, and exponential distributions.

For the normal distribution, for which you can specify various types of covariance structure for **R**, use the **REPEATED** statement. 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 following statement:

```
repeated Time / type=ar(1) subject=Patient;
```

Unknown quantities subject to estimation are the fixed-effects parameter vector $\boldsymbol{\beta}$, the random-effects parameter \boldsymbol{y} , and the covariance parameters that constitute all unknowns in **G** and **R**.

PROC BGLIMM Compared with Other SAS Procedures

Both PROC BGLIMM and PROC GENMOD can fit Bayesian generalized linear models. The posterior estimates are similar but not exact because of the nature of simulation and the different algorithms that the procedures use.

PROC MIXED and PROC GLIMMIX are two mixed modeling procedures that are related to PROC BGLIMM. All three procedures share some common syntax. PROC BGLIMM provides Bayesian solutions to GLMMs, whereas PROC MIXED and PROC GLIMMIX provide classical statistics solutions to the linear and generalized linear mixed-effects models, respectively.

Frequentist estimation methods rely on maximizing the marginal likelihood function, and inferences are often based on asymptotic theorems. This reflects frequentist assumptions in statistical inference—namely, that parameters are fixed quantities and uncertainties are the result of sampling.

PROC BGLIMM provides Bayesian solutions, and it estimates, via simulation, the joint posterior distributions of all parameters (fixed-effects, random-effects, covariance), conditional on the data. This approach reflects assumptions in the Bayesian paradigm—namely, that parameters are random and you use probability statements to quantify uncertainty associated with these random variables.

In practice, Bayesian estimates from PROC BGLIMM that use noninformative prior distributions closely resemble those that are obtained from PROC MIXED in linear mixed-effects models, a result due to linear

model theory and the normality assumption. You can expect larger discrepancies between PROC BGLIMM and PROC GLIMMIX results, especially in situations where the normal approximation to the marginal likelihood function is inaccurate.

In terms of run time, the sampling-based BGLIMM procedure could run slower than the mixed modeling procedures. The dimension of the regression problem can also hinder MCMC convergence. On the other hand, if a complex hierarchical model requires high-dimensional and computationally costly integral approximation methods in mixed procedures, using PROC BGLIMM can lead to better performance.

Model specifications in PROC BGLIMM are mostly identical to those in PROC MIXED and PROC GLIMMIX. You can expect the same construction of the design matrix, the covariance types, and the group structure in PROC BGLIMM. One important difference is in how the procedures handle wrongly specified models, such as singular or nearly singular design matrices. PROC BGLIMM requires the associated covariance matrix to be strictly positive definite; PROC MIXED and PROC GLIMMIX might rely on numerical methods (such as the general inverse) to bypass such difficulties in obtaining estimates.

PROC BGLIMM is similar to PROC MCMC in that both procedures use MCMC methods to estimate posterior distributions. PROC MCMC is a more general procedure that can handle a wider range of Bayesian statistical models. For example, PROC MCMC enables you to fit a model that includes random effects that are not normally distributed, which is something that PROC BGLIMM does not support. But because PROC MCMC relies on more detailed user inputs to construct a model, it uses generic sampling algorithms that are not tailored to specific models. On the other hand, PROC BGLIMM provides convenient access to Bayesian analysis of complex mixed models, with improved performance that results from using optimal sampling algorithms.

Getting Started: BGLIMM Procedure

This “Getting Started” section presents an example of a random-effects logistic model.

Logistic Regression 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 from the procedures. In each center, n_A patients were randomly selected and assigned to treatment group A, and n_B patients were randomly assigned to treatment group B. The following DATA step creates the data set, MultiCenter, for the analysis:

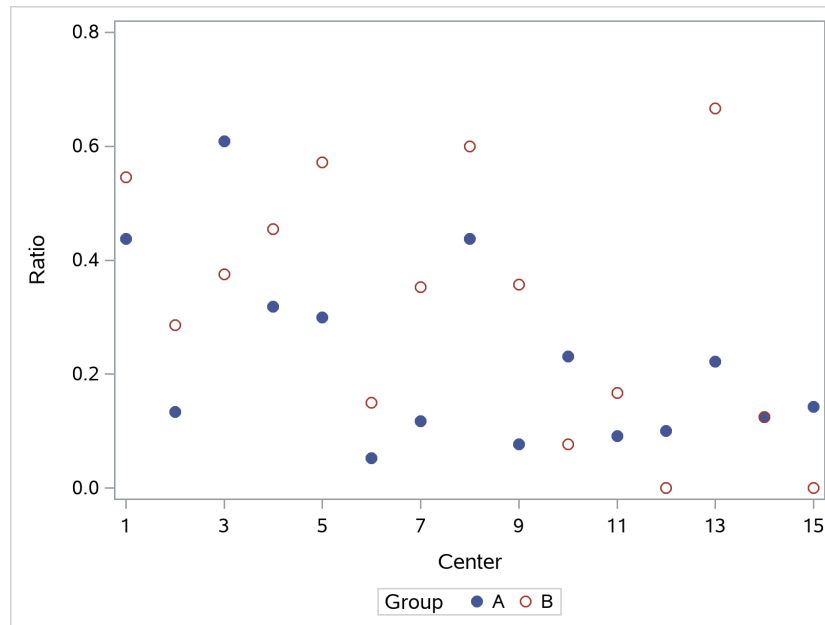
```
data MultiCenter;
  input Center Group$ N SideEffect @@;
  datalines;
1  A  32  14  1  B  33  18
2  A  30   4  2  B  28   8
3  A  23  14  3  B  24   9
4  A  22   7  4  B  22  10
5  A  20   6  5  B  21  12
6  A  19   1  6  B  20   3
7  A  17   2  7  B  17   6
```


| | | | | | | | |
|----|---|----|---|----|---|----|---|
| 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 |

i

The variable Group identifies the two procedures, N is the number of patients who received a given procedure at a particular center, and SideEffect is the number of patients who reported side effects. Side effects ratios, from different centers, are shown in Figure 28.1. Data from group A are represented by solid circles; data from group B are represented by clear circles.

Figure 28.1 MultiCenter Data



If y_{iA} and y_{iB} denote the number of patients at 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 having side effects from the two procedures, p_{iA} and p_{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:

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

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

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

Because p_{iA} is the conditional mean of the sample proportion, $E[y_{iA}/n_{iA}|\gamma_i] = p_{iA}$, 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:

```
ods graphics on;
proc bglimm data=MultiCenter nmc=10000 thin=2 seed=976352
  plots=all;
  class Center Group;
  model SideEffect/N = Group / noint;
  random int / subject = Center;
run;
```

The ODS GRAPHICS ON statement invokes the ODS Graphics environment, and the PLOTS=ALL option displays the diagnostic plots—the trace, autocorrelation function, and kernel density plots of the posterior samples.

The PROC BGLIMM statement invokes the procedure, and the DATA= option specifies the input data set MultiCenter. The NMC= option specifies the number of posterior simulation iterations in the main simulation loop after burn-in (the default number of burn-in iterations is 500). The THIN= option controls the thinning of the Markov chain and specifies that one of every two samples be kept.¹ The SEED= option specifies a random number generator seed, which reproduces the results.

The CLASS statement treats the variables Center and Group as classification variables.

The MODEL statement specifies the response variable as a sample proportion by 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. Note that because of the *events/trials* syntax, the BGLIMM procedure defaults to the binomial distribution, and that distribution's default link is the logit link. The MODEL statement specifies an independent fixed-effects variable (Group). An intercept is included in the fixed-effects model by default. You can remove it by using the NOINT option, so that you can obtain the estimates for both treatment groups A and B.

Along with the MODEL statement for fixed effects, the RANDOM statement is used for random effects. It specifies that the linear predictor contains an intercept term that randomly varies at the level of the Center effect. The SUBJECT=CENTER option in the RANDOM statement defines Center as a subject index for the random-effects grouping, so that each center has its own intercept. In other words, a random intercept is drawn separately and independently for each center in the study.

The “Model Information Table” in Figure 28.2 summarizes important information about the model that you fit and about aspects of the sampling technique. Information includes the input data set, response variable, likelihood distribution, link function, sampling algorithm, MCMC burn-in size, simulation size, thinning rate, and random number seed.

¹Thinning is often used to reduce correlation among posterior sample draws. For more information about Markov chain sample size, burn-in, and thinning, see the section “Burn-In, Thinning, and Markov Chain Samples” on page 143 in Chapter 7, “Introduction to Bayesian Analysis Procedures.”

Figure 28.2 Model Information
The BGLIMM Procedure

| Model Information | |
|-------------------------|---------------------|
| Data Set | WORK.MULTICENTER |
| Response Variable | SideEffect |
| Distribution | Binomial |
| Link Function | Logit |
| Fixed Effects Included | Yes |
| Random Effects Included | Yes |
| Sampling Algorithm | Gamerman, Conjugate |
| Burn-In Size | 500 |
| Simulation Size | 10000 |
| Thinning | 2 |
| Random Number Seed | 976352 |
| Number of Threads | 1 |

PROC BGLIMM 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.

Figure 28.3 displays two tables. The “Class Level Information” table lists the levels of the variables that are specified in the CLASS statement and the ordering of the levels. The “Number of Observations” table displays the number of observations that are read and used in the analysis.

Figure 28.3 Class Level Information and Number of Observations

| 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 | |
|-----------------------------|-----|
| Number of Observations Read | 30 |
| Number of Observations Used | 30 |
| Number of Events | 155 |
| Number of Trials | 503 |

Two variables are listed in the CLASS statement. The Center variable has 15 levels, and the Group variable has 2 levels. Because the response is specified through the *events/trial* syntax, the “Number of Observations” table also contains the total number of events and trials that are used in the analysis.

In Figure 28.4, the “Posterior Summaries and Intervals” table lists the summary statistics (posterior means, standard deviations, and HPD intervals) for each parameter, the fixed coefficients (β), and the variance of the random center intercepts (σ_c^2). Posterior summary statistics of random-effects parameters are not displayed by default. You can display them by using the MONITOR option in the RANDOM statement.

Figure 28.4 Posterior Summaries and Intervals

| Posterior Summaries and Intervals | | | | | |
|-----------------------------------|------|----------|-----------|--------------|---------|
| Parameter | N | Standard | | 95% | |
| | | Mean | Deviation | HPD Interval | |
| Group A | 5000 | -1.3895 | 0.3102 | -2.0071 | -0.7956 |
| Group B | 5000 | -0.8839 | 0.2968 | -1.4819 | -0.3186 |
| Random Var | 5000 | 0.9184 | 0.4198 | 0.3024 | 1.7515 |

It is important to check for convergence of all model parameters, because chains that are not converged can lead to biased conclusions. PROC BGLIMM produces a number of convergence diagnostics. By default, it displays effective sample sizes (ESS; Kass et al. 1998) of each parameter (Figure 28.5). ESS is theoretically equivalent to the number of independent samples that are generated directly from a target distribution.

Figure 28.5 PROC BGLIMM Effective Sample Size

| Effective Sample Sizes | | | |
|------------------------|--------|-----------------|------------|
| Parameter | ESS | Autocorrelation | |
| | | Time | Efficiency |
| Group A | 637.5 | 7.8432 | 0.1275 |
| Group B | 649.1 | 7.7028 | 0.1298 |
| Random Var | 2213.4 | 2.2590 | 0.4427 |

PROC BGLIMM produces a number of graphs, shown in Figure 28.6, that also help your convergence diagnostic checks. The trace plots show the stability of the Markov chain over the simulation and a number of whiskers in the tail areas, indicating good mixing of the chain. The autocorrelation plots show a decrease in autocorrelations, and the kernel density plots estimate the posterior marginal distributions for each parameter.

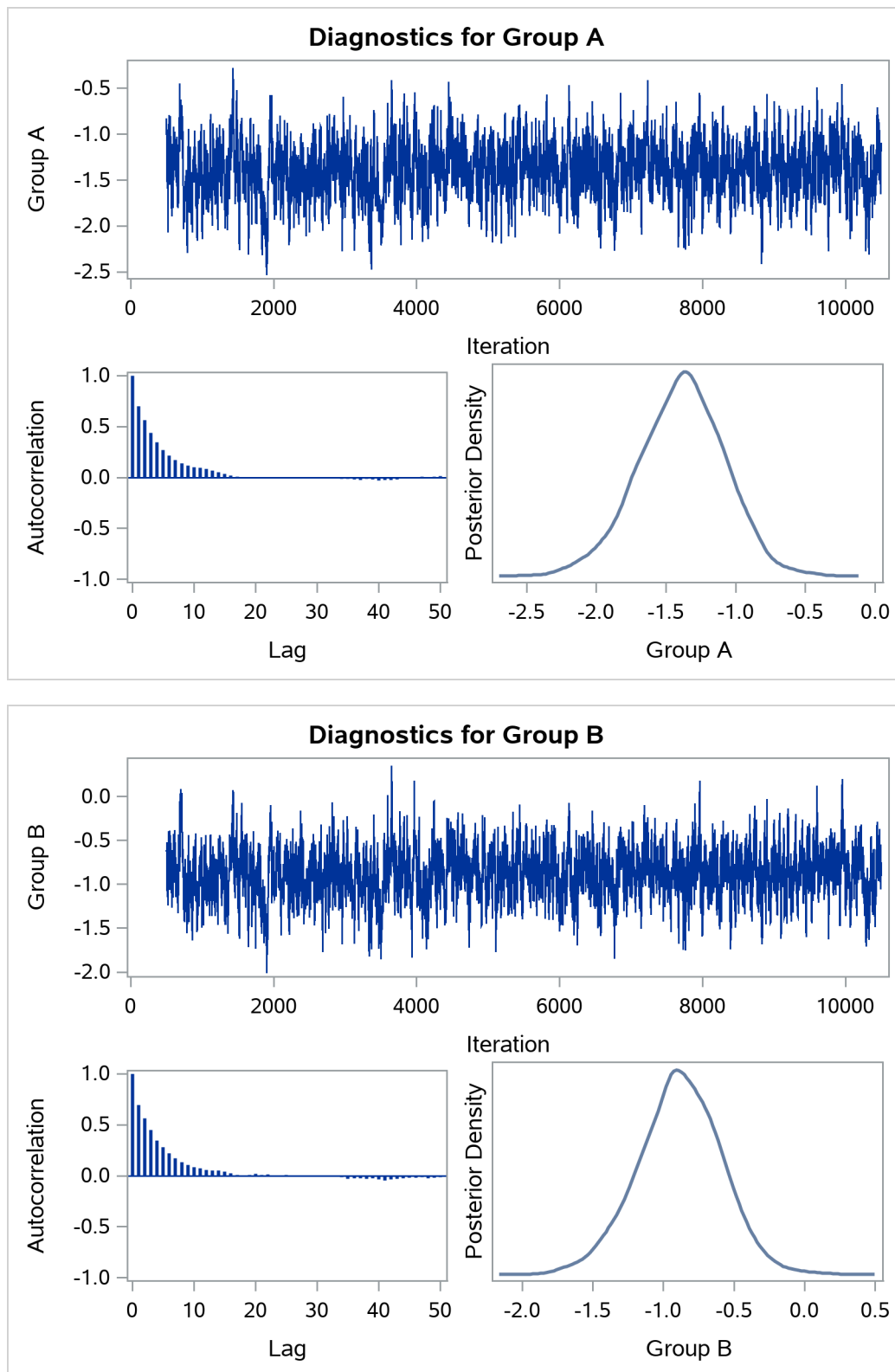
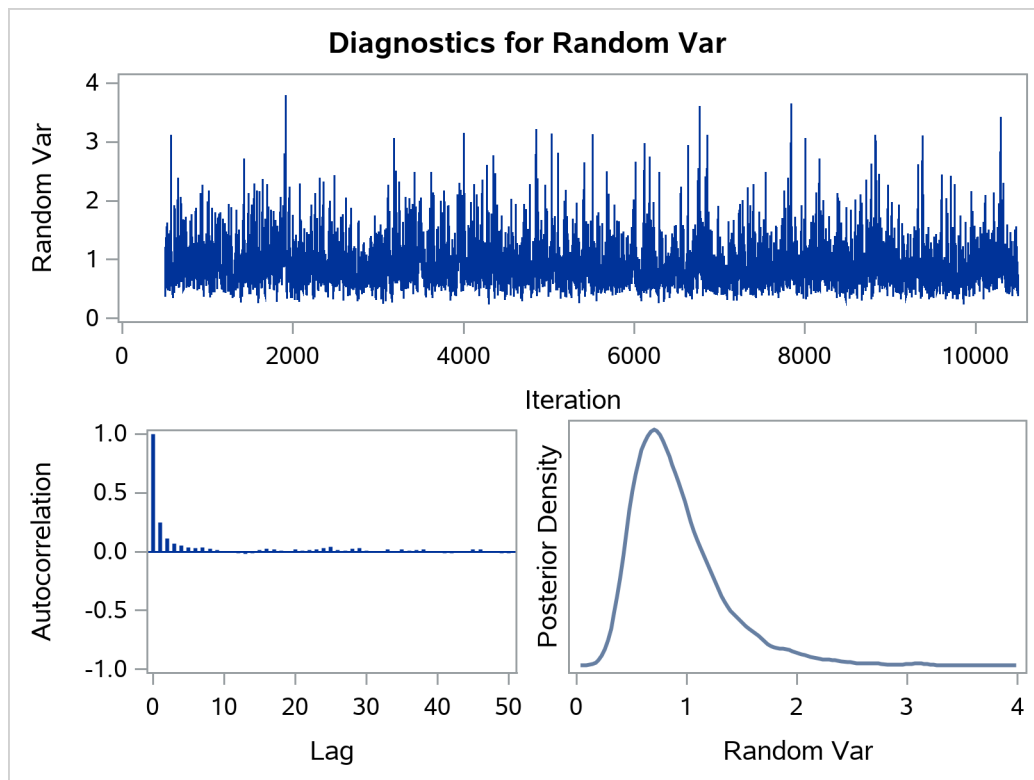
Figure 28.6 PROC BGLIMM Diagnostic Plots

Figure 28.6 continued



You can use the **ESTIMATE** statement as follows to compute the log of odds ratios between the two treatment groups, A and B:

```
proc bglimm data=MultiCenter nmc=10000 thin=2 seed=976352
  outpost=CenterOut;
  class Center Group;
  model SideEffect/N = Group / noint;
  random int / subject=Center monitor;
  estimate "log OR" group 1 -1;
run;
```

The **ESTIMATE** statement computes $\beta_A - \beta_B$ by using every posterior draw of the parameters and saves the values in the **OUTPOST=** data set under the variable name **Log_or**.

The “Estimated Differences in the Logits” table in Figure 28.7 shows that the posterior mean of the log of odds ratio is around -0.5 , with the 95% HPD interval all negative. This indicates less chance of developing side effects among patients who undergo procedure A than among those who undergo procedure B.

Figure 28.7 Estimated Differences in the Logits
The BGLIMM Procedure

| Results from ESTIMATE Statements | | | | |
|----------------------------------|---------|-----------------------|---------------------|---------|
| Label | Mean | Standard Deviation | 95% HPD Interval | |
| log OR | -0.5056 | 0.2087 | -0.9292 | -0.1102 |

The **ESTIMATE** statement does not compute the difference in probabilities of side effects directly. You can use the posterior samples and compute the probability differences directly. The following DATA step computes the probability difference between the two groups and saves it to the variable pDiff:

```
data prob;
  set CenterOut;
  pDiff = logistic(group_a) - logistic(group_b);
run;
```

You can use the **%SUMINT** autocall macro to compute the posterior summary statistics of pDiff:

```
%sumint(data=prob, var=pDiff)

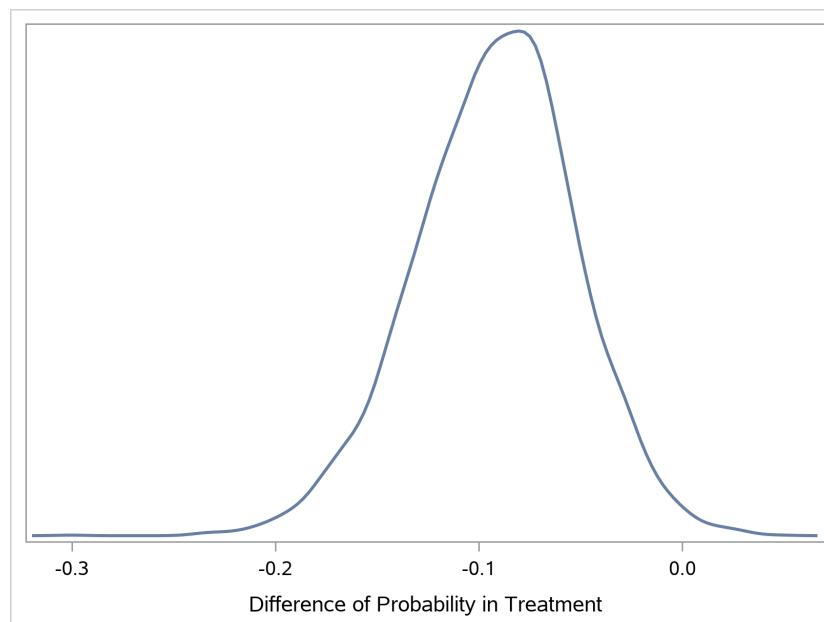
proc sgplot data=prob noautolegend;
  yaxis display=(nolabel noline noticks novalues);
  xaxis label="Difference of Probability in Treatment";
  density pDiff / type=kernel;
run;
```

The results are shown in [Figure 28.8](#). The density plot of the posterior distribution is shown in [Figure 28.9](#). There is a significant difference between the two groups.

Figure 28.8 Posterior Summary Statistics

| Posterior Summaries and Intervals | | | | | |
|-----------------------------------|------|---------|-----------|--------------|---------|
| Parameter | N | Mean | Standard | 95% | |
| | | | Deviation | HPD Interval | |
| pDiff | 5000 | -0.0920 | 0.0395 | -0.1750 | -0.0195 |

Figure 28.9 Posterior Density of the Difference in Probabilities



Syntax: BGLIMM Procedure

The following statements are available in the BGLIMM procedure. Items within angle brackets (< >) are optional.

```
PROC BGLIMM < options > ;
  BY variables ;
  CLASS variable < (options) > . . . < variable < (options) > > < / global-options > ;
  MODEL response < (response-options) > = < fixed-effects > < / model-options > ;
  MODEL events / trials = < fixed-effects > < / model-options > ;
  RANDOM random-effects < / options > ;
  REPEATED repeated-effect < / options > ;
  ESTIMATE 'label' estimate-specification < (divisor=n) > < / options > ;
```

The **PROC BGLIMM** statement and the **MODEL** statement are required. The **CLASS** statement must precede the **MODEL** statement. You can have multiple **RANDOM** statements in a program.

The following sections provide a description of each statement.

PROC BGLIMM Statement

```
PROC BGLIMM < options > ;
```

The PROC BGLIMM statement invokes the procedure. [Table 28.1](#) summarizes the available options in the PROC BGLIMM statement by function. The options are then described fully in alphabetical order.

Table 28.1 PROC BGLIMM Statement Options

| Option | Description |
|---------------------------------------------------|----------------------------------------------------------------------|
| Basic Options | |
| DATA= | Specifies the SAS input data set |
| NBI= | Specifies the number of burn-in iterations |
| NMC= | Specifies the number of iterations, excluding the burn-in iterations |
| NTHREADS= | Specifies the number of threads to use |
| OUTPOST= | Names the output data set to contain posterior samples of parameters |
| SAMEBYSEED= | Uses the same seed for each BY group |
| SEED= | Sets the seed for pseudorandom number generation |
| THIN= | Specifies the thinning rate |
| Display Options | |
| NOCLPRINT | Limits or suppresses the display of classification variable levels |
| NOPRINT | Suppresses ODS output |
| Summary, Diagnostics, and Plotting Options | |
| DIAG= | Controls the convergence diagnostics |
| DIC | Computes the deviance information criterion (DIC) |
| PLOTS= | Controls plotting |
| STATS= | Controls posterior statistics |

Table 28.1 *continued*

| Option | Description |
|----------------------|------------------------------------------------------------------|
| Other Options | |
| LOGPOST | Calculates the logarithm of the posterior density and likelihood |
| MISSING= | Indicates how to handle missing values |
| SINGCHOL= | Tunes the singularity criterion for Cholesky decomposition |
| SINGULAR= | Tunes the general singularity criterion |

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

DATA=SAS-data-set

names the input data set for PROC BGLIMM to use. The default is the most recently created data set. Observations in this data set are used to compute the log-likelihood function.

DIAGNOSTICS=NONE | (*keyword-list*)**DIAG=NONE** | (*keyword-list*)

specifies options for convergence diagnostics. By default, PROC BGLIMM computes the effective sample sizes. The sample autocorrelations, Monte Carlo errors, Geweke test, Raftery-Lewis test, and Heidelberger-Welch test are also available. You can request all the diagnostic tests by specifying DIAGNOSTICS=ALL. You can suppress all the diagnostic tests by specifying DIAGNOSTICS=NONE.

You can specify one or more of the following *keyword-list* options:

ALL

computes all diagnostic tests and statistics. You can combine this option with any other specific tests to modify test options. For example, DIAGNOSTICS=(ALL AUTOCORR(LAGS=(1 5 35))) computes all tests by using default settings and autocorrelations at lags 1, 5, and 35.

AUTOCORR < (*autocorrelation-options*) >**AC** < (*autocorrelation-options*) >

computes default autocorrelations at lags 1, 5, 10, and 50 for each variable. You can choose other lags by using the following *autocorrelation-option*:

LAGS=(*numeric-list*)

specifies autocorrelation lags. The *numeric-list* takes only positive integer values.

ESS

computes the effective sample sizes (Kass et al. 1998) of the posterior samples of each parameter. It also computes the correlation time and the efficiency of the chain for each parameter. Small values of ESS might indicate a lack of convergence.

GEWEKE < (*Geweke-options*) >

computes the Geweke spectral density diagnostics; this is a two-sample *t* test between the first f_1 portion (as specified by the FRAC1= option) and the last f_2 portion (as specified by the FRAC2= option) of the chain. By default, FRAC1=0.1 and FRAC2=0.5, but you can choose other fractions by using the following *Geweke-options*:

FRAC1=value**F1=value**

specifies the beginning proportion of the Markov chain. By default, FRAC1=0.1.

FRAC2=value**F2=value**

specifies the end proportion of the Markov chain. By default, FRAC2=0.5.

HEIDELBERGER <(Heidel-options)>**HEIDEL** <(Heidel-options)>

computes the Heidelberg-Welch diagnostic (which consists of a stationarity test and a halfwidth test) for each variable. The stationary diagnostic test tests the null hypothesis that the posterior samples are generated from a stationary process. If the stationarity test is passed, a halfwidth test is then carried out.

You can also specify the following *Heidel-options*, such as **DIAGNOSTICS=HEIDELBERGER(EPS=0.05)**:

EPS=value

specifies a small positive number ϵ such that if the halfwidth is less than ϵ times the sample mean of the retaining iterations, the halfwidth test is passed. By default, EPS=0.1.

HALPHA=value

specifies the α level ($0 < \alpha < 1$) for the halfwidth test. By default, HALPHA=0.05.

SALPHA=value

specifies the α level ($0 < \alpha < 1$) for the stationarity test. By default, SALPHA=0.05.

MAXLAG=number

specifies the maximum number of autocorrelation lags to use to compute the effective sample size. The value of *number* is also used in the calculation of the Monte Carlo standard error. By default, MAXLAG=MIN(500, MCsample/4), where MCsample is the Markov chain sample size that is kept after thinning—that is, $\text{MCsample} = \left\lceil \frac{\text{NMC}}{\text{NTHIN}} \right\rceil$. If *number* is too low, you might observe significant lags, and the effective sample size cannot be calculated accurately. A warning message appears in the SAS log, and you can increase the value of either the MAXLAG= option or the **NMC=** option accordingly. Specifying this option implies the ESS and MCSE options.

MCSE**MCERROR**

computes the Monte Carlo standard error for the posterior samples of each parameter.

NONE

suppresses all the diagnostic tests and statistics. This option is not recommended.

RAFTERY <(Raftery-options)>**RL** <(Raftery-options)>

computes the Raftery-Lewis diagnostic, which evaluates the accuracy of the estimated quantile ($\hat{\theta}_Q$ for a given $Q \in (0, 1)$) of a chain. $\hat{\theta}_Q$ can achieve any degree of accuracy when the chain is allowed to run for a long time. The algorithm stops when the estimated probability $\hat{P}_Q = \Pr(\theta \leq \hat{\theta}_Q)$ reaches within $\pm R$ of the value Q with probability S ; that is, $\Pr(Q - R \leq \hat{P}_Q \leq Q + R) = S$.

You can specify Q , R , S , and a precision level ϵ for a stationarity test by specifying the following *Raftery-options*—for example, `DIAGNOSTICS=RAFTERY(QUANTILE=0.05)`:

ACCURACY=*value*

R=*value*

specifies a small positive number as the margin of error for measuring the accuracy of estimation of the quantile. By default, `ACCURACY=0.005`.

EPS=*value*

specifies the tolerance level (a small positive number) for the stationarity test. By default, `EPS=0.001`.

PROB=*value*

S=*value*

specifies the probability of attaining the accuracy of the estimation of the quantile. By default, `PROB=0.95`.

QUANTILE=*value*

Q=*value*

specifies the order (a value between 0 and 1) of the quantile of interest. By default, `QUANTILE=0.025`.

DIC

computes the deviance information criterion (DIC). DIC is calculated by using the posterior mean estimates of the parameters.

LOGPOST

computes the logarithm of the posterior density of the parameters and the likelihood at each iteration. The `LogLike` and `LogPost` variables are saved in the `OUTPOST=` data set.

MISSING=*keyword*

MISS=*keyword*

specifies how to handle missing values. For more information, see the section “[Missing Data](#)” on page 1266. PROC BGLIMM models missing response variables and discards observations that have missing covariates. You can specify the following *keywords*:

CC

COMPLETECASE

assumes a complete case analysis, so all observations that have missing variable values are discarded before the simulation.

CCMODELY

models the missing response variables and discards observations that have missing covariates.

By default, `MISSING=CCMODELY`.

NBI=*number*

specifies the number of burn-in iterations to perform before saving parameter estimate chains. By default, `NBI=500`.

NMC=number

specifies the number of iterations in the main simulation loop. If you specify a data set in the **OUTPOST=** option, *number* is the number of posterior samples that are saved for each parameter. This is the MCMC sample size if **THIN=1**. By default, NMC=5000.

NOCLPRINT<=number>

suppresses the display of the “Class Level Information” table if you do not specify *number*. If you specify *number*, the values of the classification variables are displayed for only those variables whose number of levels is less than *number*. Specifying *number* helps reduce the size of the “Class Level Information” table if some classification variables have a large number of levels. By default, NOCLPRINT = 200.

NOPRINT

suppresses the generation of ODS output.

NTHREADS=number**NTHREAD=number**

specifies the number of threads (CPUs) on which to run analytic computations and simulations simultaneously. Multithreading is the use of more than one thread to perform computations concurrently. When multithreading is possible, you can realize substantial performance gains compared to the performance that you get from sequential (single-threaded) execution. The more threads there are, the faster the computation runs. But do not specify a *number* greater than the number of CPUs on the host where the analytic computations are performed.

PROC BGLIMM performs two types of threading. In sampling fixed-effects parameters, the procedure allocates data to different threads and accumulates values from each thread; in sampling of random-effects parameters, each thread generates a subset of these parameters simultaneously at each iteration. Most sampling algorithms are threaded. NTHREADS=–1 sets the number of available threads to the number of hyperthreaded cores available on the system. By default, NTHREADS=1.

OUTPOST=SAS-data-set

specifies an output data set to contain the posterior samples of all parameters and the iteration numbers. It contains the log of the posterior density (LOGPOST) and the log likelihood (LOGLIKE) if you specify the **LOGPOST** option. By default, no OUTPOST= data set is created.

PLOTS <(global-plot-option)> <= plot-requests <(option)>>

controls the display of diagnostic plots. You can request three types of plots: trace plots, autocorrelation function plots, and kernel density plots. By default, the plots are displayed in panels unless you specify the *global-plot-option* UNPACK. Also, when you specify more than one type of plot, the plots are grouped by parameter unless you specify the *global-plot-option* GROUPBY=TYPE. When you specify only one *plot-request*, you can omit the parentheses around it, as shown in the following example:

```
plots=none
plots(unpack)=trace
plots=(trace density)
```

If ODS Graphics is enabled and you specify PLOTS=ALL, then PROC BGLIMM produces, for each parameter, a panel that contains the trace plot, the autocorrelation function plot, and the density plot. This is equivalent to specifying PLOTS=(TRACE AUTOCORR DENSITY).

You can specify the following *global-plot-options*:

FRINGE

adds a fringe plot to the horizontal axis of the density plot.

GROUPBY=PARAMETER | TYPE**GROUP=PARAMETER | TYPE**

specifies how the plots are grouped when there is more than one type of plot. By default, GROUPBY=PARAMETER. You can specify the following values:

PARAMETER

groups the plots by parameter.

TYPE

groups the plots by type.

LAGS=number

specifies the number of autocorrelation lags to use in plotting the ACF graph. By default, LAGS=50.

SMOOTH

smooths the trace plot by using a fitted penalized B-spline curve (Eilers and Marx 1996).

UNPACKPANEL**UNPACK**

unpacks all paneled plots so that each plot in a panel is displayed separately.

You can specify the following *plot-requests*:

ALL

requests all types of plots. PLOTS=ALL is equivalent to specifying PLOTS=(TRACE AUTO-CORR DENSITY).

AUTOCORR**ACF**

displays the autocorrelation function plots for the parameters.

DENSITY**D****KERNEL****K**

displays the kernel density plots for the parameters.

NONE

suppresses the display of all plots.

TRACE**T**

displays the trace plots for the parameters.

Consider a model that has four parameters, X1–X4. The following list shows which plots are produced for various option settings:

- PLOTS=(TRACE AUTOCORR) displays the trace and autocorrelation plots for each parameter side by side, with two parameters per panel:

| | | |
|-----------|-----------|--------------|
| Display 1 | Trace(X1) | Autocorr(X1) |
| | Trace(X2) | Autocorr(X2) |

| | | |
|-----------|-----------|--------------|
| Display 2 | Trace(X3) | Autocorr(X3) |
| | Trace(X4) | Autocorr(X4) |

- PLOTS(GROUPBY=TYPE)=(TRACE AUTOCORR) displays all the paneled trace plots, followed by panels of autocorrelation plots:

| | |
|-----------|-----------|
| Display 1 | Trace(X1) |
| | Trace(X2) |

| | |
|-----------|-----------|
| Display 2 | Trace(X3) |
| | Trace(X4) |

| | | |
|-----------|--------------|--------------|
| Display 3 | Autocorr(X1) | Autocorr(X2) |
| | Autocorr(X3) | Autocorr(X4) |

- PLOTS(UNPACK)=(TRACE AUTOCORR) displays a separate trace plot and a separate correlation plot, parameter by parameter:

| | |
|-----------|-----------|
| Display 1 | Trace(X1) |
|-----------|-----------|

| | |
|-----------|--------------|
| Display 2 | Autocorr(X1) |
|-----------|--------------|

| | |
|-----------|-----------|
| Display 3 | Trace(X2) |
|-----------|-----------|

| | |
|-----------|--------------|
| Display 4 | Autocorr(X2) |
|-----------|--------------|

| | |
|-----------|-----------|
| Display 5 | Trace(X3) |
|-----------|-----------|

| | |
|-----------|--------------|
| Display 6 | Autocorr(X3) |
|-----------|--------------|

| | |
|-----------|-----------|
| Display 7 | Trace(X4) |
|-----------|-----------|

| | |
|-----------|--------------|
| Display 8 | Autocorr(X4) |
|-----------|--------------|

- PLOTS(UNPACK GROUPBY=TYPE)=(TRACE AUTOCORR) displays all the separate trace plots, followed by the separate autocorrelation plots:

| | |
|-----------|--------------|
| Display 1 | Trace(X1) |
| Display 2 | Trace(X2) |
| Display 3 | Trace(X3) |
| Display 4 | Trace(X4) |
| Display 5 | Autocorr(X1) |
| Display 6 | Autocorr(X2) |
| Display 7 | Autocorr(X3) |
| Display 8 | Autocorr(X4) |

SAMEBYSEED

uses the same seed that you specify in the **SEED=** option to start the pseudorandom number generator in each **BY** group. If you omit this option, the initial seed for the next **BY** group is the one that is generated at the end of the previous **BY** group.

SEED=number

specifies an integer that is used to start the pseudorandom number generator. If you omit this option or if *number* ≤ 0, the seed is generated from the time of day, which is read from the computer's clock.

SINGCHOL=number

tunes the singularity criterion in Cholesky decomposition and matrix inversion operations. The default is 1E4 times the machine epsilon, or approximately 1E–12 on most computers.

SINGULAR=number

tunes the general singularity criterion applied by the procedure in divisions and inversions. The default is 1E4 times the machine epsilon, or approximately 1E–12 on most computers.

STATISTICS <(global-stats-options)> = **NONE** | **ALL** | *stats-request***STATS** <(global-stats-options)> = **NONE** | **ALL** | *stats-request*

specifies options for posterior statistics. By default, PROC BGLIMM computes the posterior mean, standard deviation, quantiles, and two 95% credible intervals: equal-tail and highest posterior density (HPD). Other available statistics include the posterior correlation and covariance. You can request all the posterior statistics by specifying **STATS=ALL**. You can suppress all the calculations by specifying **STATS=NONE**.

You can specify the following *global-stats-options*:

ALPHA=numeric-list

specifies the α level for the equal-tail and HPD intervals. The value of α must be between 0 and 0.5. By default, **ALPHA=0.05**.

PERCENT=*numeric-list*

PERCENTAGE=*numeric-list*

calculates the posterior percentages. The *numeric-list* contains values between 0 and 100, separated by spaces. By default, PERCENTAGE=(25 50 75).

You can specify the following *stats-requests*:

ALL

computes all posterior statistics. You can combine the ALL option with any other options. For example, STATS(ALPHA=(0.02 0.05 0.1))=ALL computes all statistics by using the default settings and intervals at α levels of 0.02, 0.05, and 0.1.

BRIEF

computes the posterior means, standard deviations, and $100(1 - \alpha)\%$ HPD credible interval for each variable. By default, ALPHA=0.05, but you can use the global *global-stats-option* ALPHA= to specify other values. This is the default output for posterior statistics.

CORR

computes the posterior correlation matrix.

COV

computes the posterior covariance matrix.

INTERVAL

INT

computes the $100(1 - \alpha)\%$ equal-tail and HPD credible intervals for each variable. By default, ALPHA=0.05, but you can use the *global-stats-option* ALPHA= to specify other intervals of any probabilities.

NONE

suppresses all the statistics.

SUMMARY

SUM

computes the posterior means, standard deviations, and percentile points for each variable. By default, the 25th, 50th, and 75th percentile points are produced, but you can use the *global-stats-option* PERCENT= to request specific percentile points.

THIN=*number*

NTHIN=*number*

controls the thinning rate of the simulation. PROC BGLIMM keeps every *n*th simulation sample and discards the rest. All posterior statistics and diagnostics are calculated by using the thinned samples. By default, THIN=1.

BY Statement

BY *variables* ;

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

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

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

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

CLASS Statement

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

The CLASS statement names the classification variables to be used as explanatory variables in the analysis. Response variables do not need to be specified in the CLASS statement.

The CLASS statement must precede the **MODEL** statement. Most options can be specified either as individual variable *options* or as *global-options*. You can specify *options* for each variable by enclosing the options in parentheses after the variable name. You can also specify *global-options* for the CLASS statement by placing them after a slash (/). *Global-options* are applied to all the variables that are specified in the CLASS statement. If you specify more than one CLASS statement, the *global-options* that are specified in any one CLASS statement apply to all CLASS statements. However, individual CLASS variable *options* override the *global-options*. You can specify the following values for either an *option* or a *global-option*:

CPREFIX=*n*

specifies that, at most, the first *n* characters of a CLASS variable name be used in creating names for the corresponding design variables. The default is 32 – min(32, max(2, *f*)), where *f* is the formatted length of the CLASS variable.

DESCENDING

DESC

reverses the sort order of the classification variable. If you specify both the DESCENDING and **ORDER=** options, PROC BGLIMM orders the categories according to the ORDER= option and then reverses that order.

LPREFIX=*n*

specifies that, at most, the first *n* characters of a CLASS variable label be used in creating labels for the corresponding design variables. The default is $256 - \min(256, \max(2, f))$, where *f* is the formatted length of the CLASS variable.

MISSING

treats missing values (., _, .A, ..., .Z for numeric variables and blanks for character variables) as valid values of the CLASS variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sort order for the levels of classification variables. This ordering determines which parameters in the model correspond to each level in the data, so this option can be useful when you use the CONTRAST statement. By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine-dependent. When ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values.

The following table shows how PROC BGLIMM interprets values of the ORDER= option:

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

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

PARAM=keyword

specifies the parameterization method for the classification variable or variables. You can specify any of the *keywords* shown in the following table.

The default is PARAM=GLM. Design matrix columns are created from CLASS variables according to the corresponding coding schemes.

| Value of PARAM= | Coding |
|--------------------------------|-----------------------------------------------------------------------------------------------------|
| EFFECT | Effect coding |
| GLM | Less-than-full-rank reference cell coding (this <i>keyword</i> can be used only in a global option) |
| REFERENCE REF | Reference cell coding |

All parameterizations are full rank, except for the GLM parameterization. The **REF=** option in the CLASS statement determines the reference level for EFFECT and REFERENCE coding and for their orthogonal parameterizations. It also indirectly determines the reference level for a singular GLM parameterization through the order of levels.

If a `PARAM=` option is specified as a variable option for some variables, then any variables for which `PARAM=` is not specified use either the `EFFECT` parameterization if the global `PARAM=` option is not specified, or the full-rank parameterization indicated in the global `PARAM=` option if specified. If the global `PARAM=GLM` option is specified and `PARAM=` is also specified for some variables, GLM parameterization is used for all variables.

REF= *'level' | keyword*

specifies the reference level for `PARAM=EFFECT`, `PARAM=REFERENCE`, and their orthogonalizations. For `PARAM=GLM`, the `REF=` option specifies a level of the classification variable to be put at the end of the list of levels. This level thus corresponds to the reference level in the usual interpretation of the linear estimates with a singular parameterization.

For an individual variable `REF=` option (but not for a global `REF=` option), you can specify the *level* of the variable to use as the reference level. Specify the formatted value of the variable if a format is assigned. For a global or individual variable `REF=` option, you can use one of the following *keywords*:

FIRST designates the first ordered level as reference.

LAST designates the last ordered level as reference.

By default, `REF=LAST`.

TRUNCATE *<=n>*

specifies the length *n* of `CLASS` variable values to use in determining `CLASS` variable levels. The default is to use the full formatted length of the `CLASS` variable. If you specify `TRUNCATE` without the length *n*, the first 16 characters of the formatted values are used. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases before SAS 9. The `TRUNCATE` option is available only as a global option.

ESTIMATE Statement

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

The `ESTIMATE` statement provides a mechanism for computing custom linear combination of the parameters. The basic element of this statement is the *estimate-specification*, which consists of `MODEL` statement effects, random effects, and their coefficients. Specifically, an *estimate-specification* takes the form

$$< \text{fixed-effect values} \dots > < | \text{random-effect values} \dots >$$

You can estimate the linear combination of the parameters $\mathbf{L}'\boldsymbol{\phi}$, where $\mathbf{L}' = (\mathbf{K}'\mathbf{M}')$ and $\boldsymbol{\phi}' = (\boldsymbol{\beta}'\boldsymbol{\gamma}')$. Based on the *estimate-specification* in your `ESTIMATE` statement, PROC BGLIMM constructs the vector $\mathbf{L}' = [\mathbf{K}'\mathbf{M}']$, where \mathbf{K} is associated with the fixed effects and \mathbf{M} is associated with the G-side random effects.

PROC BGLIMM then produces for $\mathbf{L}'\boldsymbol{\phi}$ an estimate (by using the posterior mean), the standard deviation (by using the posterior standard deviation), and the HPD intervals. Results from all `ESTIMATE` statements are combined in the ODS table named Estimates.

The ESTIMATE statement has the following arguments:

| | |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>label</i> | identifies the ESTIMATE statement in the table. A label is required for every ESTIMATE statement that you specify. Labels can be up to 32 characters and must be enclosed in quotation marks. |
| <i>fixed-effect</i> | identifies an effect that appears in the MODEL statement. You can use the keyword INTERCEPT as an effect when you are fitting an intercept in the model. You do not need to include all effects that are specified in the MODEL statement. |
| <i>random-effect</i> | identifies an effect that appears in the RANDOM statement. The first random effect must follow a vertical bar (); however, you are not required to specify random effects. |
| <i>values</i> | are constants that are elements of the L vector that are associated with the fixed and random effects. |

The vector of **L** is specified in order. The **K** component of **L** is specified on the left side of the vertical bars (|). The **M** component of **L** is specified on the right side of the vertical bars. The estimability checking is necessary.

If PROC BGLIMM finds a portion of the specified estimate statement to be nonestimable, then it displays a message in the log.

In the following program, the first ESTIMATE statement compares the first level with the second level for the effect A, and the second ESTIMATE statement compares the first level with the third level for the effect A in a split-plot study where A has three levels and B has two levels:

```
estimate 'A 1 vs 2' A 1 -1 0 A*B .5 .5 -.5 -.5 0 0;
estimate 'A 1 vs 3' A 1 0 -1 A*B .5 .5 0 0 -.5 -.5;
```

Note that no random effects are specified in the preceding statement. The following statements make the same comparison for A when Block and A*Block are random effects:

```
estimate 'A 1 vs 2'
  A      1 -1 0
  A*B    .5 .5 -.5 -.5 0 0 |
  A*Block .25 .25 .25 .25
          -.25 -.25 -.25 -.25
          0 0 0 0 ;
estimate 'A 1 vs 3'
  A      1 0 -1
  A*B    .5 .5 0 0 -.5 -.5 |
  A*Block .25 .25 .25 .25
          0 0 0 0
          -.25 -.25 -.25 -.25 ;
```

The preceding statements do not contain coefficients for B and Block, because they cancel out in estimated differences between levels of A. Coefficients for B and Block are necessary to estimate the mean of one of the levels of A.

Table 28.2 summarizes the *options* available in the ESTIMATE statement after a slash (/).

Table 28.2 ESTIMATE Statement Options

| Option | Description |
|-----------------|-------------------------------------------------------------|
| DIVISOR= | Specifies a value to divide the coefficients |
| E | Prints the L matrix |
| GROUP | Sets up random-effects contrasts between different groups |
| SUBJECT | Sets up random-effects contrasts between different subjects |

DIVISOR=*value*

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

EST

E

displays the **L** matrix coefficients.

GROUP *coeffs*

sets up random-effects contrasts between different groups when you include a **GROUP=** variable in the **RANDOM** statement, as in the following example:

```
estimate 'Trt 1 vs 2 @ x=0.4' trt 1 -1 0 | x 0.4 / group 1 -1;
```

By default, ESTIMATE statement coefficients on random effects are distributed equally across groups.

SUBJECT *coeffs*

sets up random-effects contrasts between different subjects when you include a **SUBJECT=** variable in the **RANDOM** statement. By default, ESTIMATE statement coefficients on random effects are distributed equally across subjects.

MODEL Statement

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

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

The MODEL statement, which is required, names the dependent variable and the fixed effects. The *fixed-effects* determine the **X** matrix of the model (see the section “[Notation for the Generalized Linear Mixed Model](#)” for details). You specify effects in the same way as in other SAS procedures.

An intercept is included in the fixed-effects model by default. You can remove it by using the **NOINT** option.

You can specify the dependent variable by using either the *response* syntax or the *events/trials* syntax. The *events/trials* syntax is specific to models for binomial data. A binomial(n, π) variable is the sum of n independent Bernoulli trials with event probability π . Each Bernoulli trial results in either an event or a

nonevent (with probability $1 - \pi$). You use the *events/trials* syntax to indicate to the BGLIMM 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, PROC BGLIMM defaults to the binomial distribution. The response is then the *events* variable. The *trials* variable is accounted for in model fitting as an additional weight. If you use the *response* syntax, the procedure defaults to the normal distribution.

The MODEL statement uses two sets of options. The *response-options* determine how PROC BGLIMM models probabilities for binary data. The *model-options* control other aspects of model formation and inference. Table 28.3 summarizes these options, and subsequent sections describe them in detail.

Table 28.3 MODEL Statement Options

| Option | Description |
|----------------------------------|----------------------------------------------------------------------------------------|
| Response Variable Options | |
| DESCENDING | Reverses the order of response categories |
| EVENT= | Specifies the event category in binary models |
| ORDER= | Specifies the sort order for the response variable |
| REF= | Specifies the reference category in generalized logit models |
| Model Options | |
| COEFPRIOR= | Specifies the prior of the fixed-effects coefficients |
| DIST= | Specifies the response distribution |
| INIT= | Controls the generation of initial values of the fixed-effects coefficients |
| LINK= | Specifies the link function |
| NOINT | Excludes the fixed-effects intercept from the model |
| NOOUTPOST | Suppresses storing the posterior samples of missing responses in the OUTPOST= data set |
| OFFSET= | Specifies the offset variable |
| SCALEPRIOR= | Specifies the prior of the scale parameter |

Response Variable Options

Response variable options determine how the BGLIMM procedure models probabilities for binary data.

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

DESCENDING

DESC

reverses the order of the response categories. If you specify both the DESCENDING and ORDER= options, PROC BGLIMM orders the response categories according to the ORDER= option and then reverses that order.

EVENT='category' | FIRST | LAST

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

You can specify any of the following values:

'category'

specifies that observations whose value matches *category* (formatted, if a format is applied) in quotation marks represent events in the data. For example, the following statements specify that observations that have a formatted value of '1' represent events in the data. The probability that is modeled by PROC BGLIMM is thus the probability that the variable *def* takes the (formatted) value '1'.

```
proc bglimm data=MyData;
  class A B C;
  model Def(event = '1') = A B C;
run;
```

FIRST

designates the first ordered category as the event.

LAST

designates the last ordered category as the event.

By default, EVENT=FIRST.

ORDER=FORMATTED | FREQ | INTERNAL

specifies the sort order of 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 BGLIMM run or in the DATA step that created the data set), the levels are ordered by their internal (numeric) value. [Table 28.4](#) shows the interpretation of the ORDER= option.

Table 28.4 Sort Order

| ORDER= | Levels Sorted By |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| FORMATTED | External formatted value, except for numeric variables that have no explicit format, which are sorted by their unformatted (internal) value |
| FREQ | Descending frequency count (levels that have the most observations come first in the order) |
| INTERNAL | Unformatted value |

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

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

REFERENCE='category' | FIRST | LAST

REF='category' | FIRST | LAST

specifies 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 any of the following values:

'category'

specifies that observations whose value matches *category* (formatted, if a format is applied) are designated as the reference.

FIRST

designates the first ordered category as the reference.

LAST

designates the last ordered category as the reference.

By default, REF=LAST.

Model Options

You can specify the following *model-options* in the MODEL statement after a slash (/):

COEFFPRIOR=NORMAL <(options)> | CONSTANT

CPRIOR=NORMAL <(options)> | CONSTANT

specifies the prior distribution for the fixed-effects coefficients. The default is COEFFPRIOR=CONSTANT, which specifies the noninformative and improper prior of a constant. If you specify COEFFPRIOR=NORMAL, it is $N(0, 10^4 \mathbf{I})$, where \mathbf{I} is the identity matrix. You can specify the following *options* for the normal prior, enclosed in parentheses:

INPUT=SAS-data-set

specifies a SAS data set that contains the mean and covariance information of the normal prior. The data set must have a `_TYPE_` variable to represent the type of each observation and a variable for each regression coefficient. If the data set also contains a `_NAME_` variable, the values of this variable are used to identify the covariance for the `_TYPE_='COV'` observations; otherwise, the `_TYPE_='COV'` observations are assumed to be in the same order as the explanatory variables in the MODEL statement. PROC BGLIMM reads the mean vector from the observation for which `_TYPE_='MEAN'` and reads the covariance matrix from observations for which `_TYPE_='COV'`. For an independent normal prior, the variances can be specified using `_TYPE_='VAR'`; alternatively, the precision (inverse of the variances) can be specified using `_TYPE_='PRECISION'`.

VAR <=c>

specifies the normal prior $N(0, c\mathbf{I})$, where \mathbf{I} is the identity matrix and c is a scalar.

DISTRIBUTION=*keyword***DIST=***keyword***ERROR=***keyword***ERR=***keyword*

specifies the response distribution for the model. The *keywords* and their associated distributions are shown in [Table 28.5](#).

Table 28.5 Built-In Distribution Functions

| DISTRIBUTION= | Distribution Function |
|----------------------------------|------------------------------|
| BINARY | Binary |
| BINOMIAL | Binary or binomial |
| EXPONENTIAL EXPO | Exponential |
| GAMMA GAM | Gamma |
| GEOMETRIC GEOM | Geometric |
| INVGAUSS IG | Inverse Gaussian |
| NEGBINOMIAL NEGBIN NB | Negative binomial |
| NORMAL GAUSSIAN GAUSS | Normal |
| POISSON POI | Poisson |

If you do not specify a distribution, PROC BGLIMM defaults to the normal distribution for continuous response variables and to the binomial distribution for classification or character variables, unless you use the *events/trial* syntax in the MODEL statement. If you choose the *events/trial* syntax, the procedure defaults to the binomial distribution.

If you do not specify a link function in the **LINK=** option, a default link function is used. The default link function for each distribution and other commonly used link functions are shown in [Table 28.6](#). You can also use the **LINK=** option to specify any link function shown in [Table 28.7](#).

Table 28.6 Default and Commonly Used Link Functions

| DISTRIBUTION= | Default Link Function | Other Commonly Used Link Functions |
|----------------------------------|------------------------------|-------------------------------------------|
| BINARY | Logit | Probit, complementary log-log, log-log |
| BINOMIAL | Logit | Probit, complementary log-log, log-log |
| EXPONENTIAL EXPO | Log | Reciprocal |
| GAMMA GAM | Log | Reciprocal |
| GEOMETRIC GEOM | Log | |
| INVGAUSS IG | Reciprocal square | |
| NEGBINOMIAL NEGBIN NB | Log | |
| NORMAL GAUSSIAN GAUSS | Identity | Log |
| POISSON POI | Log | |

INIT=*keyword-list* | (*numeric-list*)

INITIAL=*keyword-list* | (*numeric-list*)

specifies options for generating the initial values for the coefficients parameters that you specify as *fixed-effects* in the MODEL statement. You can specify the following *keywords*:

LIST=*numeric-list*

assigns the numbers to use as the initial values of the fixed effects in the corresponding list order, including the intercept. The length of the *numeric-list* must be the same as the number of fixed effects. For example, the following statement assigns the values 1, 2, and 3 to the first, second, and third coefficients in the model and prints the table of initial values:

```
model y = x / init=(list=(1 2 3) pinit);
```

If the number of items in the *numeric-list* is less than the number of fixed effects, the initial value of each remaining parameter is replaced by the corresponding default initial value. For example, the corresponding mode of the posterior density is used. If the number of items in the *numeric-list* is greater than the number of fixed effects, the extra numbers are ignored.

PINIT

tabulates initial values for the fixed effects. (By default, PROC BGLIMM does not display the initial values.)

POSTMODE

uses the mode of the posterior density as the initial value of the parameter.

PRIORMODE

uses the mode of the prior density as the initial value of the parameter.

By default, INIT=POSTMODE.

LINK=*keyword*

specifies the link function for the model. The *keywords* and the associated link functions are shown in Table 28.7. Default and commonly used link functions for the available distributions are shown in Table 28.6.

Table 28.7 Built-In Link Functions

| LINK= | Link Function | $g(\mu) = \eta =$ |
|------------------------------------|------------------------|------------------------|
| CLOGLOG CLL | Complementary log-log | $\log(-\log(1 - \mu))$ |
| IDENTITY ID | Identity | μ |
| INVERSE RECIPROCAL | Reciprocal | $1/\mu$ |
| LOG | Logarithm | $\log(\mu)$ |
| LOGIT | Logit | $\log(\mu/(1 - \mu))$ |
| LOGLOG | Log-log | $-\log(-\log(\mu))$ |
| POWERMINUS2 | Power with exponent -2 | $1/\mu^2$ |
| PROBIT | Probit | $\Phi^{-1}(\mu)$ |

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

NOINT

excludes the intercept from the fixed-effects model. An intercept is included by default.

NOOUTPOST

suppresses storing the posterior samples of missing responses in the **OUTPOST=** data set. By default, PROC BGLIMM outputs the posterior samples of all missing responses to the **OUTPOST=** data set.

OFFSET=*variable*

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

SCALEPRIOR=GAMMA<(options)> | **IGAMMA**<(options)> | **IMPROPER****SPRIOR=GAMMA**<(options)> | **IGAMMA**<(options)> | **IMPROPER**

specifies the prior distribution for the scale parameter, if there is a scale parameter in the model. For models that do not have a dispersion parameter (the Poisson and binomial), this option is ignored.

A gamma prior $G(a, b)$ with density $f(t) = \frac{b(bt)^{a-1}e^{-bt}}{\Gamma(a)}$ is specified by **SCALEPRIOR=GAMMA**, which can be followed by one of the following gamma *options* enclosed in parentheses. The hyperparameters a and b are the shape and inverse-scale parameters of the gamma distribution, respectively. The default is $G(10^{-4}, 10^{-4})$.

SHAPE= a **ISCALE=** b

when both are specified, results in a $G(a, b)$ prior.

SHAPE= c

when specified alone, results in a $G(c, c)$ prior.

ISCALE= c

when specified alone, results in a $G(c, c)$ prior.

An inverse gamma prior $IG(a, b)$ with density $f(t) = \frac{b^a}{\Gamma(a)} t^{-(a+1)} e^{-b/t}$ is specified by **SCALEPRIOR=IGAMMA**, which can be followed by one of the following inverse gamma *options* enclosed in parentheses. The hyperparameters a and b are the shape and scale parameters of the inverse gamma distribution, respectively. The default is $IG(2, 2)$.

SHAPE= a **SCALE=** b

when both are specified, results in a $IG(a, b)$ prior.

SHAPE= c

when specified alone, results in an $IG(c, c)$ prior.

SCALE= c

when specified alone, results in an $IG(c, c)$ prior.

An improper prior with density $f(t)$ proportional to t^{-1} is specified by **SCALEPRIOR=IMPROPER**.

For the normal distribution, the prior for the scale parameter is always an inverse gamma, and the default is $IG(2, 2)$. Other prior specification is ignored if the prior is not an inverse gamma.

For other distributions, you can choose from a gamma prior, an inverse gamma prior, or an improper prior. The default is the improper prior.

RANDOM Statement

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

Using notation from the section “[Notation for the Generalized Linear Mixed Model](#)” on page 1207, the RANDOM statement defines the **Z** matrix of the mixed model, the random effects in the **y** vector, and the covariance structure of **G**.

The **Z** matrix is constructed exactly like the **X** matrix for the fixed effects, and the **G** matrix is constructed to correspond to the effects that constitute **Z**. The covariance structure of **G** is defined by using the **TYPE=** option. The random effects can be classification or continuous effects, and you can specify multiple RANDOM statements.

You can specify INTERCEPT (or INT) as a random effect to indicate the intercept. PROC BGLIMM does not include the intercept in the RANDOM statement by default as it does in the [MODEL](#) statement.

[Table 28.8](#) summarizes the *options* available in the RANDOM statement. All *options* are then discussed in alphabetical order.

Table 28.8 RANDOM Statement Options

| Option | Description |
|----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| Construction and Sampling of Covariance Structure | |
| COVPRIOR= | Specifies the prior for the G matrix |
| GROUP= | Varies the G covariance matrix by group |
| NUTS | Specifies options for the No-U-Turn Sampler of the Hamiltonian Monte Carlo algorithm for the G matrix |
| RESIDUAL | Designates a covariance structure as R-side |
| SUBJECT= | Identifies the subjects in the model |
| TYPE= | Specifies the type of the G matrix |
| Statistical Output | |
| G | Displays the estimated G matrix |
| GCORR | Displays the correlation matrix that corresponds to the estimated G matrix |
| MONITOR | Displays the summary, diagnostic statistics, and plots of individual random effects |
| NOOUTPOST | Suppresses storing the posterior samples of individual random effects in the OUTPOST= data set |

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

COVPRIOR=keyword

specifies a prior for the covariance matrix, **G**, of the random effects, when the **G** matrix is of the UN, UN(1), VC, or TOEP(1) type, where a conjugate sampler is used to sample the covariance matrix. This option is ignored for other covariance types, because you assign a flat prior to the **G** matrix if it is of a type other than UN, UN(1), VC, or TOEP(1). For more information, see the section “[Prior for the G-Side Covariance](#)” on page 1262.

You can specify the following priors:

COVPRIOR=IGAMMA <(SHAPE=a, SCALE=b)>

specifies an inverse gamma prior, $\text{IGamma}(a, b)$, with density $f(t) = \frac{b^a}{\Gamma(a)} t^{-(a+1)} e^{-b/t}$ for each diagonal term of the **G** matrix. It is the default prior for the covariance types UN(1), VC, and TOEP.

You can specify the following parameters for an inverse gamma distribution, enclosed in parentheses:

SHAPE=a

specifies the shape parameter for the inverse gamma distribution. By default, SHAPE=2.

SCALE=b

specifies the scale parameter of the inverse gamma distribution. BY default, SCALE=2.

COVPRIOR=IWISHART <(DF=a, SCALE=b)>**COVPRIOR=IWISH <(DF=a, SCALE=b)>****COVPRIOR=IW <(DF=a, SCALE=b)>**

specifies an inverse Wishart prior, $\text{IWISHART}(a, b)$, for the **G** matrix of the random effects. It is the default prior for the UN covariance type.

You can specify the following parameters for an inverse Wishart distribution, enclosed in parentheses:

DF=a

specifies the degrees of freedom of the inverse Wishart distribution. The default is the dimension of the **G** matrix of the random effects plus 3.

SCALE=b

specifies $b\mathbf{I}$ for the scale parameter of the inverse Wishart distribution, where **I** is the identity matrix. The default is the dimension of the **G** matrix of the random effects plus 3.

COVPRIOR=HALFCAUCHY <(SCALE=a)>**COVPRIOR=HCAUCHY <(SCALE=a)>****COVPRIOR=HC <(SCALE=a)>**

specifies the prior to be a half-Cauchy distribution. The half-Cauchy prior is applied only to the diagonal terms (variances) of the **G** matrix. The off-diagonal terms of the **G** matrix are assumed to have a flat prior.

You can specify the scale parameter of the half-Cauchy distribution, enclosed in parentheses. The scale parameter has to be positive. The default value is 25.

COVPRIOR=HALFNORMAL <(VAR=*a*)>

COVPRIOR=HNORMAL <(VAR=*a*)>

COVPRIOR=HN <(VAR=*a*)>

specifies the prior to be a half-normal distribution. The half-normal is applied only to the diagonal terms (variances) of the **G** matrix. The off-diagonal terms of the **G** matrix are assumed to have a flat prior.

You can specify the variance of the half-normal distribution, enclosed in parentheses. The variance must be positive. The default value is 25.

COVPRIOR=SIWISHART <(DF=*a*, SCALE=*b*, VAR=*c*)>

COVPRIOR=SIWISH <(DF=*a*, SCALE=*b*, VAR=*c*)>

COVPRIOR=SIW <(DF=*a*, SCALE=*b*, VAR=*c*)>

specifies a scaled inverse Wishart prior for the **G** matrix of the random effects.

You can specify the following parameters for an scaled inverse Wishart distribution, enclosed in parentheses:

DF=*a*

specifies the degrees of freedom of the scaled inverse Wishart distribution. The default is the dimension of the **G** matrix of the random effects plus 3.

SCALE=*b*

specifies $b\mathbf{I}$ for the scale parameter of the scaled inverse Wishart distribution, where **I** is the identity matrix. The default is the dimension of the **G** matrix of the random effects plus 3.

VAR=*c*

specifies a value for the variance parameter of the normal prior for $\log(\delta_i)$. By default, VAR=1.

COVPRIOR=UNIFORM <(LOWER=*a*, UPPER=*b*)>

COVPRIOR=UNIF <(LOWER=*a*, UPPER=*b*)>

specifies a uniform prior, **UNIFORM**(*a*,*b*), for the **G** matrix of the random effects. The uniform prior is applied to standard deviations (the square root of the diagonal terms) of the **G** matrix.

You can specify the lower and upper bounds of the uniform distribution as follows, enclosed in parentheses:

LOWER=*a*

specifies the lower bound of the uniform distribution. The lower bound must be nonnegative. By default, LOWER=0.

UPPER=*b*

specifies the upper bound of the uniform distribution. The lower bound must be positive. By default, UPPER=1E10.

G

displays the estimated **G** matrix for G-side random effects that are associated with this **RANDOM** statement. PROC BGLIMM displays blanks for values that are 0. The ODS table name is **G**.

GCORR

displays the correlation matrix that corresponds to the estimated **G** matrix for G-side random effects that are associated with this RANDOM statement. PROC BGLIMM displays blanks for values that are 0. The ODS table name is GCORR.

GROUP=effect**GRP=effect**

identifies groups by which to vary the covariance parameters. All observations that have the same level of the *effect* have the same covariance parameters. Each new level of the grouping effect produces a new set of covariance parameters. You should exercise caution in properly defining the *effect*, because strange covariance patterns can result when it is misused. Also, the *effect* can greatly increase the number of estimated covariance parameters, which can adversely affect the sampling process.

The GROUP= *effect* must be specified in the CLASS statement.

MONITOR | SOLUTION | S

MONITOR=(numeric-list) | SOLUTION=(numeric-list) | S=(numeric-list)

MONITOR=RANDOM (number) | SOLUTION=RANDOM (number) | S=RANDOM (number)

displays results (summary, diagnostic statistics, and plots) for the individual-level random-effects parameters. By default, to save time and space, PROC BGLIMM does not print results for individual-level random-effects parameters. In models that have a large number of individual random effects (for example, tens of thousands), it can take a long time to display the summary, diagnostic statistics, and plots for all the individual-level parameters, so be cautious when using this option.

You can monitor a subset of the random-effects parameters. You can provide a numeric list of the SUBJECT indexes, or PROC BGLIMM can randomly choose a subset of all subjects for you.

To monitor a list of random-effects parameters for certain subjects, you can provide their indexes as follows:

```
random x / subject=index monitor=(1 to 5 by 2 23 57);
```

In this case, PROC BGLIMM outputs results of random effects for subjects 1, 3, 5, 23, and 57. If the number of items in the list is greater than the number of subjects, the extra list items are ignored.

PROC BGLIMM can also randomly choose a subset of all the subjects to monitor, if you submit a statement such as the following:

```
random x / subject=index monitor=(random(12));
```

In this case, PROC BGLIMM outputs results of random effects for 12 randomly selected subjects. You control the sequence of the random indexes by specifying the **SEED=** option in the PROC BGLIMM statement.

When you specify the MONITOR option, it uses the values that you specify in the **STATISTICS=** and **PLOTS=** options in the PROC BGLIMM statement.

NOOUTPOST

suppresses storing the posterior samples of individual random-effects parameters in the **OUTPOST=** data set. By default, PROC BGLIMM outputs the posterior samples of all random-effects parameters to the **OUTPOST=** data set. You can use this option to avoid saving the random-effects parameters. In models that have a large number of individual random effects (for example, tens of thousands), PROC BGLIMM can run faster if it does not save the posterior samples of all the individual random effects.

When you specify both the NOOUTPOST option and the **MONITOR** option, PROC BGLIMM outputs the list of variables that are monitored.

There is a limit on the maximum number of variables that you can save to an **OUTPOST=** data set. If you run a large-scale random-effects model in which the number of parameters exceeds that limit, the NOOUTPOST option is invoked automatically and PROC BGLIMM does not save the individual random-effects draws to the output data set. You can use the **MONITOR** option to select a subset of the parameters to store in the **OUTPOST=** data set.

NUTS<(nuts-options)>

specifies options for the No-U-Turn Sampler (NUTS) of the Hamiltonian algorithm that is used to sample the parameters in the **G** matrix of the corresponding **RANDOM** statement. The NUTS algorithm is a version of the adaptive Hamiltonian Monte Carlo algorithm with automatic tuning of the step size and number of steps in each iteration. For more information, see the section “[Hamiltonian Monte Carlo Sampler](#)” on page 1260. You can specify the following *nuts-options*:

DELTA=value

specifies the target acceptance rate during the tuning process of the NUTS algorithm. Increasing the *value* can often improve mixing, but it can also significantly slow down the sampling. By default, DELTA=0.6.

MAXHEIGHT=value

specifies the maximum height of the NUTS algorithm tree. The taller the tree, the more gradient evaluations per iteration the procedure calculates. The number of evaluations is 2^{height} . Usually, the height of a tree should be no more than 10 during the sampling stage, but it can go higher during the tuning stage. A larger number indicates that the algorithm is having difficulty converging. By default, MAXHEIGHT=10.

NTU=value

specifies the number of tuning iterations for the NUTS algorithm to use. By default, NTU=1000.

STEPSIZE=value

specifies the initial step size in the NUTS algorithm. By default, STEPSIZE=0.1.

RESIDUAL**RSIDE**

specifies that the random effects listed in this statement be R-side effects. You use this option in the **RANDOM** statement if the covariance matrix is for the R-side. Specifying this option is equivalent to using the **REPEATED** statement. For example, if it is necessary to order the columns of the R-side AR(1) covariance structure by the Time variable, you can use the **RESIDUAL** option as in the following statements:


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

SUBJECT=*effect*

SUB=*effect*

identifies the subjects in the model for the random effects.

A set of random effects is estimated for each subject. All variables in the *effect* must be declared as categorical variables in the **CLASS** statement. PROC BGLIMM assumes independence across subjects, conditional on other parameters in the model. Specifying a subject effect is equivalent to nesting all other effects in the **RANDOM** statement within the subject effect. Thus, for the **RANDOM** statement, the **SUBJECT=** option produces a block-diagonal structure that has identical blocks.

For more information about specifying a random effect with or without the **SUBJECT=** variable, see the section “[Treatment of Subjects in the RANDOM Statement](#)” on page 1264.

TYPE=*covariance-structure*

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

Although a variety of structures are available, many applications call for either simple diagonal (**TYPE=VC**) or unstructured covariance matrices. The default structure is **TYPE=VC** (variance components), which models a different variance component for each random effect.

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* and their descriptions are provided in [Table 28.9](#).

Table 28.9 Covariance Structures

| Structure | Description | Parms | (i, j) Element |
|------------------|---------------------------------|---------------------------|---------------------------------------------------------------------|
| ANTE(1) | Antedependence | $2t - 1$ | $\sigma_i \sigma_j \prod_{k=i}^{j-1} \rho_k$ |
| AR(1) | Autoregressive(1) | 2 | $\sigma^2 \rho^{ i-j }$ |
| ARH(1) | Heterogeneous AR(1) | $t + 1$ | $\sigma_i \sigma_j \rho^{ i-j }$ |
| ARMA(1,1) | ARMA(1,1) | 3 | $\sigma^2 [\gamma \rho^{ i-j -1} 1(i \neq j) + 1(i = j)]$ |
| CS | Compound symmetry | 2 | $\sigma_1 + \sigma^2 1(i = j)$ |
| CSH | Heterogeneous compound symmetry | $t + 1$ | $\sigma_i \sigma_j [\rho 1(i \neq j) + 1(i = j)]$ |
| FA(1) | Factor analytic | $2t$ | $\sum_{k=1}^{\min(i,j,1)} \lambda_{ik} \lambda_{jk} + d_i 1(i = j)$ |
| HF | Huynh-Feldt | $t + 1$ | $(\sigma_i^2 + \sigma_j^2)/2 - \lambda 1(i \neq j)$ |
| TOEP | Toeplitz | t | $\sigma_{ i-j +1}$ |
| TOEP(q) | Banded Toeplitz | q | $\sigma_{ i-j +1} 1(i-j < q)$ |
| TOEPH | Heterogeneous Toeplitz | $2t - 1$ | $\sigma_i \sigma_j \rho_{ i-j }$ |
| TOEPH(q) | Banded heterogeneous Toeplitz | $t + q - 1$ | $\sigma_i \sigma_j \rho_{ i-j } 1(i-j < q)$ |
| UN | Unstructured | $t(t + 1)/2$ | σ_{ij} |
| UN(q) | Banded | $\frac{q}{2}(2t - q + 1)$ | $\sigma_{ij} 1(i-j < q)$ |

Table 28.9 *continued*

| Structure | Description | Parms | (<i>i, j</i>) Element |
|-----------|---------------------|----------|-------------------------------------------------------------------------|
| VC | Variance components | <i>q</i> | $\sigma_k^2 1(i = j)$ and <i>i</i> corresponds to <i>k</i> th effect |

In Table 28.9, Parms refers to the number of covariance parameters in the structure, *t* is the overall dimension of the covariance matrix, *q* is the order parameter, and $1(A)$ equals 1 when *A* is true and 0 otherwise. For example, $1(i = j)$ equals 1 when *i* = *j* and 0 otherwise, and $1(|i - j| < q)$ equals 1 when $|i - j| < q$ and 0 otherwise. For the TYPE=TOEPH structures, $\rho_0 = 1$.

ANTE(1)

specifies a first-order antedependence structure (Kenward 1987; Patel 1991) parameterized in terms of variances and correlation parameters. If *t* ordered random variables ξ_1, \dots, ξ_t have a first-order antedependence structure, then each $\xi_j, j > 1$, is independent of all other $\xi_k, k < j$, given ξ_{j-1} . This Markovian structure is characterized by its inverse variance matrix, which is tridiagonal. Parameterizing an ANTE(1) structure for a random vector of size *t* requires $2t - 1$ parameters: variances $\sigma_1^2, \dots, \sigma_t^2$ and $t - 1$ correlation parameters $\rho_1, \dots, \rho_{t-1}$. The covariances among random variables ξ_i and ξ_j are then constructed as

$$\text{Cov}[\xi_i, \xi_j] = \sqrt{\sigma_i^2 \sigma_j^2} \prod_{k=i}^{j-1} \rho_k$$

PROC BGLIMM constrains the correlation parameters to satisfy $|\rho_k| < 1, \forall k$. For variable-order antedependence models see Macchiavelli and Arnold (1994).

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, respectively, and are not necessarily the observation numbers. For example, in the following statements, the values correspond to the class levels for the Time effect of the *i*th and *j*th observation within a particular subject:

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

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

ARH(1)

specifies a heterogeneous first-order autoregressive structure,

$$\text{Cov}[\xi_i, \xi_j] = \sqrt{\sigma_i^2 \sigma_j^2} \rho^{|i^* - j^*|}$$

where $|\rho| < 1$. This covariance structure has the same correlation pattern as the TYPE=AR(1) structure, but the variances are allowed to differ.

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, ρ is the autoregressive parameter, γ models a moving average component, and σ^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 28.10](#) 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 BGLIMM imposes the constraints $|\rho| < 1$ and $|\gamma| < 1$ for stationarity, although for some values of ρ and γ in this region, the resulting covariance matrix is not positive definite.

CS

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

$$\text{Cov} [\xi_i, \xi_j] = \begin{cases} \sigma_1 + \sigma^2 & i = j \\ \sigma_1 & i \neq j \end{cases}$$

The compound symmetry structure arises naturally with nested random effects, such as when subsampling error is nested within experimental error. 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 in the subunit, leading to constant correlations between the observations. Compound symmetry structures are thus usually not appropriate for processes in which correlations decline according to some metric, such as spatial and temporal processes.

CSH

specifies the heterogeneous compound symmetry structure, which is an equicorrelation structure but allows for different variances,

$$\text{Cov} [\xi_i, \xi_j] = \begin{cases} \sqrt{\sigma_i^2 \sigma_j^2} & i = j \\ \rho \sqrt{\sigma_i^2 \sigma_j^2} & i \neq j \end{cases}$$

FA(1)

specifies the factor-analytic structure with one factor (Jennrich and Schluchter 1986). This structure is of the form $\mathbf{\Lambda}\mathbf{\Lambda}' + \mathbf{D}$, where $\mathbf{\Lambda}$ is a $t \times 1$ vector and \mathbf{D} is a $t \times t$ diagonal matrix with t different parameters.

HF

specifies a covariance structure that satisfies the general Huynh-Feldt condition (Huynh and Feldt 1970). For a random vector that has t elements, this structure has $t + 1$ positive parameters and covariances

$$\text{Cov} [\xi_i, \xi_j] = \begin{cases} \sigma_i^2 & i = j \\ 0.5(\sigma_i^2 + \sigma_j^2) - \lambda & i \neq j \end{cases}$$

A covariance matrix Σ generally satisfies the Huynh-Feldt condition if it can be written as $\Sigma = \tau \mathbf{1}' + \mathbf{1} \tau' + \lambda \mathbf{I}$. The preceding parameterization chooses $\tau_i = 0.5(\sigma_i^2 - \lambda)$. Several simpler covariance structures give rise to covariance matrices that also satisfy the Huynh-Feldt condition. For example, TYPE=CS, VC, and UN(1) are nested within TYPE=HF. Note also that the HF structure is nested within an unstructured covariance matrix.

TOEP

models a Toeplitz covariance structure. This structure can be viewed as an autoregressive structure whose order is equal to the dimension of the matrix,

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

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 whose order is 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.

TOEPH<(q)>

models a Toeplitz covariance structure. The correlations of this structure are banded as in the TOEP or TOEP(q) structures, but the variances are allowed to vary:

$$\text{Cov} [\xi_i, \xi_j] = \begin{cases} \sigma_i^2 & i = j \\ \rho_{|i-j|} \sqrt{\sigma_i^2 \sigma_j^2} & i \neq j \end{cases}$$

The correlation parameters satisfy $|\rho_{|i-j|}| < 1$. If you specify the optional value q , the correlation parameters with $|i-j| \geq q$ are set to 0, creating a banded correlation structure. The specification TYPE=TOEPH(1) results in a diagonal covariance matrix with heterogeneous variances.

UN<(q)>

specifies a completely general (unstructured) covariance matrix that is 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 constrained to be nonnegative definite. If you specify the order parameter q , then PROC BGLIMM estimates only the first q bands of the matrix, setting elements in all higher bands equal to 0.

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 28.10 lists some examples of the structures in Table 28.9.

Table 28.10 Covariance Structure Examples

| Description | Structure | Example |
|-------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| First-order antedependence | ANTE(1) | $\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_1 & \sigma_1\sigma_3\rho_1\rho_2 \\ \sigma_2\sigma_1\rho_1 & \sigma_2^2 & \sigma_2\sigma_3\rho_2 \\ \sigma_3\sigma_1\rho_2\rho_1 & \sigma_3\sigma_2\rho_2 & \sigma_3^2 \end{bmatrix}$ |
| First-order autoregressive | AR(1) | $\sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$ |
| Heterogeneous AR(1) | ARH(1) | $\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho & \sigma_1\sigma_3\rho^2 & \sigma_1\sigma_4\rho^3 \\ \sigma_2\sigma_1\rho & \sigma_2^2 & \sigma_2\sigma_3\rho & \sigma_2\sigma_4\rho^2 \\ \sigma_3\sigma_1\rho^2 & \sigma_3\sigma_2\rho & \sigma_3^2 & \sigma_3\sigma_4\rho \\ \sigma_4\sigma_1\rho^3 & \sigma_4\sigma_2\rho & \sigma_4\sigma_3\rho & \sigma_4^2 \end{bmatrix}$ |
| First-order autoregressive moving average | ARMA(1,1) | $\sigma^2 \begin{bmatrix} 1 & \gamma & \gamma\rho & \gamma\rho^2 \\ \gamma & 1 & \gamma & \gamma\rho \\ \gamma\rho & \gamma & 1 & \gamma \\ \gamma\rho^2 & \gamma\rho & \gamma & 1 \end{bmatrix}$ |
| Compound symmetry | CS | $\begin{bmatrix} \sigma_1 + \sigma^2 & \sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 + \sigma^2 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma_1 + \sigma^2 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma_1 & \sigma_1 + \sigma^2 \end{bmatrix}$ |
| Heterogeneous compound symmetry | CSH | $\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho & \sigma_1\sigma_3\rho & \sigma_1\sigma_4\rho \\ \sigma_2\sigma_1\rho & \sigma_2^2 & \sigma_2\sigma_3\rho & \sigma_2\sigma_4\rho \\ \sigma_3\sigma_1\rho & \sigma_3\sigma_2\rho & \sigma_3^2 & \sigma_3\sigma_4\rho \\ \sigma_4\sigma_1\rho & \sigma_4\sigma_2\rho & \sigma_4\sigma_3\rho & \sigma_4^2 \end{bmatrix}$ |
| First-order factor analytic | FA(1) | $\begin{bmatrix} \lambda_1^2 + d_1 & \lambda_1\lambda_2 & \lambda_1\lambda_3 & \lambda_1\lambda_4 \\ \lambda_2\lambda_1 & \lambda_2^2 + d_2 & \lambda_2\lambda_3 & \lambda_2\lambda_4 \\ \lambda_3\lambda_1 & \lambda_3\lambda_2 & \lambda_3^2 + d_3 & \lambda_3\lambda_4 \\ \lambda_4\lambda_1 & \lambda_4\lambda_2 & \lambda_4\lambda_3 & \lambda_4^2 + d_4 \end{bmatrix}$ |
| Huynh-Feldt | HF | $\begin{bmatrix} \sigma_1^2 & \frac{\sigma_1^2 + \sigma_2^2}{2} - \lambda & \frac{\sigma_1^2 + \sigma_3^2}{2} - \lambda \\ \frac{\sigma_2^2 + \sigma_1^2}{2} - \lambda & \sigma_2^2 & \frac{\sigma_2^2 + \sigma_3^2}{2} - \lambda \\ \frac{\sigma_3^2 + \sigma_1^2}{2} - \lambda & \frac{\sigma_3^2 + \sigma_2^2}{2} - \lambda & \sigma_3^2 \end{bmatrix}$ |

Table 28.10 continued

| Description | Structure | Example |
|-------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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) | $\begin{bmatrix} \sigma^2 & \sigma_1 & 0 & 0 \\ \sigma_1 & \sigma^2 & \sigma_1 & 0 \\ 0 & \sigma_1 & \sigma^2 & \sigma_1 \\ 0 & 0 & \sigma_1 & \sigma^2 \end{bmatrix}$ |
| Heterogeneous Toeplitz | TOEPH | $\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_1 & \sigma_1\sigma_3\rho_2 & \sigma_1\sigma_4\rho_3 \\ \sigma_2\sigma_1\rho_1 & \sigma_2^2 & \sigma_2\sigma_3\rho_1 & \sigma_2\sigma_4\rho_2 \\ \sigma_3\sigma_1\rho_2 & \sigma_3\sigma_2\rho_1 & \sigma_3^2 & \sigma_3\sigma_4\rho_1 \\ \sigma_4\sigma_1\rho_3 & \sigma_4\sigma_2\rho_2 & \sigma_4\sigma_3\rho_1 & \sigma_4^2 \end{bmatrix}$ |
| 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) | $\begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{bmatrix}$ |
| Variance components | VC (default) | $\begin{bmatrix} \sigma_B^2 & 0 & 0 & 0 \\ 0 & \sigma_B^2 & 0 & 0 \\ 0 & 0 & \sigma_{AB}^2 & 0 \\ 0 & 0 & 0 & \sigma_{AB}^2 \end{bmatrix}$ |

REPEATED Statement

REPEATED *repeated-effect* < / *options* > ;

The REPEATED statement specifies the **R** matrix in the model. Its syntax is similar to that of the REPEATED statement in PROC MIXED. If you omit this statement, **R** is assumed to be equal to $\sigma^2\mathbf{I}$. The REPEATED statement is available only for the normal distribution with the identity link in this release.

Specifying a *repeated-effect* is required in order to inform PROC BGLIMM of the proper location of the observed repeated responses. The *repeated-effect* must contain only classification variables. You specify the **SUBJECT=** option to set up the blocks of **R**. You can use the **TYPE=** option to define the covariance structure. The levels of the *repeated-effect* must be different for each observation within a subject; otherwise, PROC BGLIMM produces an error message.

Table 28.11 summarizes the *options* available in the REPEATED statement. All *options* are then discussed in alphabetical order.

Table 28.11 Summary of REPEATED Statement Options

| Option | Description |
|---------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| Construction of Covariance Structure | |
| COVPRIOR= | Specifies the prior for the R matrix |
| GROUP= | Defines an effect that specifies heterogeneity in the R covariance structure |
| NUTS | Specifies options for the No-U-Turn Sampler of the Hamiltonian Monte Carlo algorithm for the R matrix |
| SUBJECT= | Identifies the subjects in the R-side model |
| TYPE= | Specifies the type of the R matrix |
| Statistical Output | |
| R | Displays the estimated R matrix |
| RCORR | Display the correlation matrix that corresponds to the estimated R matrix |

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

COVPRIOR=keyword

specifies a prior for the **R** matrix of the residuals, when the **R** matrix is of the UN, UN(1), VC, or TOEP(1) type, where a conjugate sampler is used. This option is ignored for other covariance types, because you assign a flat prior to the **R** matrix if it is of a type other than UN, UN(1), VC, or TOEP(1).

You can specify the following priors:

COVPRIOR=IGAMMA <(SHAPE=a, SCALE=b)>

specifies an inverse gamma prior, $\text{IGamma}(a, b)$, with density $f(t) = \frac{b^a}{\Gamma(a)} t^{-(a+1)} e^{-b/t}$ for each diagonal term of the **R** matrix. It is the default prior for the covariance types UN(1), VC, and TOEP.

You can specify the following parameters for an inverse gamma distribution, enclosed in parentheses:

SHAPE=a

specifies the shape parameter for the inverse gamma distribution. By default, SHAPE=2.

SCALE=b

specifies the scale parameter of the inverse gamma distribution. BY default, SCALE=2.

COVPRIOR=IWISHART <(DF=a, SCALE=b)>

COVPRIOR=IWISH <(DF=a, SCALE=b)>

COVPRIOR=IW <(DF=a, SCALE=b)>

specifies an inverse Wishart prior, **IWISHART**(a,b), for the **R** matrix of the random effects. It is the default prior for the UN covariance type.

You can specify the following parameters for an inverse Wishart distribution, enclosed in parentheses:

DF=*a*

specifies the degrees of freedom of the inverse Wishart distribution. The default is the dimension of the **R** matrix of the random effects plus 3.

SCALE=*b*

specifies $b\mathbf{I}$ for the scale parameter of the inverse Wishart distribution, where **I** is the identity matrix. The default is the dimension of the **R** matrix of the random effects plus 3.

GROUP=effect**GRP=effect**

defines an effect that specifies heterogeneity in the covariance structure of **R**. All observations that have the same level of the *effect* have the same covariance parameters. Each new level of the *effect* produces a new set of covariance parameters that has the same structure as the original group. You should exercise caution in properly defining the *effect*, because strange covariance patterns can result when it is misused. The *effect* can greatly increase the number of estimated covariance parameters.

The GROUP= *effect* must be specified as a variable in the CLASS statement.

NUTS(nuts-options)

specifies options for the No-U-Turn Sampler (NUTS) of the Hamiltonian algorithm that is used to sample the parameters in the **R** matrix of the residuals. The NUTS algorithm is a version of the adaptive Hamiltonian Monte Carlo algorithm with automatic tuning of the step size and number of steps in each iteration. For more information, see the section “[Hamiltonian Monte Carlo Sampler](#)” on page 1260. You can specify the following *nuts-options*:

DELTA=value

specifies the target acceptance rate during the tuning process of the NUTS algorithm. Increasing the *value* can often improve mixing, but it can also significantly slow down the sampling. By default, DELTA=0.6.

MAXHEIGHT=value

specifies the maximum height of the NUTS algorithm tree. The taller the tree, the more gradient evaluations per iteration the procedure calculates. The number of evaluations is 2^{height} . Usually, the height of a tree should be no more than 10 during the sampling stage, but it can go higher during the tuning stage. A larger number indicates that the algorithm is having difficulty converging. By default, MAXHEIGHT=10.

NTU=value

specifies the number of tuning iterations that the NUTS algorithm uses. By default, NTU=1000.

STEPSIZE=value

specifies the initial step size in the NUTS algorithm. By default, STEPSIZE=0.1.

These options are similar to those in the [RANDOM](#) statement.

R

displays the estimated **R** matrix. The ODS table name is R.

RCORR

produces the correlation matrix that corresponds to the estimated **R** matrix. The ODS table name is RCorr.

SUBJECT=effect**SUB=effect**

identifies the subjects for the blocking structure in **R**. Complete independence is assumed across subjects; therefore, this option produces a block diagonal structure in **R** that has identical blocks. The *effect* must be specified as a categorical variable in the CLASS statement.

TYPE=covariance-structure

specifies the covariance structure of the **R** matrix. The **SUBJECT=** option defines the blocks of **R**, and the **TYPE=** option specifies the structure of these blocks. Valid values for *covariance-structure* and their descriptions are provided in Table 28.9, and some examples are shown in Table 28.10. By default, **TYPE=VC**.

Details: BGLIMM Procedure

Generalized Linear Mixed Models

First consider the simplest model: a normal linear model. The quantity of primary interest, y_i , is called the response or outcome variable for the i th individual. The variable \mathbf{x}_i is the $1 \times p$ covariate vector for the fixed effects. The distribution of y_i given \mathbf{x}_i is normal with a mean that is a linear function of \mathbf{x}_i ,

$$y_i = \mathbf{x}_i \boldsymbol{\beta} + \epsilon_i, \quad i = 1, \dots, I$$

$$\epsilon_i \sim N(0, \sigma^2)$$

where $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients (also known as fixed effects) and ϵ_i is the noise with a variance σ^2 .

The normal linear model can be expanded to include random effects, and the model becomes a normal linear mixed model,

$$y_i = \mathbf{x}_i \boldsymbol{\beta} + \mathbf{z}_i \boldsymbol{\gamma}_i + \epsilon_i$$

$$\boldsymbol{\gamma}_i \sim N(0, \mathbf{G}_i)$$

$$\epsilon_i \sim N(0, \sigma^2)$$

where $\boldsymbol{\gamma}_i$ is a $q \times 1$ vector of random effects, \mathbf{z}_i is an $1 \times q$ matrix of covariates for the $\boldsymbol{\gamma}_i$, and \mathbf{G}_i is the covariance matrix of the random effects $\boldsymbol{\gamma}_i$ (**G** is a block diagonal matrix where each block is \mathbf{G}_i).

When an individual i has n_i repeated measurements, the random-effects model for outcome vector \mathbf{y}_i is given by

$$\mathbf{y}_i = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{Z}_i \boldsymbol{\gamma}_i + \boldsymbol{\epsilon}_i, \quad i = 1, \dots, I$$

where \mathbf{y}_i is $n_i \times 1$, \mathbf{X}_i is an $n_i \times p$ matrix of fixed covariates, $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients (also known as fixed effects), $\boldsymbol{\gamma}_i$ is a $q \times 1$ vector of random effects, \mathbf{Z}_i is an $n_i \times q$ matrix of covariates for the $\boldsymbol{\gamma}_i$, and $\boldsymbol{\epsilon}_i$ is an $n_i \times 1$ vector of random errors.

It is further assumed that

$$\boldsymbol{\gamma}_i \sim N(\mathbf{0}, \mathbf{G}_i)$$

$$\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \mathbf{R}_i)$$

where \mathbf{G}_i is the covariance matrix of $\boldsymbol{\gamma}_i$ (\mathbf{G} is a block diagonal matrix where each block is \mathbf{G}_i) and \mathbf{R}_i is the covariance matrix of the residual errors for the i th subject (\mathbf{R} is a block diagonal matrix where each block is \mathbf{R}_i).

There are cases where the relationship between the design matrix (\mathbf{X} and \mathbf{Z}) and the expectation of the response is not linear, or where the distribution for the response is far from normal, even after transformation of the data. The class of generalized linear mixed models unifies the approaches that you need in order to analyze data in those cases. Let \mathbf{y} be the collection of all \mathbf{y}_i , and let \mathbf{X} and \mathbf{Z} be the collection of all \mathbf{X}_i and \mathbf{Z}_i , respectively. A generalized linear mixed model consists of the following:

- the linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$
- the link function $g(\cdot)$ that relates the linear predictor to the mean of the outcome via a monotone link function,

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

where $g(\cdot)$ is a differentiable monotone link function and $g^{-1}(\cdot)$ is its inverse.

- a response distribution in the exponential family of distributions. The distribution can also depend on a scale parameter, ϕ .

A density or mass function in the exponential 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 θ is called the natural (canonical) parameter. The parameter ϕ is a scale parameter, and it is not present in all exponential family distributions. For example, in logistic regression and Poisson regression, $\phi = 1$.

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 θ as a function of μ , the relationship is known as the natural link function or the canonical link function. In other words, modeling data by using a canonical link assumes that $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$; the effect contributions are additive on the canonical scale. The second derivative of $b(\cdot)$, expressed as a function of μ , 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. However, you cannot proceed in the opposite direction.

Likelihood-based inference is based on the marginal likelihood in which the random effects are integrated out. The integration requires numerical methods, such as Gaussian quadrature methods (which can be computationally costly) or Laplace methods (which are faster but not as accurate). The Bayesian approach estimates the joint distribution of all parameters in the model, and it is made possible by the Markov chain Monte Carlo (MCMC) methods. The presence of the random-effects parameters $\boldsymbol{\gamma}$ adds an extra sampling step to the Gibbs algorithm, thus eliminating the need to numerically integrate out $\boldsymbol{\gamma}$ to make inferences about $\boldsymbol{\beta}$. The MCMC methods produce marginal distribution estimates of all fixed-effects parameters, include the \mathbf{G} and \mathbf{R} covariance matrices, making estimation convenient.

Response Probability Distributions

Probability distributions of the response y in generalized linear models are usually parameterized in terms of the mean μ and dispersion parameter ϕ instead of the *natural parameter* θ . The probability distributions that are available in the BGLIMM procedure are shown in the following list. The PROC BGLIMM scale parameter and the variance of y are also shown.

- Normal:

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

- Inverse Gaussian:

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

- Gamma:

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

- Geometric: This is a special case of the negative binomial where $k = 1$.

$$\begin{aligned} f(y) &= \frac{(\mu)^y}{(1+\mu)^{y+1}} \quad \text{for } y = 0, 1, 2, \dots \\ \phi &= 1 \\ \text{Var}(Y) &= \mu(1+\mu) \end{aligned}$$

- Negative binomial:

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

- Poisson:

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

- Binomial:

$$\begin{aligned}
 f(y) &= \binom{n}{r} \mu^r (1 - \mu)^{n-r} \quad \text{for } y = \frac{r}{n}, r = 0, 1, 2, \dots, n \\
 \phi &= 1 \\
 \text{Var}(Y) &= \frac{\mu(1 - \mu)}{n}
 \end{aligned}$$

The negative binomial and zero-inflated negative binomial distributions contain a parameter k , called the negative binomial dispersion parameter. This is not the same as the generalized linear model dispersion ϕ ; rather, it is an additional distribution parameter that must be estimated or set to a fixed value.

For the binomial distribution, the response is the binomial proportion $y = \text{events}/\text{trials}$. The variance function is $V(\mu) = \mu(1 - \mu)$, and the binomial trials parameter n is regarded as a weight w .

Likelihood

The BGLIMM procedure forms the log likelihoods of generalized linear models as

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

where $l(\mu_i, \phi; y_i)$ is the log-likelihood contribution of the i th observation. The individual log-likelihood contributions for the various distributions are as follows:

Binary:

$$\begin{aligned}
 l(\mu_i, \phi; y_i) &= (y_i \log\{\mu_i\} + (1 - y_i) \log\{1 - \mu_i\}) \\
 \text{Var}[Y] &= \mu(1 - \mu), \phi \equiv 1.
 \end{aligned}$$

Binomial:

$$l(\mu_i, \phi; y_i) = (y_i \log\{\mu_i\} + (n_i - y_i) \log\{1 - \mu_i\}) \\ + (\log\{\Gamma(n_i + 1)\} - \log\{\Gamma(y_i + 1)\} - \log\{\Gamma(n_i - y_i + 1)\})$$

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:

$$l(\mu_i, \phi; y_i) = -\log\{\mu_i\} - y_i/\mu_i$$

$$\text{Var}[Y] = \mu^2, \phi \equiv 1.$$

Gamma:

$$l(\mu_i, \phi; y_i) = 1/\phi \log\left\{\frac{y_i \phi}{\mu_i}\right\} - \frac{y_i \phi}{\mu_i} - \log\{y_i\} - \log\{\Gamma(\phi)\}$$

$$\text{Var}[Y] = \phi \mu^2, \phi > 0.$$

Geometric:

$$l(\mu_i, \phi; y_i) = y_i \log\{\mu_i\} - (y_i + 1) \log\{1 + \mu_i\} \\ + \log\left\{\frac{\Gamma(y_i + 1)}{\Gamma(y_i + 1)}\right\}$$

$$\text{Var}[Y] = \mu + \mu^2, \phi \equiv 1.$$

Inverse Gaussian:

$$l(\mu_i, \phi; y_i) = -\frac{1}{2} \left[\frac{(y_i - \mu_i)^2}{y_i \phi \mu_i^2} + \log\{\phi y_i^3\} + \log\{2\pi\} \right]$$

$$\text{Var}[Y] = \phi \mu^3, \phi > 0.$$

Negative binomial:

$$l(\mu_i, \phi; y_i) = y_i \log\{k\mu_i\} - (y_i + 1/k) \log\{1 + k\mu_i\} \\ + \log\left\{\frac{\Gamma(y_i + 1/k)}{\Gamma(1/k)\Gamma(y_i + 1)}\right\}$$

$$\text{Var}[y] = \mu + k\mu^2, k > 0, \phi \equiv 1.$$

For a given k , the negative binomial distribution is a member of the exponential family. The parameter k is related to the scale of the data, because it is part of the variance function. However, it cannot be factored from the variance, as is the case with the ϕ parameter in many other distributions. The parameter k is designated as “Scale” in the output of PROC BGLIMM.

Normal (Gaussian):

$$l(\mu_i, \phi; y_i) = -\frac{1}{2} \left[\frac{(y_i - \mu_i)^2}{\phi} + \log\{\phi\} + \log\{2\pi\} \right]$$

$$\text{Var}[y] = \phi, \phi > 0.$$

Poisson:

$$l(\mu_i, \phi; y_i) = (y_i \log\{\mu_i\} - \mu_i - \log\{\Gamma(y_i + 1)\})$$

$$\text{Var}[y] = \mu, \phi \equiv 1.$$

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 negative log likelihood are then

$$\mathbf{g} = -\frac{\partial L(\boldsymbol{\theta}; \mathbf{y})}{\partial \boldsymbol{\theta}} \quad \mathbf{H} = -\frac{\partial^2 L(\boldsymbol{\theta}; \mathbf{y})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'}$$

Scale and Dispersion Parameters

The parameter ϕ in the log likelihood functions is a scale parameter. McCullagh and Nelder (1989, p. 29) refer to it as the dispersion parameter. With the exception of the normal distribution, ϕ does not correspond to the variance of an observation; the variance of an observation in a generalized linear model is a function of ϕ and μ . In a generalized linear model, the BGLIMM procedure displays the estimate of ϕ as “Scale” in the “Posterior Summaries and Intervals” table. Note that the scale parameter is the same as that reported by the GLIMMIX procedure for almost all distributions, but it is different from that reported by the GENMOD procedure for some distributions of this scale (see the “Parameter Estimates” table in PROC GLIMMIX and PROC GENMOD). The scale that is reported by PROC GENMOD is sometimes a transformation of the dispersion parameter in the log-likelihood function. Table 28.11 displays the “Scale” entries that are reported by the three procedures in terms of the ϕ (or k) parameter.

Table 28.11 Scale Reported in Output Table

| Distribution | PROC BGLIMM | PROC GLIMMIX | PROC GENMOD |
|-------------------|----------------------------------------|----------------------------------------|---------------------|
| Gamma | $\hat{\phi}$ | $1/\hat{\phi}$ | $1/\hat{\phi}$ |
| Inverse Gaussian | $\hat{\phi}$ | $\hat{\phi}$ | $\sqrt{\hat{\phi}}$ |
| Negative binomial | \hat{k} | \hat{k} | \hat{k} |
| Normal | $\hat{\phi} = \widehat{\text{Var}}[y]$ | $\hat{\phi} = \widehat{\text{Var}}[y]$ | $\sqrt{\hat{\phi}}$ |

How PROC BGLIMM Works

PROC BGLIMM is a simulation-based procedure that uses a variety of sampling algorithms to draw samples from the joint posterior distribution of parameters from a generalized linear mixed model (GLMM). Sampling methods include the conjugate sampler, direct sampler, Gamerman algorithm (a variation of the Metropolis-Hastings algorithm that is tailored to generalized linear models), and No-U-Turn Sampler (NUTS, a self-tuning variation of the Hamiltonian Monte Carlo (HMC) method).

For situations in which the conjugate samplers are used, see the section “[Conjugate Sampling](#)” on page 1258. The direct sampling method is used for missing values, where the sampling distribution is known. The

Gamerman algorithm is used for both the fixed-effects and random-effects parameters in nonnormal models. The NUTS algorithm is used for covariance parameters when conjugacy is not available.

PROC BGLIMM updates parameters conditionally, through Gibbs sampling. The fixed-effects parameters β are drawn jointly at each iteration. The random-effect parameters (in a **RANDOM** statement) are updated by clusters, unless the **SUBJECT=** option is not specified. In that situation, the random-effects parameters from the same **RANDOM** statement are updated jointly (for more information about how the random-effects parameters can be parameterized differently with or without the presence of the **SUBJECT=** option, see the section “Treatment of Subjects in the **RANDOM** Statement” on page 1264). Missing data values are updated in sequence, and the G-side and the R-side covariance parameters are updated separately, in their full posterior conditionals.

The rest of this section describes how PROC BGLIMM computes the full conditional distributions in the Gibbs updating. Let $\theta = \{\beta, \mathbf{G}, \mathbf{R}\}$, the collection of all fixed-effects parameters and the covariance matrices; let γ denote random-effects parameters and γ_j denote the random-effects parameters from cluster j . For simplicity, it is assumed that there is only one random effect, thus omitting an extra subindex for γ . The treatment of random effects is identical for effects in multiple **RANDOM** statements.

GLM Parameters

In generalized linear models that have only fixed effects, the log of the posterior density is

$$\log(p(\beta|\mathbf{y}, \mathbf{R})) = \log(\pi(\beta)) + \sum_{i=1}^n \log(f(y_i|\beta, \mathbf{R}))$$

where β is the vector of fixed-effects parameters and $\log(\pi(\beta))$ is the log of the joint prior density of β . The log likelihood, $\log(f(y_i|\beta, \mathbf{R}))$, is computed for the i th observation. The summation reflects the assumption that all observations in the data set are independent. The response variable y_i can be a vector, and \mathbf{R} can be either a scalar or a covariance. The logarithm of the prior distribution of \mathbf{R} is not included because it is constant with respect to β .

The objective function of \mathbf{R} is similar to that of β :

$$\log(p(\mathbf{R}|\mathbf{y}, \beta)) = \log(\pi(\mathbf{R})) + \sum_{i=1}^n \log(f(y_i|\beta, \mathbf{R}))$$

GLMM with Random Effects

In a random-effects model, the conditional distribution of β is similar to that of the fixed-effects-only model,

$$\log(p(\beta|\gamma, \mathbf{y}, \mathbf{R})) = \log(\pi(\beta)) + \sum_{i=1}^n \log(f(y_i|\beta, \gamma, \mathbf{R}))$$

where the log-likelihood function now includes the random effects γ . This construction reflects two PROC BGLIMM modeling settings: all random-effects parameters enter the likelihood function (linearly at the mean level), and the fixed-effects parameters cannot be hyperparameters of γ (hence no $\log(\pi(\gamma_j|\beta))$ terms).

The conditional distribution of \mathbf{R} again mirrors that of β :

$$\log(p(\mathbf{R}|\gamma, \mathbf{y}, \beta)) = \log(\pi(\mathbf{R})) + \sum_{i=1}^n \log(f(y_i|\beta, \gamma, \mathbf{R}))$$

For $\boldsymbol{\gamma}_j$, the following conditional is used:

$$\log(p(\boldsymbol{\gamma}_j | \boldsymbol{\theta}, \mathbf{y})) = \log(\pi(\boldsymbol{\gamma}_j | \mathbf{G})) + \sum_{i \in \{j\text{th cluster}\}} \log(f(\mathbf{y}_i | \boldsymbol{\beta}, \boldsymbol{\gamma}_j, \mathbf{R}))$$

In this computation, only subjects from the j th cluster are used. This reflects the conditional independence assumption that the **RANDOM** statement makes. This simplification in the calculation makes updating the random-effects parameters computationally efficient and enables the procedure to handle random effects that contain large number of clusters just as easily.

The G-side covariance matrix \mathbf{G} depends only on the random effects $\boldsymbol{\gamma}$ and not on the data or other parameters, $\boldsymbol{\beta}$ or \mathbf{R} ,

$$\log(p(\mathbf{G} | \boldsymbol{\gamma})) = \log(\pi(\mathbf{G})) + \sum_j \log(\pi(\boldsymbol{\gamma}_j | \mathbf{G}))$$

where $\pi(\mathbf{G})$ is the prior distribution of \mathbf{G} .

Models with Missing Values

Missing response values are treated as parameters by default and sampled in the MCMC simulation. This mechanism of modeling missing values is referred to as missing at random (MAR). You can delete all observations that contain missing values by using the **MISSING=CC** option in the PROC BGLIMM statement.

Suppose that

$$\mathbf{y} = \{\mathbf{y}_{\text{obs}}, \mathbf{y}_{\text{mis}}\}$$

The response variable \mathbf{y} consists of n_1 observed values, \mathbf{y}_{obs} , and n_2 missing values, \mathbf{y}_{mis} . At each iteration, PROC BGLIMM samples every missing response value (by using the likelihood function as the sampling distribution). After these samples are drawn, the GLMM is reduced to a full data scenario with no missing data. PROC BGLIMM then proceeds to update $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, \mathbf{G} , and \mathbf{R} sequentially, in the same way as described in the section “How PROC BGLIMM Works” on page 1256.

Sampling Methods

This section describes the sampling algorithms that PROC BGLIMM uses.

Conjugate Sampling

Conjugate prior distributions are a type of prior distribution in which the prior and posterior distributions are in the same family of distributions. For example, if you model an independently and identically distributed random variable y_i by using a normal likelihood with known variance σ^2 ,

$$y_i \sim \text{normal}(\mathbf{x}_i' \boldsymbol{\beta}, \sigma^2)$$

then a normal prior on $\boldsymbol{\beta}$,

$$\boldsymbol{\beta} \sim \text{normal}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$$

is a conjugate prior, because the posterior distribution of $\boldsymbol{\beta}$ is also a normal distribution, where the covariance is $\boldsymbol{\Sigma}_0^{-1} + \frac{1}{\sigma^2} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i'$ and the mean is $(\boldsymbol{\Sigma}_0^{-1} + \frac{1}{\sigma^2} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i')^{-1} (\boldsymbol{\Sigma}_0^{-1} \boldsymbol{\beta}_0 + \frac{1}{\sigma^2} \sum_{i=1}^N \mathbf{x}_i y_i)$

PROC BGLIMM uses conjugate samplers in the normal and multivariate normal cases, as shown in Table 28.12.

Table 28.12 Conjugate Sampling in PROC BGLIMM

| Family | Parameter | Prior |
|---------------------|----------------------------------|---------------------|
| Normal | $\boldsymbol{\beta}$ | Multivariate normal |
| Normal | Variance σ^2 | Inverse gamma |
| Multivariate normal | $\boldsymbol{\beta}$ | Multivariate normal |
| Multivariate normal | Covariance $\boldsymbol{\Sigma}$ | Inverse Wishart |

The fixed-effects parameters $\boldsymbol{\beta}$ and the covariances \mathbf{G} and \mathbf{R} are sampled when applicable.

Gamerman Algorithm

The Gamerman algorithm (Gamerman 1997), which is named after the inventor Dani Gamerman, is a special case of the Metropolis algorithm in which the proposal distribution is derived from one iteration of the iterative weighted least squares (IWLS) algorithm. As the name suggests, a weighted least squares algorithm runs inside an iteration loop. For each iteration, a set of weights for the observations is used in the least squares fit. The weights are constructed by applying a weight function to the current residuals. The proposal distribution uses the current iteration's parameter values to form the proposal distribution from which to generate a proposed random value (Gamerman 1997).

The Gamerman algorithm is suitable for both GLM and GLMM models.

The maximum likelihood (ML) estimator in a GLM and the asymptotic variance are obtained by iterative application of weighted least squares (IWLS) to transformed observations. Following McCullagh and Nelder (1989), define the transformed response as

$$\tilde{y}_i(\boldsymbol{\beta}) = \eta_i + (y_i - \mu_i)g'(\mu_i)$$

and define the corresponding weights as

$$\mathbf{W}_i^{-1}(\boldsymbol{\beta}) = b''(\theta_i)[g'(\mu_i)]^2$$

The Gamerman algorithm is summarized as follows:

1. Start with $\boldsymbol{\beta}^{(0)}$ and $t = 1$.
2. Sample $\boldsymbol{\beta}^*$ from the proposal density $\mathbf{N}(\mathbf{m}^{(t)}, \mathbf{C}^{(t)})$, where

$$\begin{aligned} \mathbf{m}^{(t)} &= \{\boldsymbol{\Omega}_{\boldsymbol{\beta}}^{-1} + \mathbf{X}'\mathbf{W}(\boldsymbol{\beta}^{(t-1)})\mathbf{X}\}^{-1} \{\boldsymbol{\Omega}_{\boldsymbol{\beta}}^{-1} \bar{\boldsymbol{\beta}} + \mathbf{X}'\mathbf{W}(\boldsymbol{\beta}^{(t-1)})\tilde{\mathbf{y}}(\boldsymbol{\beta}^{(t-1)})\} \\ \mathbf{C}^{(t)} &= \{\boldsymbol{\Omega}_{\boldsymbol{\beta}}^{-1} + \mathbf{X}'\mathbf{W}(\boldsymbol{\beta}^{(t-1)})\mathbf{X}\}^{-1} \end{aligned}$$

3. Accept β^* with probability

$$\alpha(\beta_{(t-1)}, \beta^*) = \min\left[1, \frac{p(\beta^*|\mathbf{y})q(\beta^*, \beta^{(t-1)})}{p(\beta^{(t-1)}|\mathbf{y})q(\beta^{(t-1)}, \beta^*)}\right]$$

where $p(\beta|\mathbf{y})$ is the posterior density and $q(\beta^*, \beta^{(t-1)})$ and $q(\beta^{(t-1)}, \beta^*)$ are the transitional probabilities that are based on the proposal density $\mathbf{N}(\mathbf{m}^{(\cdot)}, \mathbf{C}^{(\cdot)})$. More specifically, $q(\beta^*, \beta^{(t-1)})$ is an $\mathbf{N}(\mathbf{m}^*, \mathbf{C}^*)$ density that is evaluated at $\beta^{(t-1)}$, whereas \mathbf{m}^* and \mathbf{C}^* have the same expression as $\mathbf{m}^{(t)}$ and $\mathbf{C}^{(t)}$ but depend on β^* instead of $\beta^{(t-1)}$. If β^* is not accepted, the chain stays with $\beta^{(t-1)}$.

4. Set $t = t + 1$ and return to step 1.

PROC BGLIMM uses this algorithm to draw samples for both the fixed-effects parameters β and the random-effects parameters γ : the GLMM simplifies to a GLM when γ is conditioned on; similarly, for the i th cluster, the model for γ_i is simplified to a GLM when β are treated as known and conditioned on.

For the random-effects γ_i block, the same Metropolis-Hastings sampling with the least squares proposal can apply. The conditional posterior is

$$p(\gamma_i|\mathbf{y}, \beta, \mathbf{G}) \propto \exp\left\{\frac{\mathbf{y}_i\boldsymbol{\theta}_i - b(\boldsymbol{\theta}_i)}{\phi_i} - \frac{1}{2}\boldsymbol{\gamma}_i'\mathbf{G}^{-1}\boldsymbol{\gamma}_i\right\}$$

The transformed response is now $\tilde{\mathbf{y}}_i(\gamma_i^{(t-1)})$, and the proposal density is $\mathbf{N}(\mathbf{m}_i^{(t)}, \mathbf{C}_i^{(t)})$, where

$$\begin{aligned}\mathbf{m}_i^{(t)} &= \{\mathbf{G}^{-1} + \mathbf{Z}'\mathbf{W}(\boldsymbol{\gamma}_i^{(t-1)})\mathbf{Z}\}^{-1}\mathbf{Z}'\mathbf{W}(\boldsymbol{\gamma}_i^{(t-1)})\{\tilde{\mathbf{y}}_i(\boldsymbol{\gamma}_i^{(t-1)}) - \mathbf{X}_i\boldsymbol{\beta}\} \\ \mathbf{C}_i^{(t)} &= \{\mathbf{G}^{-1} + \mathbf{Z}'\mathbf{W}(\boldsymbol{\gamma}_i^{(t-1)})\mathbf{X}_i\}^{-1}\end{aligned}$$

Hamiltonian Monte Carlo Sampler

The Hamiltonian Monte Carlo (HMC) algorithm, also known as the hybrid Monte Carlo algorithm, is a version of the Metropolis algorithm that uses gradient information and auxiliary momentum variables to draw samples from the posterior distribution (Neal 2011). The algorithm uses Hamiltonian dynamics to enable distant proposals in the Metropolis algorithm, making it efficient in many scenarios. The HMC algorithm is applicable only to continuous parameters.

HMC translates the target density function to a potential energy function and adds an auxiliary momentum variable \mathbf{r} for each model parameter $\boldsymbol{\theta}$. The resulting joint density has the form

$$p(\boldsymbol{\theta}, \mathbf{r}) \propto p(\boldsymbol{\theta}) \exp\left(-\frac{1}{2}\mathbf{r}'\mathbf{r}\right)$$

where $p(\boldsymbol{\theta})$ is the posterior of the parameters $\boldsymbol{\theta}$ (up to a normalizing constant). HMC draws from the joint space of $(\boldsymbol{\theta}, \mathbf{r})$, discards \mathbf{r} , and retains $\boldsymbol{\theta}$ as samples from $p(\boldsymbol{\theta})$. The algorithm uses the idea of Hamiltonian dynamics in preserving the total energy of a physical system, in which $\boldsymbol{\theta}$ is part of the potential energy function and \mathbf{r} is part of the kinetic energy (velocity). As the velocity changes, the potential energy changes accordingly, leading to the movements in the parameter space.

At each iteration, the HMC algorithm first generates the momentum variables \mathbf{r} , usually from standard normal distributions, that are independent of $\boldsymbol{\theta}$. Then the algorithm follows with a Metropolis update that includes

many steps along a trajectory while maintaining the total energy of the system. One of the most common approaches in moving along this trajectory is the leapfrog method, which involves L steps with a step size ϵ ,

$$\begin{aligned} \mathbf{r}^{t+\epsilon/2} &= \mathbf{r}^t + (\epsilon/2) \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}^t) \\ \boldsymbol{\theta}^{t+\epsilon} &= \boldsymbol{\theta}^t + \epsilon \mathbf{r}^{t+\epsilon/2} \\ \mathbf{r}^{t+\epsilon} &= \mathbf{r}^{t+\epsilon/2} + (\epsilon/2) \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}^{t+\epsilon}) \end{aligned}$$

where $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta})$ is the gradient of the log posterior with respect to $\boldsymbol{\theta}$. After L steps, the proposed state $(\boldsymbol{\theta}^*, \mathbf{r}^*)$ is accepted as the next state of the Markov chain with probability $\min\{1, p(\boldsymbol{\theta}^*, \mathbf{r}^*)/p(\boldsymbol{\theta}, \mathbf{r})\}$.

Although HMC can lead to rapid convergence, it also heavily relies on two requirements: the gradient calculation of the logarithm of the posterior density and carefully selected tuning parameters, in step size ϵ and number of steps L . Step sizes that are too large or too small can lead to acceptance rates that are too low or too high, both of which affect the convergence of the Markov chain. A large L leads to a large trajectory length ($\epsilon \cdot L$), which can move the parameters back to their original positions. A small L limits the movement of the chain.

An example of an adaptive HMC algorithm with automatic tuning of ϵ and L is the No-U-Turn Sampler (NUTS; Hoffman and Gelman 2014). The NUTS algorithm uses a doubling process to build a binary tree whose leaf nodes correspond to the states of the parameters and momentum variables. The initial tree has a single node with no heights ($j = 0$). The doubling process expands the tree to either the left or right in a binary fashion, and in each direction, the algorithm takes 2^j leapfrog steps of size ϵ . Obviously, as the height of the tree (j) increases, the computational cost increases dramatically. The tree expands until one sampling trajectory makes a U-turn and starts to revisit parameter space that has been already explored. The NUTS algorithm tunes ϵ so that the actual acceptance rate during the doubling process is close to a predetermined target acceptance probability δ (usually set to 0.6 or higher). When the tuning stage ends, the NUTS algorithm proceeds to the main sampling stage and starts to draw posterior samples that have a fixed ϵ value. Increasing the targeted acceptance probability δ can often improve mixing, but it can also slow down the process significantly. For more information about the NUTS algorithm and its efficiency, see Hoffman and Gelman (2014).

Prior Distributions

A GLMM model contains various types of parameters: the coefficients for the fixed effects that are specified in the **MODEL** statement; the coefficients for the random effects that are specified in the **RANDOM** statement; and the parameters for the covariance matrices, including both the G-side and R-side covariance matrices.

Prior for the Fixed-Effects Coefficients

In GLMMs, flat priors on the fixed-effects coefficients ($\boldsymbol{\beta}$) are considered to be noninformative. The flat prior assigns equal likelihood for all possible values of the parameter, $\pi(\boldsymbol{\beta}) \propto 1$. This is the default prior for the fixed-effects coefficients in PROC BGLIMM.

In addition to the flat prior, a normal prior that has very large variance is also considered to be noninformative or weakly informative. The following statement specifies such a prior:

```
model y = x / cprior=normal(var=1e4);
```

This normal prior is noninformative because its variance value is sufficiently larger than the posterior variances of all the β parameters.

On the other hand, you can use an informative prior by making the variance small in the normal distribution. If you want to have a specific mean and covariance for the normal prior, you can provide a SAS data set that contains the mean and covariance information of the normal prior through the `INPUT=SAS-data-set` in the `CPRIOR=` option,

```
model y = x / cprior=normal(input=MyPrior);
```

where `MyPrior` is the name of a SAS data set.

Prior for the Random-Effects Coefficients

The random-effects coefficients γ are assumed to have the normal prior

$$\gamma \sim \text{Normal}(\mathbf{0}, \mathbf{G})$$

where $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_n)^T$ and \mathbf{G} is a block diagonal matrix where each block is \mathbf{G}_i . For a specific subject i , the random-effects coefficients γ_i are normally distributed with mean zero and the covariance matrix \mathbf{G}_i . When the dimension is the same, $\mathbf{G}_i = \mathbf{G}_j$ for any i and j .

You use the `TYPE=` option in the `RANDOM` statement to specify the types of the \mathbf{G} matrix.

Prior for the G-Side Covariance

PROC BGLIMM supports the following prior distributions of \mathbf{G} :

- Inverse Wishart distribution: $\pi(\mathbf{G}_i) = \text{IW}(a, bI)$. You can specify this distribution as follows:

```
random x / sub=Id covprior=iwishart(df=a, scale=b);
```

The inverse Wishart distribution is a generalization of the inverse gamma distribution; when the dimension is 1, the inverse Wishart distribution is equivalent to an inverse gamma distribution. The inverse Wishart prior is frequently used in Bayesian analysis, in part because it is a conjugate prior for a normal covariance, which leads to efficient sampling. An inverse Wishart prior that has small degrees of freedom (interpreted as prior sample size) and small diagonal values of the scale matrix is considered to be weakly informative.

The inverse Wishart prior applies to the covariance types UN, UN(1), VC, and TOEP. It is the default prior for the UN type.

- Inverse gamma distribution: $\pi(\sigma_g^2) = \text{IG}(a, b)$. You can specify this distribution as follows:

```
random x / sub=Id covprior=igamma(shape=a, scale=b);
```

The inverse gamma prior applies to the diagonal variance terms of a covariance that is of type UN, UN(1), VC, or TOEP. It is the default prior for the covariance types UN(1), VC, and TOEP.

- Uniform distribution: $\pi(\sigma_\gamma) = \frac{1}{\text{Upper-Lower}}$. You can specify this distribution as follows:

```
random x / sub=Id covprior=uniform(lower=, upper=);
```

The uniform prior is on the standard deviation, not the variance, terms of \mathbf{G} . This prior applies to the diagonal standard deviation terms of a covariance that is of type UN, UN(1), VC, or TOEP.

- Half-Cauchy distribution: $\pi(\sigma_\gamma^2) = \text{half-Cauchy}(a)$. You can specify this distribution as follows:

```
random x / sub=Id covprior=halfcauchy(scale=a);
```

The half-Cauchy prior is a truncated prior whose lower bound is equal to zero. The prior applies to the variance terms of a covariance that is of type UN, UN(1), VC, or TOEP.

The half-Cauchy prior is a special case of the conditionally conjugate folded-noncentral- t distribution. The right tail of the half-Cauchy distribution is thick and decays slowly. Large a translates to weakly informative, a goes to infinity, and the half-Cauchy distribution becomes a uniform distribution.

- Half-normal distribution: $\pi(\sigma_\gamma^2) = \text{half-normal}(0, \sigma^2 = a)$. You can specify this distribution as follows:

```
random x / sub=Id covprior=halfnormal(var=a);
```

The half-normal prior is a truncated prior whose lower bound is equal to zero. The prior applies to the variance terms of a covariance that is of type UN, UN(1), VC, or TOEP.

- Scaled inverse Wishart distribution. You can specify this distribution as follows:

```
random x / sub=Id covprior=siwishart(df=a, scale=b, var=c);
```

This is a generalization of the inverse Wishart prior. The distribution decomposes the \mathbf{G}_i matrix into variance and correlation components and specifies separate priors for each component (O'Malley and Zaslavsky 2008). Define $\mathbf{\Delta} = \text{diag}(\boldsymbol{\delta})$, where $\boldsymbol{\delta} = (\delta_1, \delta_2, \dots, \delta_n)$ and $\delta_i > 0$; let $\mathbf{\Phi}$ be a positive definite matrix; and write $\mathbf{G}_i = \mathbf{\Delta}\mathbf{\Phi}\mathbf{\Delta}$,

$$\pi(\mathbf{\Delta}) = \text{IW}(a, b\mathbf{I}), \pi(\log(\delta_i)) = \text{normal}(0, \sigma_\delta^2)$$

where the hyperparameters a and b are the degrees of freedom and scale parameters, respectively, of the inverse Wishart prior; $\sigma_\delta^2 = c$ is the variance of the normal prior; and the default variance is 1. You can choose values for the hyperparameters a and b , and the default is the dimension of the \mathbf{G}_i matrix plus 3 for both a and b .

Although $\mathbf{\Phi}$ determines the correlations, it is not constrained to be a correlation matrix, so the model is overparameterized. The associated correlation matrix is $\mathbf{\Omega}(\mathbf{\Phi}) = \text{diag}(\mathbf{\Phi})^{-\frac{1}{2}} \mathbf{\Phi} \text{diag}(\mathbf{\Phi})^{-\frac{1}{2}}$, where $\text{diag}(\mathbf{\Phi})$ retains the diagonal elements of $\mathbf{\Phi}$ but has zeros elsewhere.

Like the inverse Wishart prior, the scaled inverse Wishart prior applies to types UN, UN(1), VC, and TOEP.

Prior for the Scale Parameter

In models that have a scale parameter (see Table 28.11), you can specify the following prior distribution:

- Inverse gamma distribution: $\pi(\phi) = \text{IG}(a,b)$. You can specify this distribution as follows:

```
model y = x / scaleprior = igamma(shape=a, scale=b);
```

The inverse gamma prior applies to all distributions that have a scale parameter.

- Gamma distribution: $\pi(\phi) = \text{gamma}(a,b)$. You can specify this distribution as follows:

```
model y = x / scaleprior = gamma(shape=a, iscale=b);
```

The gamma prior applies to the gamma, inverse gamma, and negative binomial distributions. It does not apply to the normal likelihood function with an identity link.

- Improper prior: $\pi(\phi) \propto 1/\phi$. You can specify this distribution as follows:

```
model y = x / scaleprior = improper;
```

The improper prior applies to the gamma, inverse gamma, and negative binomial distributions. It does not apply to the normal likelihood function with an identity link.

Prior for the R-Side Covariance

You can specify either an inverse Wishart distribution or an inverse gamma distribution for the **R** covariance matrix. These two distributions are the same as what you can specify for the **G** covariance matrix. The inverse Wishart distribution is the default prior for the UN type, and the inverse gamma distribution is the default prior for types UN(1), VC, and TOEP.

Treatment of Subjects in the RANDOM Statement

PROC BGLIMM supports the syntax of specifying random effects without specifying a **SUBJECT=** option, but this is different from specifying an intercept random effect and using the same random effects in the **SUBJECT=** option.

For example, suppose you have a nested effect **A(B)**, where both A and B are categorical variables. You can specify the following statement:

```
random A(B) ;
```

In the traditional mixed modeling procedures, such as PROC MIXED and PROC GLIMMIX, this specification is the same as

```
random int / subject = A(B);
```

But this is not the case in PROC BGLIMM. The difference is as follows:

- When you specify the **SUBJECT=** option in PROC BGLIMM, the procedure fits a normal prior (with default **TYPE=VC**),

$$\pi(\gamma_j) \sim \text{normal}(0, \sigma_\gamma^2), \quad j = 1, \dots, J$$

where γ_j is the intercept from cluster j and J is the total number of unique clusters in effect **A(B)**. This is the expected way of specifying a prior on intercept random effects.

- When you omit the **SUBJECT=** option, PROC BGLIMM assumes that there is only one cluster (instead of J clusters). This cluster, which represents the entire data set, has nested (categorical) effects of **A(B)**.

This is equivalent to treating **A(B)** as categorical fixed effects in the regression model (in the **MODEL** statement):

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

The exception is that, instead of a noninformative or flat prior on the regression coefficients, a hierarchical normal prior is assumed (again, following the specification of the **TYPE=** covariance),

$$\beta_j \sim \text{normal}(0, \sigma_\beta^2), \quad j = 1, \dots, K$$

where β_j (used here instead of γ_j to emphasize the different treatment in modeling by PROC BGLIMM) is a categorical coefficient for the j th level and K is less or equal to J , depending on the rank of the regression model.

In other words, after you specify a random effect without using the **SUBJECT=** option, the effect becomes equivalent to a fixed effect (but with a shrinkage prior). The number of estimable parameters from that effect depends on the rank of the model, so you will often see that some of the cluster parameters are inestimable (hence, their value is displayed as zero).

In PROC MIXED and PROC GLIMMIX, the two specifications lead to identical numerical estimates, because neither procedure requires the design matrix (of the regression model) to be nonsingular. But PROC BGLIMM has this requirement, and this results in different posterior estimates.

If you want to specify a random-effects model in the strict Bayesian sense, use the **SUBJECT=** option in the **RANDOM** statement.

Initial Values of the Markov Chains

A GLMM model has various types of parameters: fixed-effects parameters (in the **MODEL** statement), random-effects parameters (in the **RANDOM** statement), parameters in the covariance matrix, and missing data variables in the response.

For the fixed-effects parameters, PROC BGLIMM generates initial values that are based on the optimization of the posterior density functions. To assign initial values to the fixed-effects parameters, you can use the **INIT=** option in the **MODEL** statement. If there are multiple fixed effects, you can provide a list of numbers, where the length of the list is the same as the dimension of the fixed effects. Each number is then given to all corresponding fixed-effects parameters in order.

For all the individual random-effects parameters, PROC BGLIMM sets the initial values to zero.

The initial values for either the **G** or **R** matrix are set to the identity matrix.

For missing data in the response, PROC BGLIMM uses the sample average of the nonmissing values as the initial value. The procedure requires some response value to be nonmissing—if all values of a particular variable are missing, PROC BGLIMM issues an error and stops.

Missing Data

When you have missing data, you can use the **MISSING=** option in the PROC BGLIMM statement as follows to specify how you want to handle the missing response values:²

- If you specify **MISSING=CC** (CC stands for complete cases), PROC BGLIMM discards all observations that have missing or partial missing values (for example, in a repeated measures model) before carrying out the simulation. This is equivalent to assuming that the missing values are missing completely at random (MCAR).
- If you specify **MISSING=CCMODELY**, PROC BGLIMM treats missing response values as parameters and includes the sampling of the missing data as part of the simulation. The procedure discards all observations that have missing covariates. This is equivalent to assuming that the missing values are missing at random (MAR).

Different types of missing data were first defined by Rubin (1976). For a comprehensive treatment of missing data analysis, see Little and Rubin (2002). PROC BGLIMM does not model the missing not at random (MNAR) type of missing data.

Autocall Macros for Postprocessing

Although PROC BGLIMM provides a number of convergence diagnostic tests and posterior summary statistics, it performs the calculations only for the default tests and statistics or only if you specify the necessary options. If you want to analyze the posterior draws of unmonitored parameters or functions of the parameters that are calculated in later DATA step calls, you can use the autocall macros that are listed in Table 28.13.

²A missing value is usually, although not necessarily, represented by a single period (.) in the input data set.

Table 28.13 Postprocessing Autocall Macros

| Macro | Description |
|----------|---------------------------------------------------|
| %ESS | Effective sample sizes |
| %GEWEKE* | Geweke diagnostic |
| %HEIDEL* | Heidelberger-Welch diagnostic |
| %MCSE | Monte Carlo standard errors |
| %POSTACF | Autocorrelation |
| %POSTCOR | Correlation matrix |
| %POSTCOV | Covariance matrix |
| %POSTINT | Equal-tail and HPD intervals |
| %POSTSUM | Mean, standard deviation, and various quantiles |
| %RAFTERY | Raftery diagnostic |
| %SUMINT | Mean, standard deviation, and HPD interval |
| %TADPLOT | Trace plot, autocorrelation plot and density plot |

*The %GEWEKE and %HEIDEL macros use a different optimization routine than PROC BGLIMM uses. As a result, there might be numerical differences in some cases, especially when the sample size is small.

Table 28.14 lists options that are shared by all postprocessing autocall macros. For macro-specific options, see Table 28.15.

Table 28.14 Shared Options

| Option | Description |
|-------------------|---------------------------------------------------------------------|
| DATA=SAS-data-set | Names the input data set that contains posterior samples |
| OUT=SAS-data-set | Specifies a name for the output SAS data set to contain the results |
| PRINT=YES NO | Displays the results (the default is YES) |
| VAR=variable-list | Specifies the variables on which to perform the calculation |

Suppose that the data set that contains posterior samples is called `Post` and the variables of interest are defined in the macro variable `&PARMS`. The following statement calls the %ESS macro and calculates the effective sample sizes for each variable:

```
%ESS(data=Post, var=Alpha Beta U_1-U_17)
```

By default, the ESS estimates are displayed. You can choose not to display the result and instead use the following statement to save the output to a data set:

```
%ESS(data=Post, var=&parms, print=NO, out=eout)
```

Some of the macros can take additional options, which are listed in Table 28.15.

Table 28.15 Macro-Specific Options

| Macro | Option | Description |
|----------|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| %ESS | AUTOCORLAG= <i>numeric</i> | Specifies the maximum number of autocorrelation lags used in computing the ESS estimates. By default, AUTOCORLAG=MIN(500, NOBS/4), where NOBS is the sample size of the input data set. |
| | HIST=YES NO | Displays a histogram of all ESS estimates. By default, HIST=NO. |
| %GEWEKE | FRAC1= <i>numeric</i> | Specifies the earlier portion of the Markov chain used in the test. By default, FRAC1=0.1. |
| | FRAC2= <i>numeric</i> | Specifies the latter portion of the Markov chain used in the test. By default, FRAC2=0.5. |
| %HEIDEL | SALPHA= <i>numeric</i> | Specifies the α level for the stationarity test. By default, SALPHA=0.05. |
| | HALPHA= <i>numeric</i> | Specifies the α level for the halfwidth test. By default, HALPHA=0.05. |
| | EPS= <i>numeric</i> | Specifies a small positive number ϵ such that if the halfwidth is less than ϵ times the sample mean of the remaining iterations, the halfwidth test is passed. By default, EPS=0.1. |
| %MCSE | AUTOCORLAG= <i>numeric</i> | Specifies the maximum number of autocorrelation lags used in computing the Monte Carlo standard error estimates. By default, AUTOCORLAG=MIN(500, NOBS/4), where NOBS is the sample size of the input data set. |
| %POSTACF | LAGS=%str(<i>numeric-list</i>) | Specifies which autocorrelation lags to calculate. The default values are 1, 5, 10, and 50. |
| %POSTINT | ALPHA= <i>value</i> | Specifies the α level ($0 < \alpha < 1$) for the interval estimates. By default, ALPHA=0.05. |
| %RAFTERY | Q= <i>numeric</i> | Specifies the order of the quantile of interest. By default, Q=0.025. |
| | R= <i>numeric</i> | Specifies the margin of error for measuring the accuracy of estimation of the quantile. By default, R=0.005. |
| | EPS= <i>numeric</i> | Specifies the tolerance level for the stationary test. By default, EPS=0.001. |

For example, the following statement calculates and displays autocorrelation at lags 1, 6, 11, 50, and 100. Note that the lags in the *numeric-list* must be separated by commas.

```
%POSTACF (data=Post, var=&parms, lags=%str(1 to 15 by 5, 50, 100))
```

Regenerating Diagnostics Plots

By default, PROC BGLIMM generates three plots: the trace plot, the autocorrelation plot, and the kernel density plot. Unless ODS Graphics is enabled before the procedure is called, it is hard to generate the same graph afterward. Directly using the `Stat.Bglimm.Graphics.TraceAutocorrDensity` template is not

feasible. The easiest way to regenerate the same graph is to use the %TADPLOT autocall macro. This macro requires you to specify an input data set (which is the output data set from a previous PROC BGLIMM call) and a list of variables that you want to plot.

Suppose that the output data set `Postsamp` contains posterior draws for the regression coefficients of `Mode1`, `Mode2`, and `Mode3`. If you want to examine these parameters graphically, you can use the following statement to regenerate the graphs:

```
%TADPLOT (data=Postsamp, var=Mode1 Mode2 Mode3)
```

Displayed Output

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

Model- and Data-Related ODS Tables

Model Information

The “Model Information” table displays basic information about the model, such as the response variable, frequency variable, link function, and model category that the BGLIMM procedure determines on the basis of your input and options, sampling algorithm, burn-in size, simulation size, thinning, and random number seed. This table also displays the distribution of the data that PROC BGLIMM assumes. For information about the supported response distributions, see the description of the `DIST=` option.

Number of Observations

The “Number of Observations” table displays the number of observations that are read from the input data set and the number of observations that are used in the analysis. If the events/trials syntax is used, the number of events and trials is also displayed.

Response Profile

The “Response Profile” table displays the ordered values from which the BGLIMM procedure determines the probability that is modeled as an event in binary models. For each response category level, the frequency that is used in the analysis is reported. You can determine the ordering of the response values by specifying *response-options* in the `MODEL` statement. For binary models, the note that follows this table indicates which outcome is modeled as the event in binary models and which value serves as the reference category.

The “Response Profile” table is not produced for binomial (events/trials) data. You can find information about the number of events and trials in the “Number of Observations” table.

Class Level Information

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

Sampling-Related ODS Tables

Parameters Initial Value

The “Parameters Initial” table (ODS table name `ParametersInit`) shows the initial value of each fixed regression coefficient. This table is not displayed by default. You can display it by specifying the `INIT=PINIT` option in the `MODEL` statement.

Constant Priors for Fixed Effects

The “Constant Priors for Fixed Effects” table (ODS table name `ConstantCoeffPrior`) shows the information for the constant prior for the fixed-effects coefficients. The table is displayed by default for a model that contains fixed effects.

Independent Normal Priors for Fixed Effects

The “Independent Normal Priors for Fixed Effects” table (ODS table name `IndepNormalCoeffPrior`) shows the information for the independent normal prior for the fixed-effects coefficients. The table is displayed when you specify that fixed effects have an independent normal prior.

Normal Priors for Fixed Effects

The “Normal Priors for Fixed Effects” table (ODS table name `NormalCoeffPrior`) shows the information for the normal prior for the fixed-effects coefficients. The table is displayed when you specify that fixed effects have a normal prior and are not independent.

Priors for Scale and Covariance Parameters

The “Priors for Scale and Covariance Parameters” table (ODS table name `ScaleCovPrior`) shows the prior distributions for the scale and covariance parameters. The table is displayed by default for a model that contains any scale and covariance parameters.

ODS Tables Related to Posterior Statistics

PROC BGLIMM calculates some essential posterior statistics and outputs them to a number of ODS tables. Some of the ODS tables are produced by default, and you can request others by specifying an option in the PROC BGLIMM statement. For more information about the calculations, see the section “[Summary Statistics](#)” on page 158 in Chapter 7, “[Introduction to Bayesian Analysis Procedures](#).”

Summary and Interval Statistics

The “Posterior Summaries and Intervals” table (ODS table name `PostSumInt`) contains basic summary and interval statistics for each parameter. The table lists the number of posterior samples, the posterior mean and standard deviation estimates, and the highest posterior density (HPD) interval estimates. This table is displayed by default.

Summary Statistics

The “Posterior Summaries” table (ODS table name `PostSummaries`) contains basic statistics for each parameter. The table lists the number of posterior samples, the posterior mean and standard deviation estimates, and the percentile estimates. This table is not displayed by default. You can request it by specifying the `STATISTICS=SUM` option in the PROC BGLIMM statement.

Correlation Matrix

The “Posterior Correlation Matrix” table (ODS table name Corr) contains the posterior correlation of model parameters. This table is not displayed by default. You can display it by specifying the **STATISTICS=CORR** option in the PROC BGLIMM statement.

Covariance Matrix

The “Posterior Covariance Matrix” table (ODS table name Cov) contains the posterior covariance of model parameters. This table is not displayed by default. You can display it by specifying the **STATISTICS=COV** option in the PROC BGLIMM statement.

Deviance Information Criterion

The “Deviance Information Criterion” table (ODS table name DIC) contains the deviance information criterion (DIC) of the model. This table is not displayed by default. You can display it by specifying the **DIC** option in the PROC BGLIMM statement. For more information about the calculations, see the section “Deviance Information Criterion (DIC)” on page 160 in Chapter 7, “Introduction to Bayesian Analysis Procedures.”

Interval Statistics

The “Posterior Intervals” table (ODS table name PostIntervals) contains the equal-tail and highest posterior density (HPD) interval estimates for each parameter. The default α value is 0.05, and you can change it to other levels by using the **STATISTICS=** option in the PROC BGLIMM statement. This table is not displayed by default. You can display it by specifying the **STATISTICS=INT** option.

Estimated G Matrix

The “Estimated G Matrix” table (ODS table name G) contains the estimated G matrix. This table is not displayed by default. You can display it by specifying the **G** option in a **RANDOM** statement.

Estimated G Correlation Matrix

The “Estimated G Correlation Matrix” table (ODS table name GCorr) contains the correlation matrix that corresponds to the estimated G matrix. This table is not displayed by default. You can display it by specifying the **GCORR** option in a **RANDOM** statement.

Estimated R Matrix

The “Estimated R Matrix” table (ODS table name R) contains the R matrix for the **REPEATED** statement. This table is not displayed by default. You can request it by specifying the **R** option in a **REPEATED** statement.

Estimated R Correlation Matrix

The “Estimated R Correlation Matrix” table (ODS table name RCorr) contains the correlation matrix that corresponds to the estimated R matrix. This table is not displayed by default. You can display it by specifying the **RCORR** option in the **REPEATED** statement.

ODS Tables Related to Convergence Diagnostics

PROC BGLIMM provides convergence diagnostic tests that check for Markov chain convergence. It produces a number of ODS tables that you can display and save individually. For information about calculations,

see the section “Statistical Diagnostic Tests” on page 149 in Chapter 7, “Introduction to Bayesian Analysis Procedures.”

Autocorrelation

The “Autocorrelations” table (ODS table name AUTOCORR) contains the first-order autocorrelations of the posterior samples for each parameter. The Parameter column states the name of the parameter. By default, PROC BGLIMM displays lag 1, 5, 10, and 50 estimates of the autocorrelations. You can request different autocorrelations by specifying the **DIAGNOSTICS=AUTOCORR(LAGS=)** option in the PROC BGLIMM statement. This table is displayed by default.

Effective Sample Size

The “Effective Sample Sizes” table (ODS table name ESS) calculates the effective sample size of each parameter. For more information, see the section “Effective Sample Size” on page 157 in Chapter 7, “Introduction to Bayesian Analysis Procedures.” This table is displayed by default.

Monte Carlo Standard Errors

The “Monte Carlo Standard Errors” table (ODS table name MCSE) calculates the standard errors of the posterior mean estimate. For more information, see the section “Standard Error of the Mean Estimate” on page 158 in Chapter 7, “Introduction to Bayesian Analysis Procedures.” This table is not displayed by default. You can display it by specifying the **DIAGNOSTICS=MCSE** option in the PROC BGLIMM statement.

Geweke Diagnostics

The “Geweke Diagnostics” table (ODS table name Geweke) lists the results of the Geweke diagnostic test. For more information, see the section “Geweke Diagnostics” on page 151 in Chapter 7, “Introduction to Bayesian Analysis Procedures.” This table is not displayed by default. You can display it by specifying the **DIAGNOSTICS=GEWEKE** option in the PROC BGLIMM statement.

Heidelberger-Welch Diagnostics

The “Heidelberger-Welch Diagnostics” table (ODS table name Heidelberger) lists the results of the Heidelberger-Welch diagnostic test. The test consists of two parts: a stationarity test and a halfwidth test. For more information, see the section “Heidelberger and Welch Diagnostics” on page 153 in Chapter 7, “Introduction to Bayesian Analysis Procedures.” This table is not displayed by default. You can display it by specifying the **DIAGNOSTICS=HEIDEL** option in the PROC BGLIMM statement.

Raftery-Lewis Diagnostics

The “Raftery-Lewis Diagnostics” table (ODS table name Raftery) lists the results of the Raftery-Lewis diagnostic test. For more information, see the section “Raftery and Lewis Diagnostics” on page 154 in Chapter 7, “Introduction to Bayesian Analysis Procedures.” This table is not displayed by default. You can display it by specifying the **DIAGNOSTICS=RAFTERY** option in the PROC BGLIMM statement.

ODS Table Names

Each table that the BGLIMM procedure creates has a name that is associated with it. You must use this name to refer to the table when you use ODS statements. These names are listed in [Table 28.16](#).

Table 28.16 ODS Tables Produced by PROC BGLIMM

| Table Name | Description | Statement or Option |
|-----------------------|-------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| AutoCorr | Autocorrelation statistics for each parameter | DIAG=AUTOCORR |
| ClassLevels | Class level information | Default |
| Coef | L matrix coefficients | E option in ESTIMATE statements |
| ConstantCoeffPrior | Constant prior information for fixed effects | Default |
| Corr | Correlation matrix of the posterior samples | STATS=CORR |
| Cov | Covariance matrix of the posterior samples | STATS=COV |
| DIC | Deviance information criterion | DIC |
| ESS | Effective sample size for each parameter | Default |
| Estimates | Results from ESTIMATE statements | ESTIMATE |
| G | Estimated G matrix | G |
| GCORR | Correlation matrix of the estimated G matrix | GCORR |
| Geweke | Geweke diagnostics for each parameter | DIAG=GEWEKE |
| Heidelberger | Heidelberger-Welch diagnostics for each parameter | DIAG=HEIDEL |
| IndepNormalCoeffPrior | Independent normal prior information for fixed effects | COEFFPRIOR=NORMAL(VAR=c) |
| MCSE | Monte Carlo standard error for each parameter | STATS=MCSE |
| ModelInfo | Model information | Default |
| NObs | Number of observations | Default |
| NormalCoeffPrior | Normal prior information for fixed effects | COEFFPRIOR=NORMAL(INPUT=SAS-data-set) |
| ParametersInit | Parameter initial values | INIT=PINIT |
| PostIntervals | Equal-tail and HPD intervals for each parameter | STATS=INT |
| PostSumInt | Posterior statistics for each parameter, including sample size, mean, standard deviation, and HPD intervals | Default |
| PostSummaries | Posterior statistics for each parameter, including sample size, mean, standard deviation, and percentiles | STATS=SUM |
| R | Estimated R matrix | R |
| Raftery | Raftery-Lewis diagnostics for each parameter | DIAG=RAFTERY |

Table 28.16 *continued*

| Table Name | Description | Statement or Option |
|---------------|-----------------------------------------------------|---------------------|
| RCORR | Correlation matrix of the estimated R matrix | RCORR |
| ScaleCovPrior | Priors for scale and covariance parameters | Default |

ODS Graphics

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, “[Statistical Graphics Using ODS](#).”

Before you create graphs, ODS Graphics must be enabled (for example, by specifying the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section “[Enabling and Disabling ODS Graphics](#)” on page 623 in Chapter 21, “[Statistical Graphics Using ODS](#).”

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section “[A Primer on ODS Statistical Graphics](#)” on page 622 in Chapter 21, “[Statistical Graphics Using ODS](#).”

When ODS Graphics is enabled, the BGLIMM procedure by default produces plots of the partial predictions for each spline term in the model. Use the PLOTS option in the PROC BGLIMM statement to control aspects of these plots.

PROC BGLIMM assigns a name to each graph that it creates by using ODS. You can use these names to refer to the graphs when using ODS. The names are listed in [Table 28.17](#).

You can refer by name to every graph that is produced through ODS Graphics. The names of the graphs that PROC BGLIMM generates are listed in [Table 28.17](#).

Table 28.17 Graphs Produced by PROC BGLIMM

| ODS Graph Name | Plot Description | Option |
|----------------|----------------------------------------------------|--------------------------------|
| ADPanel | Autocorrelation function and density panel | PLOTS=(AUTOCORR DENSITY) |
| AutocorrPanel | Autocorrelation function panel | PLOTS=AUTOCORR |
| AutocorrPlot | Autocorrelation function plot | PLOTS(UNPACK)=AUTOCORR |
| DensityPanel | Density panel | PLOTS=DENSITY |
| DensityPlot | Density plot | PLOTS(UNPACK)=DENSITY |
| TADPanel | Trace, density, and autocorrelation function panel | PLOTS=(TRACE AUTOCORR DENSITY) |
| TAPanel | Trace and autocorrelation function panel | PLOTS=(TRACE AUTOCORR) |

Table 28.17 *continued*

| ODS Graph Name | Plot Description | Option |
|----------------|-------------------------|-----------------------|
| TDPanel | Trace and density panel | PLOTS=(TRACE DENSITY) |
| TracePanel | Trace panel | PLOTS=TRACE |
| TracePlot | Trace plot | PLOTS(UNPACK)=TRACE |

Examples: BGLIMM Procedure

Example 28.1: Normal Regression with Repeated Measurements

This example illustrates how to use PROC BGLIMM to fit a repeated measures model.

The data were collected by a pharmaceutical company that examined the effects of three drugs on the respiratory capacity of patients with asthma. Treatments involved a standard drug (A), a test drug (C), and a placebo (P). Patients received each treatment on different days. The forced expiratory volume (FEV) of each patient was measured hourly for eight hours following treatment, and a baseline FEV was also recorded. Analysis in this section is based on the scenario that each patient received all three treatments on different visits. For more information about this data set (Fev1), see Littell et al. (2006, Chapter 5).

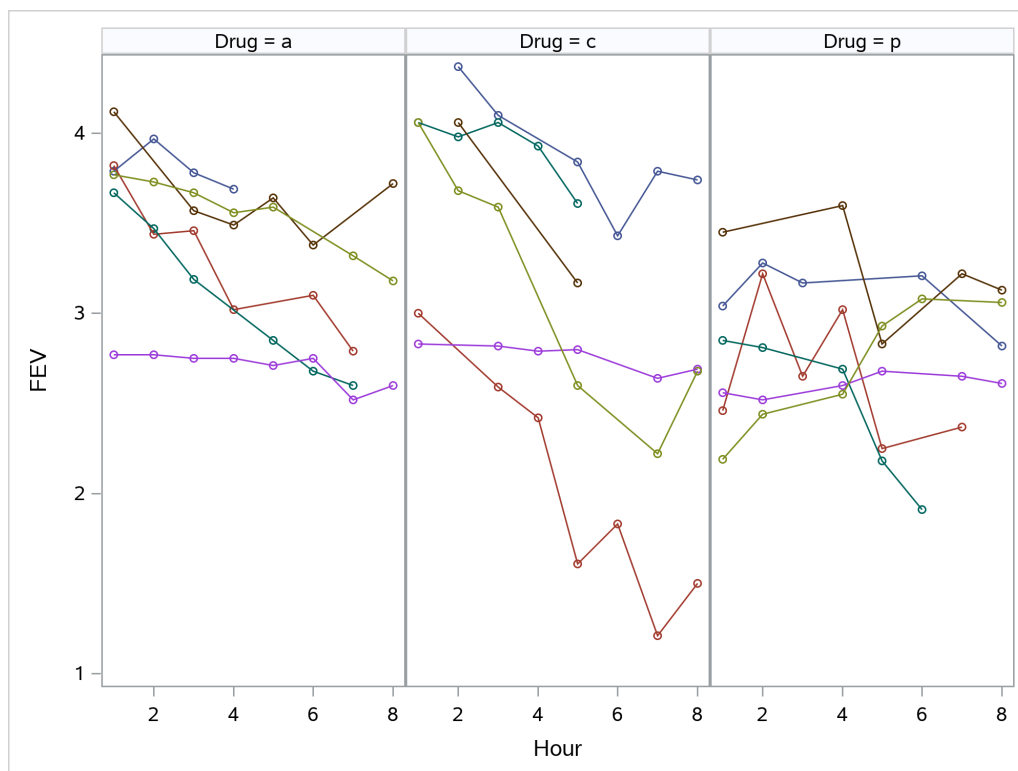
The following statements show the first few records of the Fev data set. Patient is the patient ID (total number 24), BaseVal is the baseline measurement, Drug is the drug type, Hour measures time (8 hours), and FEV is the response variable.

```
data Fev;
  input Patient Baseval Drug$ Hour FEV;
  datalines;
  201      2.46      a      1      2.68
  201      2.46      a      2      2.76
  201      2.46      a      3      2.50
  201      2.46      a      4      2.30
  201      2.46      a      5      2.14
  201      2.46      a      6      2.40
  201      2.46      a      7      2.33

  ... more lines ...

  232      2.88      p      1      3.04
  232      2.88      p      4      3.37
  232      2.88      p      5      2.69
  232      2.88      p      6      2.89
;
```

The data are unbalanced, meaning that some patients have missing measurements at certain time points. A subset of patient profiles is plotted in [Output 28.1.1](#).

Output 28.1.1 Profile Plots from Six Patients

There are a number of ways that you can fit this model to account for the patient effect as well as the time (Hour) effect. For example, you can fit a random-effects model with three fixed effects (intercept, BaseVal, and Drug) and two random intercepts—one patient-level and one hour-level. This is referred to as Model 1 in this example.

Let Fev_{ijk} be the FEV value that is measured in the i th patient at j th hour who is in the (Drug = k) group, where $i = 1, \dots, 24$, $j = 1, \dots, 8$, and $k = \{A, C, P\}$.

$$\text{Fev}_{ijk} = \mu_{ijk} + \gamma_{\text{patient},i} + \gamma_{\text{hour},j} + \epsilon_{ijk}$$

$$\mu_{ijk} = \beta_0 + \beta_1 \cdot \text{baseval}_{ijk} + \beta_2 \cdot (\text{drug}_{ij,k=A}) + \beta_3 \cdot (\text{drug}_{ij,k=C})$$

$$\gamma_{\text{patient},i} \sim N(0, \sigma_{g_p}^2) \quad i = 1, \dots, 24$$

$$\gamma_{\text{hour},j} \sim N(0, \sigma_{g_h}^2) \quad j = 1, \dots, 8$$

$$\epsilon_{ijk} \sim N(0, \sigma_y^2)$$

The following statements fit a model that has two random intercepts—one patient-level and one hour-level:

```
proc bglimm data=Fev nmc=50000 nbi=2000 seed=44672057
  outpost=Fev_mod1 ;
  class Drug Patient Hour;
  model FEV = BaseVal Drug;
  random int / subject=Hour covprior=uniform(lower=0,upper=2);
  random int / subject=Patient;
run;
```

The **MODEL** statement specifies a BaseVal effect and two Drug effects (categorical variables), and the two **RANDOM** statements add two random-intercept effects to the model. The prior for the shrinkage parameter of the Hour effect is taken to be a uniform(0, 2) distribution. The prior for the shrinkage parameter of the Patient effect is the default inverse gamma prior.

Output 28.1.2 show the posterior summary statistics of the fixed-effects parameters and the two random effects. The Scale parameter is the residual variance, Random1 Var is the G-side variance for the Hour effect, and Random2 Var is the G-side variance for the Patient effect.

Output 28.1.2 Posterior Summary Statistics from Model 1

| Posterior Summaries and Intervals | | | | | |
|-----------------------------------|-------|--------|--------------------|------------------|--------|
| Parameter | N | Mean | Standard Deviation | 95% HPD Interval | |
| Intercept | 50000 | 1.4841 | 0.3112 | 0.8766 | 2.0948 |
| Baseval | 50000 | 0.5007 | 0.1028 | 0.2991 | 0.7042 |
| Drug a | 50000 | 0.3237 | 0.0415 | 0.2426 | 0.4046 |
| Drug c | 50000 | 0.5245 | 0.0420 | 0.4427 | 0.6074 |
| Drug p | 0 | . | . | . | . |
| Scale | 50000 | 0.1184 | 0.00857 | 0.1016 | 0.1350 |
| Random1 Var | 50000 | 0.0622 | 0.0704 | 0.0118 | 0.1606 |
| Random2 Var | 50000 | 0.3647 | 0.1108 | 0.1858 | 0.5868 |

You can see that the Drug A effect (versus the placebo group) is 0.32 and the Drug C effect is 0.52. You can also, for example, use the **ESTIMATE** statement as follows to measure the effect difference between the two drug groups (the posterior mean difference is around -0.2 , and the posterior standard deviation is 0.042):

```
estimate "Drug A vs C" drug 1 -1;
```

The second way of modeling these data (Model 2) is to treat the Hour effect as fixed effects, not as a random effect. And the Hour variable enters the main model as a categorical variable, as shown in the following statements:

```
proc bglimm data=Fev nmc=10000 seed=44672057
  outpost=Fev_mod2;
  class Drug Patient Hour;
  model FEV = BaseVal Drug Hour;
  random int / subject=Patient;
  estimate "A vs B" Drug 1 -1;
run;
```

This model has only one random effect (Patient). The posterior summary statistics of the fixed-effects parameters are displayed in Output 28.1.3.

Output 28.1.3 Posterior Summary Statistics from Model 2
The BGLIMM Procedure

| Posterior Summaries and Intervals | | | | | |
|-----------------------------------|-------|--------|-----------------------|---------------------|--------|
| Parameter | N | Mean | Standard Deviation | 95% HPD Interval | |
| Intercept | 10000 | 1.2592 | 0.3090 | 0.6002 | 1.8231 |
| Baseval | 10000 | 0.5031 | 0.1021 | 0.3111 | 0.7078 |
| Drug a | 10000 | 0.3247 | 0.0413 | 0.2475 | 0.4077 |
| Drug c | 10000 | 0.5253 | 0.0419 | 0.4412 | 0.6064 |
| Drug p | 0 | . | . | . | . |
| Hour 1 | 10000 | 0.4261 | 0.0700 | 0.2869 | 0.5612 |
| Hour 2 | 10000 | 0.4779 | 0.0703 | 0.3369 | 0.6130 |
| Hour 3 | 10000 | 0.3679 | 0.0706 | 0.2317 | 0.5090 |
| Hour 4 | 10000 | 0.2562 | 0.0692 | 0.1212 | 0.3926 |
| Hour 5 | 10000 | 0.1094 | 0.0709 | -0.0256 | 0.2529 |
| Hour 6 | 10000 | 0.1140 | 0.0719 | -0.0269 | 0.2540 |
| Hour 7 | 10000 | 0.0312 | 0.0704 | -0.1028 | 0.1749 |
| Hour 8 | 0 | . | . | . | . |
| Scale | 10000 | 0.1185 | 0.00855 | 0.1031 | 0.1364 |
| Random Var | 10000 | 0.3653 | 0.1131 | 0.1815 | 0.5863 |

Although both models account for the Hour effect, they differ in the sense that the Hour random effects in Model 1 have a shared hyperprior distribution with a shrinkage parameter, whereas in Model 2 they are treated as independent fixed effects with a noninformative flat prior. Some statisticians prefer one or the other approach, because they offer different interpretations of the model. For a detailed discussion, see Gelman and Hill (2007, p. 245). The estimated effects in Model 1 and Model 2 are almost identical.

Model 3 is a minor variation of Model 2. You can consider an interaction main effect between Drug and Hour. The change in the SAS program is minor—you add a vertical bar (|) between the two effects to create two main effects and the cross effect between the two, as follows:

```
model FEV = BaseVal Drug|Hour;
```

Results from Model 3 are not displayed.

Models 1 through 3 all treat within-patient measurements as independent observations (the R-side covariance is of the VC type), which do not account for correlations. To model the longitudinal aspect of the measurements across hours, you can specify a nonidentity R-side covariance type, such as an unstructured covariance (Model 4),

$$\text{Fev}_{ik} \sim \text{MVN}(\mu_{ik}, \mathbf{R})$$

where Fev_{ik} denotes all hourly observations from the i th patient who is taking drug k .

You can use the following PROC BGLIMM program to fit a repeated measures model with three fixed-effects (BaseVal, Drug, and Hour), one patient-level random effect, and an unstructured R-side covariance matrix:

```
proc bglimm data=Fev nmc=10000 seed=44672057
  outpost=Fev_mod4;
  class Drug Patient Hour;
  model FEV = BaseVal Drug Hour;
  random int / subject=Patient;
  repeated Hour / subject=Patient(Drug) type=un r rcorr;
run;
```

The **REPEATED** statement specifies that the Hour variable is the repeated measures variable, and an unstructured covariance is specified. The nested effects, **Patient (Drug)**, that are specified in the **SUBJECT=** option indicate that the repeats occur for each drug treatment within a patient ($3 \times 24 = 72$ in total). The **R** and **RCORR** options display the estimate **R** covariance and correlation matrices, respectively.

Output 28.1.4 displays the estimated covariance and correlation matrices.

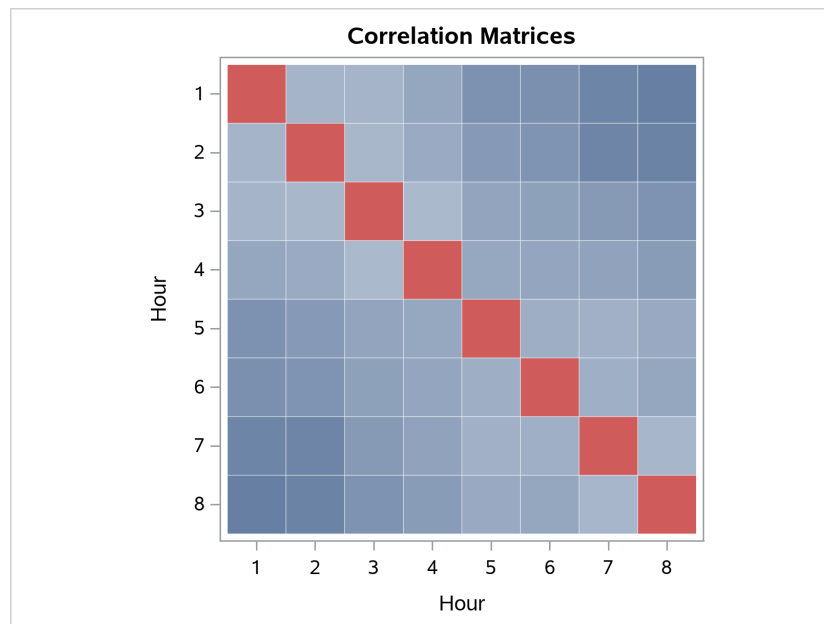
Output 28.1.4 R Covariance and Correlation Matrices

The BGLIMM Procedure

| Estimated R Matrix | | | | | | | | |
|--------------------|----------|---------|---------|---------|---------|---------|---------|----------|
| Row | Col 1 | Col 2 | Col 3 | Col 4 | Col 5 | Col 6 | Col 7 | Col 8 |
| 1 | 0.3098 | 0.06940 | 0.06950 | 0.05183 | 0.03037 | 0.02657 | 0.01326 | 0.005738 |
| 2 | 0.06940 | 0.2996 | 0.07137 | 0.05532 | 0.03923 | 0.03194 | 0.01385 | 0.01094 |
| 3 | 0.06950 | 0.07137 | 0.2981 | 0.07256 | 0.05379 | 0.04759 | 0.04014 | 0.03139 |
| 4 | 0.05183 | 0.05532 | 0.07256 | 0.2933 | 0.05579 | 0.05305 | 0.05022 | 0.04168 |
| 5 | 0.03037 | 0.03923 | 0.05379 | 0.05579 | 0.3446 | 0.06883 | 0.07368 | 0.06412 |
| 6 | 0.02657 | 0.03194 | 0.04759 | 0.05305 | 0.06883 | 0.3314 | 0.07069 | 0.05879 |
| 7 | 0.01326 | 0.01385 | 0.04014 | 0.05022 | 0.07368 | 0.07069 | 0.3497 | 0.08219 |
| 8 | 0.005738 | 0.01094 | 0.03139 | 0.04168 | 0.06412 | 0.05879 | 0.08219 | 0.3462 |

| Estimated R Correlation Matrix | | | | | | | | |
|--------------------------------|---------|---------|---------|--------|---------|---------|---------|---------|
| Row | Col 1 | Col 2 | Col 3 | Col 4 | Col 5 | Col 6 | Col 7 | Col 8 |
| 1 | 1.0000 | 0.2278 | 0.2287 | 0.1720 | 0.09295 | 0.08292 | 0.04029 | 0.01752 |
| 2 | 0.2278 | 1.0000 | 0.2388 | 0.1866 | 0.1221 | 0.1014 | 0.04278 | 0.03399 |
| 3 | 0.2287 | 0.2388 | 1.0000 | 0.2454 | 0.1678 | 0.1514 | 0.1243 | 0.09773 |
| 4 | 0.1720 | 0.1866 | 0.2454 | 1.0000 | 0.1755 | 0.1702 | 0.1568 | 0.1308 |
| 5 | 0.09295 | 0.1221 | 0.1678 | 0.1755 | 1.0000 | 0.2037 | 0.2122 | 0.1856 |
| 6 | 0.08292 | 0.1014 | 0.1514 | 0.1702 | 0.2037 | 1.0000 | 0.2077 | 0.1736 |
| 7 | 0.04029 | 0.04278 | 0.1243 | 0.1568 | 0.2122 | 0.2077 | 1.0000 | 0.2362 |
| 8 | 0.01752 | 0.03399 | 0.09773 | 0.1308 | 0.1856 | 0.1736 | 0.2362 | 1.0000 |

For easier visualization, you can use the correlation matrix to plot a heat map. The graph is displayed in [Output 28.1.5](#). You can see an evenly declining correlation matrix away from the diagonal axis. This indicates stronger correlation with nearby time points, and the correlation becomes weaker as the time lag increases.

Output 28.1.5 Heat Map of the Correlation Matrix in Model 4

Suppose you are interested in estimating the drug response profiles (over time) of a hypothetical patient who has a known BaseVal FEV value. You can use PROC BGLIMM to estimate the predictive distribution of this patient.

The first step is to create the data set Pred, which contains the new covariates information:

```
data Pred;
  input Patient BaseVal Drug$ Hour FEV;
  datalines;
  1 2.46 a 1 .
  1 2.46 a 2 .
  1 2.46 a 3 .
  1 2.46 a 4 .
  1 2.46 a 5 .
  1 2.46 a 6 .
  1 2.46 a 7 .
  1 2.46 a 8 .
  1 2.46 c 1 .
  1 2.46 c 2 .
  1 2.46 c 3 .
  1 2.46 c 4 .
  1 2.46 c 5 .
  1 2.46 c 6 .
  1 2.46 c 7 .
  1 2.46 c 8 .
  1 2.46 p 1 .
  1 2.46 p 2 .
  1 2.46 p 3 .
  1 2.46 p 4 .
  1 2.46 p 5 .
  1 2.46 p 6 .
```

```

1    2.46    p    7    .
1    2.46    p    8    .
;

```

The Patient variable takes the value of 1. As long as the patient value is not a repeat of any existing Patient values in the Fev data set, PROC BGLIMM treats these observations as data from new patients. The BaseVal variable takes a value of 2.46 (the initial FEV value for this patient), the Drug variable takes three treatments, and the Hour variable goes from 1 to 8. The FEV variable takes a missing value, indicating that you want to make predictions in these scenarios.

Next, you concatenate the Pred data set with the original input data set to create a new data set to use in PROC BGLIMM. Model 4 is used here.

```

data Combined;
  set Fev Pred;
run;

proc bglimm data=Combined nmc=10000 seed=44672057
  outpost=Fev_mod4_pred;
  class Drug Patient Hour;
  model FEV = BaseVal Drug|Hour;
  random int / subject=Patient;
  repeated Hour / subject=Patient (Drug) type=un;
run;

```

PROC BGLIMM treats the 24 observations that have missing FEV values as missing data and generates predictive samples for them. The missing data variables have names that follow the convention of FEV_*n*, where *n* is the observation index of these values in the input data set. In this example, the 24 variables are named FEV_416 to FEV_439. There are 415 observation in the Fev data set.

Next, you use the [%SUMINT](#) autocall macro to compute the posterior summary statistics for the predicted values as follows. You can plot the predicted response profiles of this patient for different treatment over time. The results are shown in [Output 28.1.6](#).

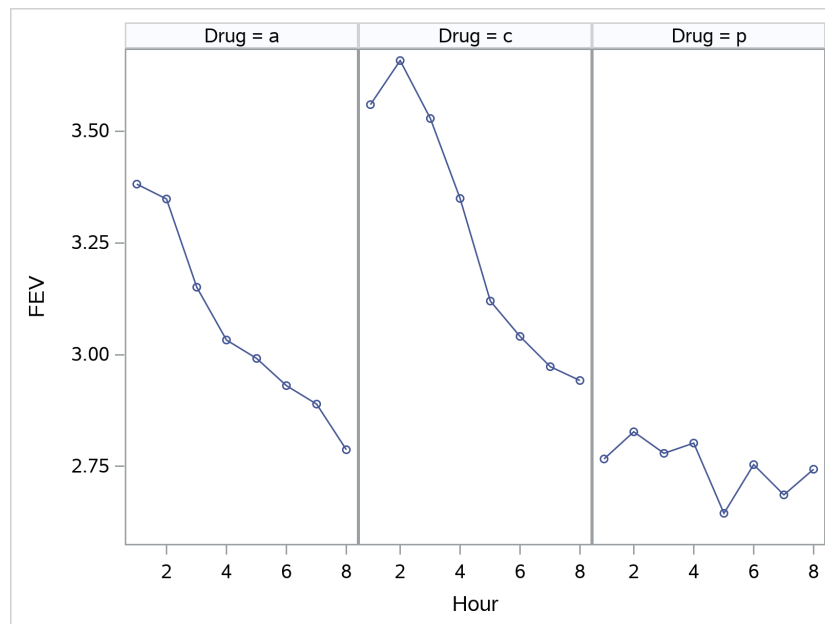
```

%sumint(data=Fev_mod4_pred, var=FEV_:, print=NO, out=pred_si)

data Comb_pred;
  merge Pred Pred_si;
run;

proc sgpanel data=Comb_pred noautolegend;
  panelby Drug / columns=3 ;
  rowaxis label = "FEV";
  scatter x=Hour y=mean / group=Patient;
  series x=Hour y=mean / group=Patient;
run;

```

Output 28.1.6 Predicted Patient Profiles over Time for Different Treatments

The first two panels in [Output 28.1.6](#) both show an initial improvement, then a gradual decline of the FEV values over time. Drug C appears to have a better treatment effect than drug A. The third panel is the placebo group. It has a horizontal profile, which is what you would expect.

Example 28.2: Mating Experiment with Crossed Random Effects

This example illustrates how to use PROC BGLIMM to fit a non-nested logistic random-effects model and perform prediction-based inference.

McCullagh and Nelder (1989, Ch. 14.5) describe a mating experiment—conducted by S. Arnold and P. Verell at the University of Chicago—that involves two geographically isolated populations of mountain dusky salamanders. One goal of the experiment was to determine whether barriers to interbreeding have evolved as a result 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, which was conducted in the summer of 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 contained five salamanders. Each set was mated against one rough butt and one whiteside set, creating eight crossings. Within the pairings of sets, each female salamander was paired with three male salamanders.

The following DATA step creates the data set, *Salamander*, 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

... more lines ...

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 in the first line, 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 the pair mated. On the same day rough butt female 7 was paired with whiteside male 8, but the pair did not mate (the first observation in the fourth line).

The model that is adopted by many authors for these data contains fixed effects for gender and population. Let p_{RR} , p_{RW} , p_{WR} , and p_{WW} denote the mating probabilities between the populations, where the first subscript identifies the female partner and the second subscript identifies the male partner. Then, you model

$$\log \left\{ \frac{p_{kl}}{1 - p_{kl}} \right\} = \beta_{kl} \quad k, l \in \{R, W\}$$

where β_{kl} denotes the average logit of mating between females of population k and males of population l .

The following statements fit a generalized model that contains fixed effects:

```

proc bglimm data=Salamander seed=725697;
  class Fpop Fnum Mpop Mnum;
  model Mating (event='1') = Fpop|Mpope / dist=binary;
run;

```

The response variable is the two-level variable `Mating`. Because it is coded as zeros and ones, and because by default PROC BGLIMM models the probability of the first level according to the response-level ordering, the `EVENT='1'` option instructs PROC BGLIMM to model the probability of a successful mating. The distribution of the mating variable is binary.

The first table in the output is the “Model Information” table, shown in [Output 28.2.1](#). This table displays basic information about the data and model, such as the name of the input data set, response variable, distribution, link function, sampling algorithm, burn-in size, simulation size, thinning number, and random number seed. The random number seed initializes the random number generators. If you repeat the analysis and use the same seed, you get an identical stream of random numbers.

Output 28.2.1 Model Information**The BGLIMM Procedure**

| Model Information | |
|-------------------------|-----------------|
| Data Set | WORK.SALAMANDER |
| Response Variable | Mating |
| Distribution | Binary |
| Link Function | Logit |
| Fixed Effects Included | Yes |
| Random Effects Included | No |
| Sampling Algorithm | Gamerman |
| Burn-In Size | 500 |
| Simulation Size | 5000 |
| Thinning | 1 |
| Random Number Seed | 725697 |
| Number of Threads | 1 |

The “Class Level Information” table in [Output 28.2.2](#) displays the levels of variables that are listed in the **CLASS** statement. Note that there are two female populations and two male populations.

Output 28.2.2 Class Level Information

| Class Level Information | | |
|-------------------------|--------|----------------------|
| Class | Levels | Values |
| Fpop | 2 | rb ws |
| Fnum | 10 | 1 2 3 4 5 6 7 8 9 10 |
| Mpop | 2 | rb ws |
| Mnum | 10 | 1 2 3 4 5 6 7 8 9 10 |

[Output 28.2.3](#) displays the “Number of Observations” table. All 120 observations in the data set are used in the analysis. For data sets that have missing or invalid values, the number of observations that are used might be less than the number of observations that are read.

Output 28.2.3 Number of Observations

| Number of Observations | |
|-----------------------------|-----|
| Number of Observations Read | 120 |
| Number of Observations Used | 120 |

The “Response Profile” table, which is displayed only for binary data, lists the levels of the response variable and their order ([Output 28.2.4](#)). 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 that is being modeled is that of the higher-ordered value.

Output 28.2.4 Response Profile

| Response Profile | | |
|------------------|--------|-----------------|
| Ordered Value | Mating | Total Frequency |
| 1 | 0 | 50 |
| 2 | 1 | 70 |

Probability modeled is Mating = '1'.

PROC BGLIMM reports posterior summary statistics (posterior means, standard deviations, and highest posterior density (HPD) intervals) for each parameter in the “Posterior Summaries and Intervals” table, as shown in [Output 28.2.5](#).

Output 28.2.5 Posterior Summary Statistics

| Posterior Summaries and Intervals | | | | | |
|-----------------------------------|------|---------|--------------------|------------------|---------|
| Parameter | N | Mean | Standard Deviation | 95% HPD Interval | |
| Intercept | 5000 | 0.8845 | 0.4082 | 0.1391 | 1.7387 |
| Fpop rb | 5000 | -0.1674 | 0.5659 | -1.2915 | 0.8939 |
| Fpop ws | 0 | . | . | . | . |
| Mpop rb | 5000 | -2.1235 | 0.5964 | -3.3360 | -1.0176 |
| Mpop ws | 0 | . | . | . | . |
| Fpop rb*Mpop rb | 5000 | 2.4627 | 0.8352 | 0.8322 | 4.0574 |
| Fpop rb*Mpop ws | 0 | . | . | . | . |
| Fpop ws*Mpop rb | 0 | . | . | . | . |
| Fpop ws*Mpop ws | 0 | . | . | . | . |

There are four levels in the Fpop*Mpop effect. An intercept is included by default in the fixed effects, so the last level of the Fpop*Mpop effect (pairing of a whiteside female and a whiteside male) is not estimable (this explains why the estimate for the last level is missing). The estimate for the intercept gives the logit for the last level of Fpop*Mpop effect—that is, the pairing of a whiteside female and a whiteside male. And the estimates for other three pairings are the deviates from the logit for the pairing of a whiteside female and a whiteside male. For example, the posterior mean of the estimate for “Fpop rb*Mpop rb” is 0.16, meaning that the logit for the pairing of a rough butt female and a rough butt male is 0.16 higher than the logit for the pairing of a whiteside female and a whiteside male.

In a pairing with a female whiteside salamander, the logit drops sharply by 2.12 when the female is paired with a whiteside male from a different population instead of a male from her own population. If the same comparisons are made in pairs with whiteside males, then you also notice a drop of 0.17 in the logit if the female comes from a different population from that of the male. However, the change is considerably less.

This model can be fitted using a logistic regression with two random effects:

$$p_i = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 \cdot \text{Fpop}_{\text{rb}} \text{Mpop}_{\text{ws}} + \beta_2 \cdot \text{Fpop}_{\text{ws}} \text{Mpop}_{\text{rb}} + \beta_3 \cdot \text{Fpop}_{\text{rb}} \text{Mpop}_{\text{rb}} + \gamma_f + \gamma_m))}$$

$$\gamma_f \sim N(0, \sigma_f^2)$$

$$\gamma_m \sim N(0, \sigma_m^2)$$

This represents a logistic regression model that has a fixed intercept, an interaction, one random effect for the female group (for $f = 1, \dots, 20$), and one random effect for the male group (for $m = 1, \dots, 20$). You can use the following statements to fit the model:

```
proc bglimm data=salamander nmc=20000 seed=901214;
  class Fpop Fnum Mpop Mnum;
  model Mating (event='1') = Fpop|Mpop / dist=binary;
  random int / sub=Fpop*Fnum;
  random int / sub=Mpop*Mnum;
run;
```

The $Fpop*Fnum$ effect in the first **RANDOM** statement creates a random intercept for each female animal. Because *Fpop* and *Fnum* are **CLASS** variables, the effect has 20 levels (10 rough butt and 10 whiteside females). Similarly, the $Mpop*Mnum$ effect in the second **RANDOM** statement creates the random intercepts for the male animals. No **TYPE=** option is specified in the **RANDOM** statement, so the covariance structure defaults to **TYPE=VC**. The random effects and their levels are independent, and each effect has its own variance component. Because the conditional distribution of the data, conditioned on the random effects, is binary, no extra scale parameter (ϕ) is added. The **RANDOM** statements in **PROC BGLIMM** take two interaction effects: $Fpop*Fnum$ and $Mpop*Mnum$. The **SUB=** option in the **RANDOM** statement in **PROC BGLIMM** supports a syntax for interaction effects.

The “Posterior Summaries and Intervals” table is shown in [Output 28.2.6](#).

Output 28.2.6 Posterior Summary Statistics

The BGLIMM Procedure

| Posterior Summaries and Intervals | | | | | |
|-----------------------------------|-------|---------|-----------------------|---------------------|---------|
| Parameter | N | Mean | Standard Deviation | 95% HPD Interval | |
| Intercept | 20000 | 1.3187 | 0.7770 | -0.2395 | 2.8228 |
| Fpop rb | 20000 | -0.2836 | 0.9660 | -2.1719 | 1.6527 |
| Fpop ws | 0 | . | . | . | . |
| Mpop rb | 20000 | -3.1839 | 0.9695 | -5.1256 | -1.3199 |
| Mpop ws | 0 | . | . | . | . |
| Fpop rb*Mpop rb | 20000 | 3.7212 | 1.1306 | 1.4989 | 5.9243 |
| Fpop rb*Mpop ws | 0 | . | . | . | . |
| Fpop ws*Mpop rb | 0 | . | . | . | . |
| Fpop ws*Mpop ws | 0 | . | . | . | . |
| Random1 Var | 20000 | 2.1019 | 1.3387 | 0.3421 | 4.6481 |
| Random2 Var | 20000 | 1.0683 | 0.6602 | 0.2376 | 2.3319 |

The parameter labeled “Random1 Var” measures the variability among the 20 females (10 rough butt and 10 whiteside); the parameter labeled “Random2 Var” measures the variability among the 20 males (10 rough butt and 10 whiteside). It seems that there is more variability among the females than among the males.

In the previous program, two **RANDOM** statements are used to specify two random intercepts. Alternatively, you can combine the two **SUBJECT=** variables into a single **RANDOM** statement, as follows:

```
random Fpop*Fnum Mpop*Mnum;
```

This is an equivalent model specification. However, the posterior sampling sequence will be different: in programs that use the **SUBJECT=** option, the random-effects parameters are updated in sequence; if you omit the **SUBJECT=** option, all random effects from that **RANDOM** statement are updated together. Therefore, the posterior samples will be different, and you can also experience mixing and performance differences, depending on the model and the number of random-effects parameters involved.

The following **ESTIMATE** statements compute the four levels of the Fpop*Mpop effect, which are labeled as "rb and rb" and so on:

```
proc bglimm data=salamander nmc=20000 seed=901214 outpost=Sal_Out;
  class Fpop Fnum Mpop Mnum;
  model Mating (event='1') = Fpop|Mpop / dist=binary;
  random int / sub=Fpop*Fnum;
  random int / sub=Mpop*Mnum;
  estimate "rb and rb" Int 1 Fpop 1 0 Mpop 1 0 Fpop*Mpop 1;
  estimate "rb and ws" Int 1 Fpop 1 0 Mpop 0 1 Fpop*Mpop 0 1;
  estimate "ws and rb" Int 1 Fpop 0 1 Mpop 1 0 Fpop*Mpop 0 0 1;
  estimate "ws and ws" Int 1 Fpop 0 1 Mpop 0 1 Fpop*Mpop 0 0 0 1;
run;
```

These are some linear combinations of the fixed-effects parameters, which, if you choose not to use the **ESTIMATE** statement, you replicate by using the following DATA step manipulations:

```
data Post;
  set Sal_Out;
  rr = Intercept + Fpop_rb + Mpop_rb + Fpop_rb_Mpop_rb;
  rw = Intercept + Fpop_rb;
  wr = Intercept + Mpop_rb;
  ww = Intercept;
  drop Intercept__;;
run;
```

The RR variable in the Post data set is identical to what the first **ESTIMATE** statement produces. You can check by comparing RR with rb_and_rb, which was created by the first **ESTIMATE** statement in the earlier program.

What is more relevant is the need to transfer these linear combinations back to the scale of the data and to estimate different mating probabilities: p_{RR} , p_{RW} , p_{WR} , and p_{WW} . If the fitted model is a fixed-effects model, as in

$$\log \left\{ \frac{p_{kl}}{1 - p_{kl}} \right\} = \beta_{kl} \quad k, l \in \{R, W\}$$

then the transformation back to p is simple:

$$p_{kl} = \frac{\exp(\beta_{kl})}{1 + \exp(\beta_{kl})}$$

And the programming is straightforward:

```

data Prob;
  set Post;
  p_rr_f = logistic(rr);
  p_rw_f = logistic(rw);
  p_wr_f = logistic(wr);
  p_ww_f = logistic(ww);
run;

```

But because the fitted model is a random-effects model,

$$\log \left\{ \frac{p_{kl}}{1 - p_{kl}} \right\} = \beta_{kl} + \gamma_f + \gamma_m \quad k, l \in \{R, W\}$$

The transformation must account for the variability in the random effects. In other words, you must integrate out the uncertainties of the γ_f and γ_m in order to obtain the marginal posterior distribution of the p_{kl} parameters:

$$\pi(p_{kl}|\text{data}) = \int_{\gamma_f} \int_{\gamma_m} \pi(p_{kl}, \gamma_f, \gamma_m|\text{data}) \cdot \pi(\gamma_f) \cdot \pi(\gamma_m) d\gamma_m d\gamma_f$$

Most frequently and most conveniently, $\pi(\gamma_f)$ and $\pi(\gamma_m)$ are taken to be the distribution of the random effects, given the posterior samples of σ_f^2 and σ_m^2 .

The following DATA step program uses the Monte Carlo method to integrate out the random effects and compute the marginal posterior of the mating probabilities:

```

data Prob;
  set Post;
  call streaminit(87925);
  g_f = rand("normal", 0, sqrt(random1_var));
  g_m = rand("normal", 0, sqrt(random2_var));
  p_rr = logistic(rr + g_f + g_m);
  g_f = rand("normal", 0, sqrt(random1_var));
  g_m = rand("normal", 0, sqrt(random2_var));
  p_rw = logistic(rw + g_f + g_m);
  g_f = rand("normal", 0, sqrt(random1_var));
  g_m = rand("normal", 0, sqrt(random2_var));
  p_wr = logistic(wr + g_f + g_m);
  g_f = rand("normal", 0, sqrt(random1_var));
  g_m = rand("normal", 0, sqrt(random2_var));
  p_ww = logistic(ww + g_f + g_m);
  drop g_f g_m;
run;

```

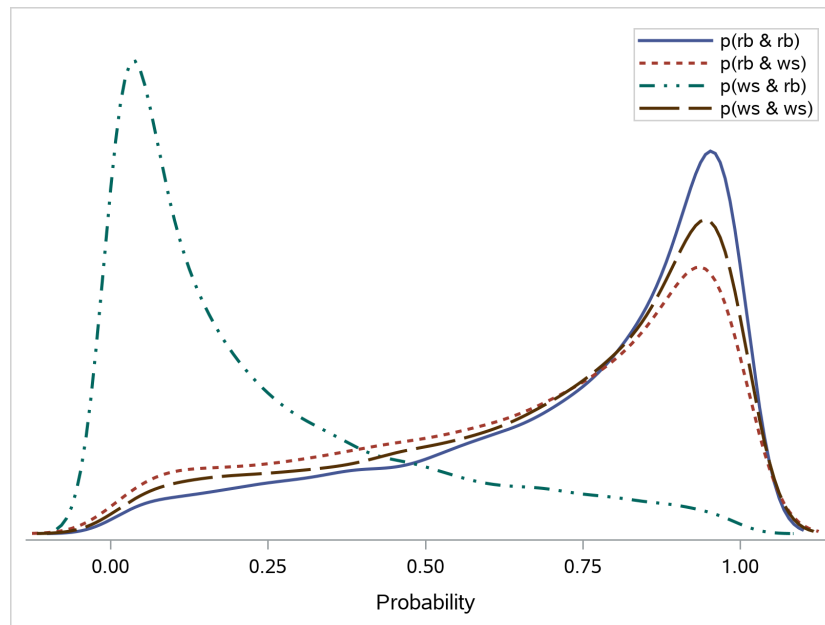
The CALL STREAMINIT statement ensures reproducibility; the RAND call draws samples from the random-effects prior distribution, conditional on posterior samples of σ_f^2 and σ_m^2 . You compute the mean and apply the logistic transformation. This process is repeated for the computation of each individual probability. The `g_f` and `g_m` are temporary symbols that do not need to be saved after the sampling and computation are done.

The Prob data set contains 20,000 draws of `p_rr`, `p_rw`, `p_wr`, and `p_ww`, which are estimates of the four mating probabilities, respectively. You can use the following program to display the posterior distributions of the four mating probabilities:

```
proc sgplot data=Prob noborder;
  density p_rr / type=kernel legendlabel='p(rb & rb) '
    lineattrs=(pattern=solid);
  density p_rw / type=kernel legendlabel='p(rb & ws) '
    lineattrs=(pattern=ShortDash);
  density p_wr / type=kernel legendlabel='p(ws & rb) '
    lineattrs=(pattern=DashDotDot);
  density p_ww / type=kernel legendlabel='p(ws & ws) '
    lineattrs=(pattern=LongDash);
  keylegend / location=inside position=topright across=1;
  xaxis label="Probability";
  yaxis display=(nolabel noline noticks novalues);
run;
```

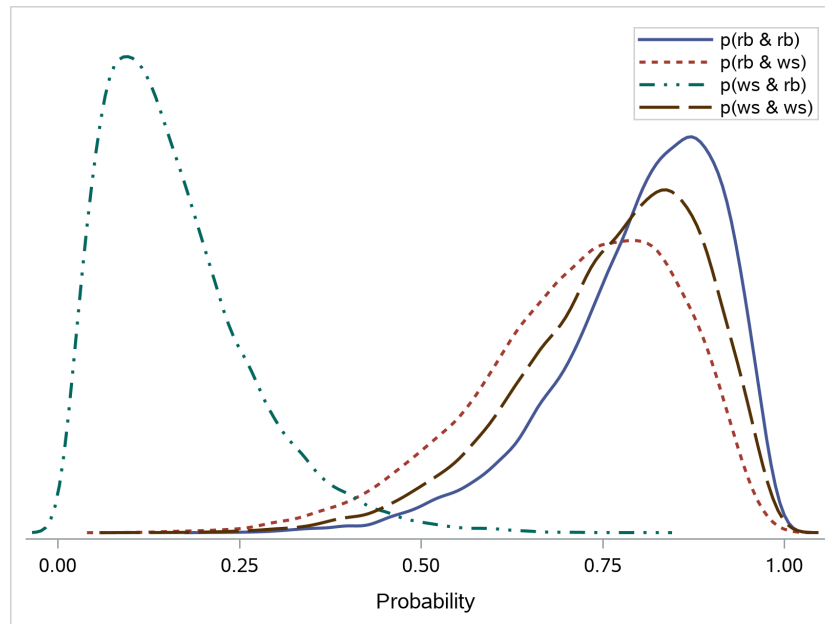
The posterior distributions of the mating probabilities are shown in [Output 28.2.7](#). The probability of a female whiteside mating with a male rough butt (the dash-dotted line) is significantly lower than the other three probabilities. The smoothness at the boundaries (probability less than 0 or greater than 1) is an artifact of the kernel method that is used in the SGLOT procedure.

Output 28.2.7 Marginal Posterior Distributions of Mating Probability



Considering the posterior distributions of these probabilities, you can make additional inferences rather easily. For example, you can estimate that the probability of a female rough butt mating with a male rough butt is greater than the probability of a female whiteside mating with a male whiteside.

To compare, [Output 28.2.8](#) shows the posterior distributions of the mating probabilities by using logistic transformation on the fixed-effects parameters only, without accounting for the variability of the random effects. These distributions are clearly different from those shown in [Output 28.2.7](#), with noticeably thinner tails, precisely because they do not account for additional variability in the model.

Output 28.2.8 Posterior Distributions of Mating Probability without Accounting for Random Effects

Example 28.3: Poisson Regression with Random Effects

This example illustrates how to use PROC BGLIMM to fit a Poisson random-effects model with offsets. The example also discusses the use of the deviance information criterion (DIC) as a way to evaluate the fit of a model.

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 (SMRs).

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, λ_i , that is specific to each county and is a random variable. Conditional on λ_i , the observed counts are independent Poisson variables with mean $E_i \lambda_i$.

An elementary mixed model for λ_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 the population who work in agriculture, fishing, and forestry and therefore spend a lot of time exposed to sunlight. The expanded model for the region-specific relative risk in Breslow and Clayton (1993) is

$$\lambda_i = \exp \{ \beta_0 + \beta_1 \text{Employment}_i / 10 + \gamma_i \}, \quad i = 1, \dots, 56$$

where β_0 and β_1 are fixed effects and γ_i are county random effects.

Because the mean of the Poisson variates, conditional on the random effects, is $\mu_i = E_i \lambda_i$, applying a log link yields

$$\log \{ \mu_i \} = \log \{ E_i \} + \beta_0 + \beta_1 \text{Employment}_i / 10 + \gamma_i$$

The term $\log\{E_i\}$ is an offset, a regressor variable whose coefficient is known to be 1. Note that it is assumed that the E_i are known; they are not treated as random variables.

The following DATA step creates the data set LipCancer:

```
data LipCancer;
  input County Observed Expected Employment SMR;
  if (Observed > 0) then ExpCount = 100*Observed/SMR;
  else ExpCount = Expected;
  x = Employment / 10;
  LogN = log(ExpCount);
  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

... more lines ...

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
;
```

The expected number of cases, ExpCount, is based on the observed standardized mortality ratio for counties that have lip cancer cases and based on the expected counts that are reported by Clayton and Kaldor (1987, Table 1) for the counties without such cases. The sum of the expected counts then equals the sum of the observed counts. The offset is created in the DATA step by using the following assignment statement:

```
LogN = log(ExpCount);
```

In addition, in the DATA step, you use the following statement to transform the covariate that measures the percentage of employment in agriculture, fisheries, and forestry to agree with the analysis of Breslow and Clayton (1993):

```
x = Employment / 10;
```

The following statements fit this Poisson model with an offset and a random intercept for each county:

```
proc bglimm data=LipCancer seed=10571042 nmc=10000
  outpost=LipCancer_Out;
  class County;
  model Observed = x / dist=poisson offset=LogN;
  random int / sub=County;
run;
```

In the **MODEL** statement, **DIST=POISSON** specifies that the response variable has a Poisson distribution with an offset, LogN. The offset is associated with the linear predictor through the **OFFSET=LOGN** option in the **MODEL** statement.

Along with the **MODEL** statement for fixed effects, the **RANDOM** statement is used for random effects. It specifies that the linear predictor contains an intercept term that randomly varies at the level of the County effect. The **SUBJECT=COUNTY** option in the **RANDOM** statement defines County as a subject index for the random-effects grouping, so that each county has its own intercept.

The “Model Information” table, the “Class Level Information” table, and the “Number of Observations” table are shown in [Output 28.3.1](#). Note that there are 56 counties.

Output 28.3.1 Model Information
The BGLIMM Procedure

| Model Information | |
|-------------------------|---------------------|
| Data Set | WORK.LIPCANCER |
| Response Variable | Observed |
| Distribution | Poisson |
| Link Function | Log |
| Fixed Effects Included | Yes |
| Random Effects Included | Yes |
| Sampling Algorithm | Gamerman, Conjugate |
| Burn-In Size | 500 |
| Simulation Size | 10000 |
| Thinning | 1 |
| Random Number Seed | 10571042 |
| Number of Threads | 1 |

| 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 | |
|-----------------------------|----|
| Number of Observations Read | 56 |
| Number of Observations Used | 56 |

The “Posterior Summaries and Intervals” table in [Output 28.3.2](#) displays the estimates of β_0 and β_1 along with their standard deviations. The covariate that measures employment percentage in agriculture, fisheries, and forestry does not have zero in the 95% HPD interval, indicating that this covariate might be a surrogate for the exposure to sunlight, an important risk factor for lip cancer.

Output 28.3.2 Posterior Summaries and Intervals

| Posterior Summaries and Intervals | | | | | |
|-----------------------------------|-------|---------|--------------------|------------------|---------|
| Parameter | N | Mean | Standard Deviation | 95% HPD Interval | |
| Intercept | 10000 | -0.5069 | 0.1712 | -0.8502 | -0.1835 |
| x | 10000 | 0.6900 | 0.1585 | 0.3785 | 0.9989 |
| Random Var | 10000 | 0.4763 | 0.1217 | 0.2702 | 0.7213 |

Looking at the estimate of the variance of the region-specific log-relative risks, you can see 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.)

You can use the posterior data set `LipCancer_Out` to compute the predicted SMR value, which is

$$\text{SMR}_i = 100 \cdot \exp(\beta_0 + \beta_1 x_i + \gamma_i)$$

where i denotes the county, β_0 is the intercept, β_1 is the parameter for the covariate x , and γ_i is the random effect for county i . The following DATA step computes the predicted SMR for each county and saves them in the data set `SMR_Pred`:

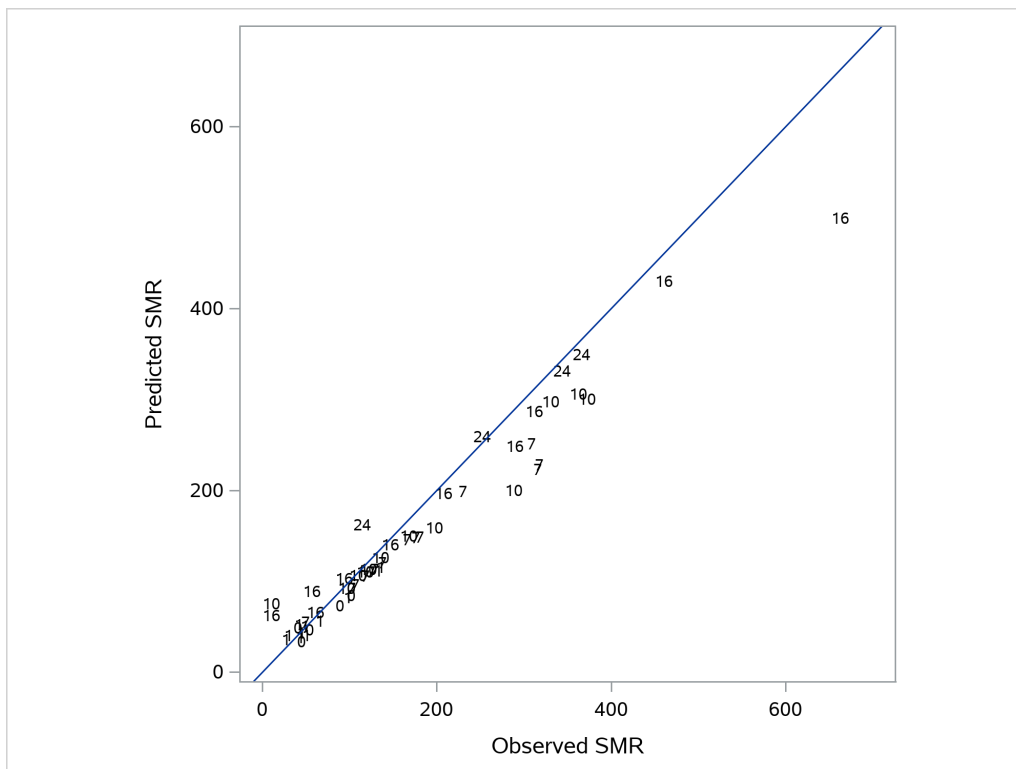
```
data SMR_PRED;
  array gamma[56] Intercept__County_1-Intercept__County_56;
  array SMR_pred[56];
  set LipCancer_Out;
  do i = 1 to 56;
    set LipCancer(rename=(x=data_x)) point=i;
    SMR_pred[i] = 100 * exp(Intercept + x * data_x + gamma[i]);
  end;
  keep smr_pred;;
run;
```

You can compute the mean estimates for each `SMR_Pred` and plot them against the observed SMR values ([Output 28.3.3](#)) to get a sense of how the model fits the data, as follows:

```
%sumint(data=SMR_PRED, var=_numeric_, print=NO, out=SMR_SI)

data combine;
  merge LipCancer SMR_SI;
run;

proc sgplot data=combine noautolegend aspect=1;
  yaxis label="Predicted SMR" max=700;
  xaxis label="Observed SMR" max=700;
  text x=SMR y=mean text=employment;
  lineparm x=0 y=0 slope=1;
run;
```

Output 28.3.3 Observed and Predicted SMRs; Data Labels Indicate Employment Types

In contrast, you can fit a fixed-effects-only model to the data by using the following program (results not shown):

```
proc bglimm data=LipCancer seed=10571042 nmc=10000
  outpost=LipCancer_fOut;
  class County;
  model Observed = x / dist=poisson offset=LogN;
run;
```

The fixed-effects model provides a worse fit than the random-effects model ([Output 28.3.4](#)).

Output 28.3.6 DIC Values from Fixed-Effects-Only Model**The BGLIMM Procedure**

| Deviance Information Criterion | |
|----------------------------------------------|---------|
| Dbar (Posterior Mean of Deviance) | 449.038 |
| Dmean (Deviance Evaluated at Posterior Mean) | 447.023 |
| pD (Effective Number of Parameters) | 2.015 |
| DIC (Smaller is Better) | 451.053 |

You can further extend the model by including another random effect, the Employment-level random-intercept effect that accounts for employment-related variability, as follows:

```
proc bglimm data=LipCancer seed=10571042 nmc=10000;
  class County Employment;
  model Observed = x / dist=poisson offset=LogN;
  random int / sub=Employment;
  random int / sub=County;
run;
```

This type of overparameterized model can often encounter convergence difficulties and might not provide any more meaningful fitting than County-level random-effects models. Note that PROC BGLIMM produces an error message if the two **RANDOM** statements are switched; the procedure requires that the observation-level random effects be specified last.

Example 28.4: Repeated Growth Measurements with Group Difference

This example illustrates how to model heterogeneity in covariance structures by using grouped growth measurements data.

Changes in the distance (measured in mm) from the center of the pituitary gland to the pterygomaxillary fissure near the teeth are important in orthodontic therapy. The dental study of Pothoff and Roy (1964) consists of growth measurements of 27 children (11 girls and 16 boys) at ages 8, 10, 12, and 14.

The following DATA step creates the data set, PR, for the analysis:

```
data pr;
  input Person Gender $ y1 y2 y3 y4;
  Distance=y1; Age=8; Time=1; output;
  Distance=y2; Age=10; Time=2; output;
  Distance=y3; Age=12; Time=3; output;
  Distance=y4; Age=14; Time=4; output;
  drop y1-y4;
  datalines;
1 F 21.0 20.0 21.5 23.0
2 F 21.0 21.5 24.0 25.5
3 F 20.5 24.0 24.5 26.0
4 F 23.5 24.5 25.0 26.5
5 F 21.5 23.0 22.5 23.5
6 F 20.0 21.0 21.0 22.5
7 F 21.5 22.5 23.0 25.0
8 F 23.0 23.0 23.5 24.0
```

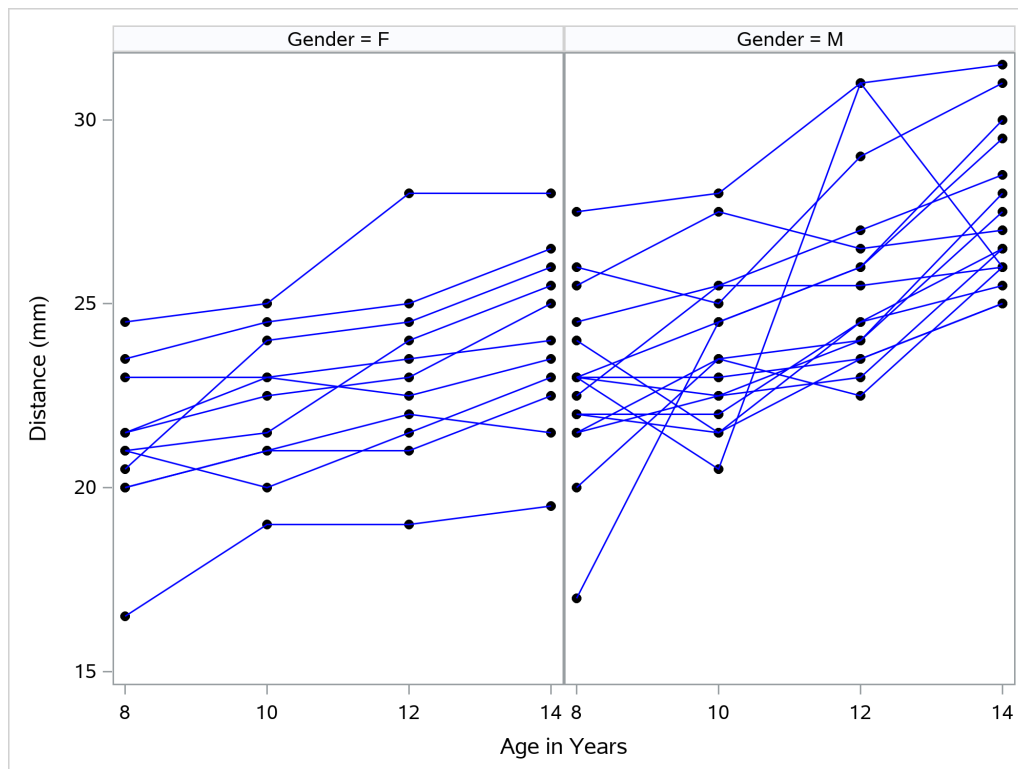
| | | | | | |
|----|---|------|------|------|------|
| 9 | F | 20.0 | 21.0 | 22.0 | 21.5 |
| 10 | F | 16.5 | 19.0 | 19.0 | 19.5 |
| 11 | F | 24.5 | 25.0 | 28.0 | 28.0 |
| 12 | M | 26.0 | 25.0 | 29.0 | 31.0 |
| 13 | M | 21.5 | 22.5 | 23.0 | 26.5 |
| 14 | M | 23.0 | 22.5 | 24.0 | 27.5 |
| 15 | M | 25.5 | 27.5 | 26.5 | 27.0 |
| 16 | M | 20.0 | 23.5 | 22.5 | 26.0 |
| 17 | M | 24.5 | 25.5 | 27.0 | 28.5 |
| 18 | M | 22.0 | 22.0 | 24.5 | 26.5 |
| 19 | M | 24.0 | 21.5 | 24.5 | 25.5 |
| 20 | M | 23.0 | 20.5 | 31.0 | 26.0 |
| 21 | M | 27.5 | 28.0 | 31.0 | 31.5 |
| 22 | M | 23.0 | 23.0 | 23.5 | 25.0 |
| 23 | M | 21.5 | 23.5 | 24.0 | 28.0 |
| 24 | M | 17.0 | 24.5 | 26.0 | 29.5 |
| 25 | M | 22.5 | 25.5 | 25.5 | 26.0 |
| 26 | M | 23.0 | 24.5 | 26.0 | 30.0 |
| 27 | M | 22.0 | 21.5 | 23.5 | 25.0 |

;

Questions of interest include the following:

- Does the distance from the center of the pituitary gland to the pterygomaxillary fissure change over time?
- What is the pattern of change?
- Does the pattern of change differ between boys and girls?
- How correlated are the repeated measurements across age for the same subject?
- Is the correlation similar for boys and girls?

Output 28.4.1 shows the distance growth plots for boys and girls as they age. The plots show apparent tracking (for children who start with small distance, the distances tend to stay small) and approximately linear growth. However, the growth rates are noticeably different for boys (faster) and girls (slower).

Output 28.4.1 Distance Growth Plot for Dental Data

You can fit a linear repeated measures model to the data: the **Age** variable is the fixed-effect covariate, and the repeats are over the variable **Time**. The **BY** statement offers a convenient way to perform a separate and independent analysis between the boys' group and the girls' group. This accounts for different growth profiles for the two groups, because correlation can differ by gender.

The following PROC BGLIMM program fits two linear repeated measures models to the data:

```
proc bglimm data=pr seed=475193 outpost=pr_out;
  by Gender;
  class Person Time;
  model Distance = Age;
  repeated Time / subject=Person type=un r;
run;
```

The **MODEL** statement specifies the response variable **Distance** and the covariate **Age**, which models an overall linear growth trend over time.

The **REPEATED** statement specifies that the repeated measurements be taken over the **Time** variable. The repeated effect is required in a **REPEATED** statement, and it must be specified as a **CLASS** variable. The **TYPE=UN** option specifies an unstructured covariance for the residuals. The repeated measurements are grouped according to **Person** (the **SUBJECT=** variable), and the covariance type is unstructured. The **R** option displays the estimated covariance matrix of **R**.

Output 28.4.2 and **Output 28.4.3** show the estimate covariance matrix for girls and boys, respectively. The point estimates are different, and you can get a more accurate sense of the degree of differences by plotting the posterior density estimates. This is shown in **Output 28.4.4**.

Output 28.4.2 Estimated Covariance Matrix of **R** for Girls
The BGLIMM Procedure

Gender=F

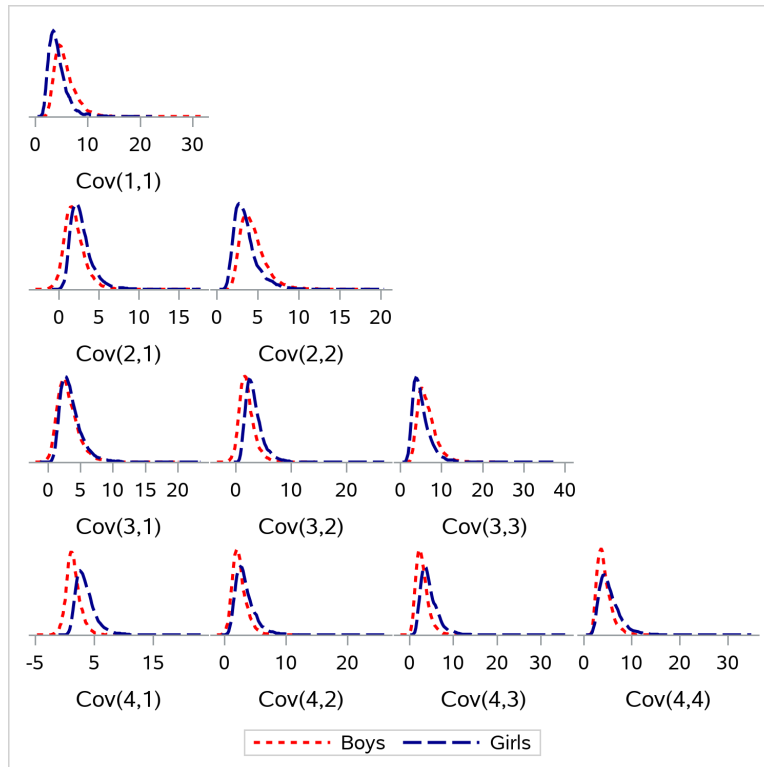
| Estimated R Matrix | | | | |
|--------------------|--------|--------|--------|--------|
| Row | Col 1 | Col 2 | Col 3 | Col 4 |
| 1 | 4.2652 | 2.8200 | 3.5671 | 3.5578 |
| 2 | 2.8200 | 3.5672 | 3.3555 | 3.3955 |
| 3 | 3.5671 | 3.3555 | 5.1359 | 4.5170 |
| 4 | 3.5578 | 3.3955 | 4.5170 | 5.4541 |

Output 28.4.3 Estimated Covariance Matrix of **R** for Boys

Gender=M

| Estimated R Matrix | | | | |
|--------------------|--------|--------|--------|--------|
| Row | Col 1 | Col 2 | Col 3 | Col 4 |
| 1 | 5.6814 | 1.9663 | 3.1060 | 1.4269 |
| 2 | 1.9663 | 4.4751 | 2.0155 | 2.4799 |
| 3 | 3.1060 | 2.0155 | 6.4496 | 2.8814 |
| 4 | 1.4269 | 2.4799 | 2.8814 | 4.2419 |

Output 28.4.4 Posterior Density Comparison of the **R** Covariance—Independent Analysis



The independent analysis accounts for the heterogeneity in the data (different groups have different covariance structure), but the model cannot represent, for example, the Gender effect, after you separate the data set by Gender. Ideally, you want both to account for the heterogeneity and to fit a model that estimates the Gender effect,

$$\mathbf{y}_i \sim \text{MVN}(\boldsymbol{\mu}_i, \mathbf{R}_k)$$

where \mathbf{y}_i is the vector of observed measurements from the i th subject; $\boldsymbol{\mu}_i$ is the regression mean that has the Gender effect; and $k = \text{F, M}$, indicating that there are two distinct covariance structures for the two gender groups.

You can fit this model by using the following statements:

```
proc bglimm data=pr seed=475193 outpost=pr_out;
  class Person Gender Time;
  model Distance = Age|Gender;
  repeated Time / type=un subject=Person group=Gender r;
run;
```

The **MODEL** statement specifies the interaction main effect between Age and Gender, by using a shorthand notation (the vertical bar) for including Age, Gender, and Age*Gender. **GROUP=GENDER** (in the **REPEATED** statement) models a separate R-side covariance matrix for each gender. The **R** option prints the estimated covariance matrix of **R**.

The results are shown in [Output 28.4.5](#) and [Output 28.4.6](#).

Output 28.4.5 Posterior Summaries and Intervals**The BGLIMM Procedure**

| Posterior Summaries and Intervals | | | | | | |
|-----------------------------------|----------|------|---------|-----------------------|---------------------|---------|
| Parameter | Group | N | Mean | Standard Deviation | 95% HPD Interval | |
| Intercept | | 5000 | 15.9070 | 1.1413 | 13.7924 | 18.2695 |
| Age | | 5000 | 0.8254 | 0.0962 | 0.6374 | 1.0135 |
| Gender F | | 5000 | 1.4767 | 1.4106 | -1.1653 | 4.3705 |
| Gender M | | 0 | . | . | . | . |
| Age*Gender F | | 5000 | -0.3439 | 0.1222 | -0.5893 | -0.1010 |
| Age*Gender M | | 0 | . | . | . | . |
| Residual UN(1,1) | Gender F | 5000 | 4.2321 | 1.8217 | 1.6445 | 7.8020 |
| Residual UN(2,1) | Gender F | 5000 | 2.7878 | 1.4636 | 0.7293 | 5.7478 |
| Residual UN(2,2) | Gender F | 5000 | 3.5504 | 1.5892 | 1.3130 | 6.4988 |
| Residual UN(3,1) | Gender F | 5000 | 3.5348 | 1.7821 | 1.0138 | 6.9532 |
| Residual UN(3,2) | Gender F | 5000 | 3.3304 | 1.6826 | 0.8530 | 6.5813 |
| Residual UN(3,3) | Gender F | 5000 | 5.0936 | 2.2021 | 1.8895 | 9.1471 |
| Residual UN(4,1) | Gender F | 5000 | 3.5407 | 1.8271 | 0.9083 | 7.1369 |
| Residual UN(4,2) | Gender F | 5000 | 3.3792 | 1.7284 | 0.8135 | 6.6893 |
| Residual UN(4,3) | Gender F | 5000 | 4.4877 | 2.1267 | 1.3969 | 8.5377 |
| Residual UN(4,4) | Gender F | 5000 | 5.4479 | 2.3883 | 2.0321 | 9.9821 |
| Residual UN(1,1) | Gender M | 5000 | 5.7815 | 2.1501 | 2.5561 | 10.0795 |
| Residual UN(2,1) | Gender M | 5000 | 1.9706 | 1.3886 | -0.3006 | 4.9087 |
| Residual UN(2,2) | Gender M | 5000 | 4.4630 | 1.6476 | 1.9748 | 7.6177 |
| Residual UN(3,1) | Gender M | 5000 | 3.1485 | 1.7035 | 0.3229 | 6.5712 |
| Residual UN(3,2) | Gender M | 5000 | 2.0219 | 1.4383 | -0.4459 | 5.0039 |
| Residual UN(3,3) | Gender M | 5000 | 6.3990 | 2.2197 | 2.9795 | 10.8205 |
| Residual UN(4,1) | Gender M | 5000 | 1.4408 | 1.3267 | -1.0554 | 4.1671 |
| Residual UN(4,2) | Gender M | 5000 | 2.5001 | 1.2867 | 0.4869 | 5.1913 |
| Residual UN(4,3) | Gender M | 5000 | 2.8916 | 1.4919 | 0.4865 | 6.0024 |
| Residual UN(4,4) | Gender M | 5000 | 4.2609 | 1.5520 | 1.9952 | 7.4605 |

Output 28.4.6 R-Side Covariance Matrix

| Estimated R Matrix | | | | | |
|--------------------|-----|--------|--------|--------|--------|
| Group | Row | Col 1 | Col 2 | Col 3 | Col 4 |
| Gender F | 1 | 4.2321 | 2.7878 | 3.5348 | 3.5407 |
| Gender F | 2 | 2.7878 | 3.5504 | 3.3304 | 3.3792 |
| Gender F | 3 | 3.5348 | 3.3304 | 5.0936 | 4.4877 |
| Gender F | 4 | 3.5407 | 3.3792 | 4.4877 | 5.4479 |
| Gender M | 1 | 5.7815 | 1.9706 | 3.1485 | 1.4408 |
| Gender M | 2 | 1.9706 | 4.4630 | 2.0219 | 2.5001 |
| Gender M | 3 | 3.1485 | 2.0219 | 6.3990 | 2.8916 |
| Gender M | 4 | 1.4408 | 2.5001 | 2.8916 | 4.2609 |

The “Posterior Summaries and Intervals” table in [Output 28.4.5](#) lists posterior estimates for the fixed-effects and residual covariance estimates. The posterior mean estimate of the boys’ intercept is 15.907, and that of the girls’ intercept is $15.907 + 1.4767 = 17.3837$. Similarly, the estimate for the boys’ slope is 0.8254, and

that of the girls' slope is $0.8254 - 0.3439 = 0.4815$. Thus the girls' starting point is greater than the boys' starting point, but the girls' growth rate is about half the boys' growth rate.

You can use **ESTIMATE** statements as follows to obtain the posterior distribution of these gender-related intercepts and slopes:

```
proc bglimm data=pr seed=475193 outpost=pr_out;
  class Person Gender Time;
  model Distance = Age|Gender;
  repeated Time / type=un subject=Person group=Gender r;
  estimate 'Girls Intercept' Int 1 Gender 1 0;
  estimate 'Boys Intercept' Int 1 Gender 0 1;
  estimate 'Girls Slope' Age 1 Age*Gender 1 0;
  estimate 'Boys Slope' Age 1 Age*Gender 0 1;
run;
```

Output 28.4.7 lists the posterior summary and interval statistics of the transformed parameters. These values are saved in the **OUTPOST=** data set, under variable names of **Girls_Intercept**, **Boys_Intercept**, **Girls_Slope**, and **Boys_Slope**.

Output 28.4.7 Intercepts and Slopes
The BGLIMM Procedure

| Results from ESTIMATE Statements | | | | |
|----------------------------------|---------|-----------------------|---------------------|---------|
| Label | Mean | Standard Deviation | 95% HPD Interval | |
| Girls Intercept | 17.3838 | 0.8261 | 15.7077 | 18.9964 |
| Boys Intercept | 15.9070 | 1.1413 | 13.7924 | 18.2695 |
| Girls Slope | 0.4814 | 0.0735 | 0.3405 | 0.6304 |
| Boys Slope | 0.8254 | 0.0962 | 0.6374 | 1.0135 |

With these posterior estimates, you can use the following statements to directly infer the probability—for example, that the boys' growth rate is greater than the girls' growth rate. The results are shown in Output 28.4.8.

```
data prob;
  set pr_out;
  pDiff = boys_slope - girls_slope;
  prob = (pDiff > 0);
  keep pDiff prob;
run;

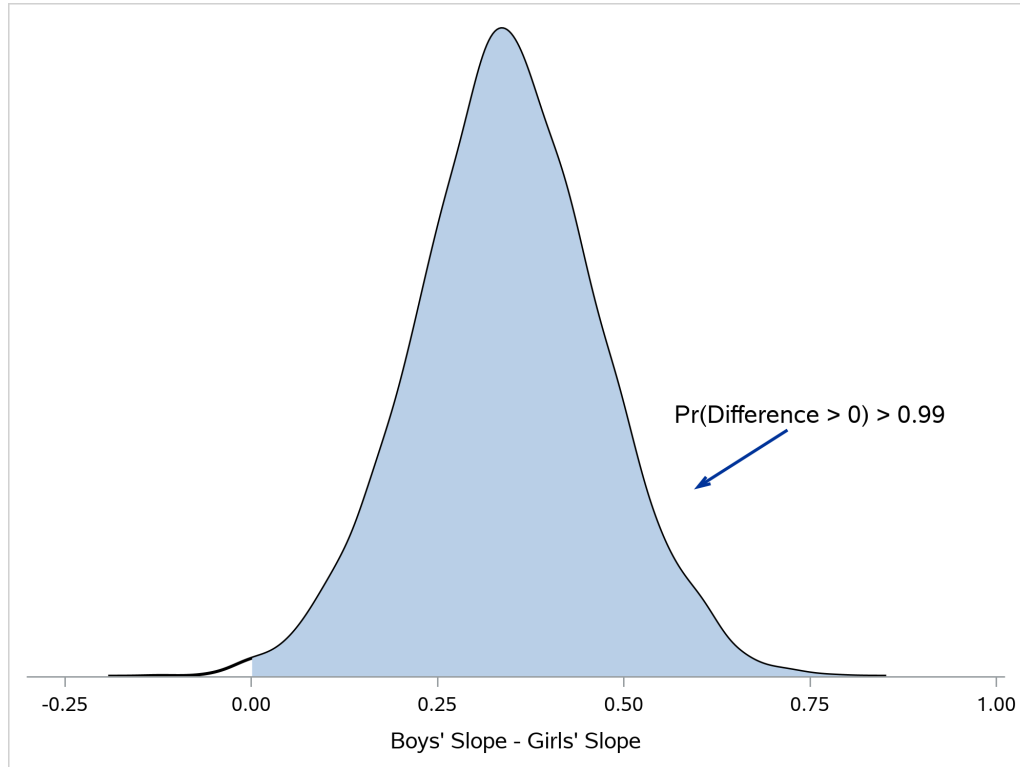
%sumint(data=prob, var=pDiff prob)
```

Output 28.4.8 Probability in Growth Rate Difference

| Posterior Summaries and Intervals | | | | | |
|-----------------------------------|------|--------|-----------------------|---------------------|--------|
| Parameter | N | Mean | Standard Deviation | 95% HPD Interval | |
| pDiff | 5000 | 0.3439 | 0.1222 | 0.1010 | 0.5893 |
| prob | 5000 | 0.9972 | 0.0528 | 1.0000 | 1.0000 |

The expected slope difference is 0.3439. [Output 28.4.9](#) plots the distribution of the difference in slopes; the shaded area under the positive axis represents the probability that the difference is greater than zero. The area is greater than 99%, and the probability that the boys have a greater growth slope than the girls is over 99%.

Output 28.4.9 Visualizing the Slope Difference



In [Output 28.4.5](#), two of the estimates equal 0; this is a result of the overparameterized model that PROC BGLIMM uses. You can obtain a full-rank parameterization by using the following **MODEL** statement:

```
model y = Gender Gender*Age / noint;
```

The posterior estimates from the full-rank parameterization (program not shown) are listed in [Output 28.4.10](#). The four estimates are the girls' intercept (17.3857), boys' intercept (15.9519), girls' slope (0.4803), and boys' slope (0.8212). You might find that this parameterization makes it easier to interpret the model.

Output 28.4.10 Posterior Summaries and Intervals from Full-Rank Parameterization**The BGLIMM Procedure**

| Posterior Summaries and Intervals | | | | | | |
|-----------------------------------|----------|------|---------|--------------------|------------------|---------|
| Parameter | Group | N | Mean | Standard Deviation | 95% HPD Interval | |
| Gender F | | 5000 | 17.3857 | 0.8215 | 15.7553 | 18.9998 |
| Gender M | | 5000 | 15.9519 | 1.1635 | 13.6144 | 18.1640 |
| Age*Gender F | | 5000 | 0.4803 | 0.0731 | 0.3371 | 0.6247 |
| Age*Gender M | | 5000 | 0.8212 | 0.0982 | 0.6247 | 1.0101 |
| Residual UN(1,1) | Gender F | 5000 | 4.2270 | 1.8240 | 1.6174 | 7.8089 |
| Residual UN(2,1) | Gender F | 5000 | 2.7806 | 1.4563 | 0.7068 | 5.6685 |
| Residual UN(2,2) | Gender F | 5000 | 3.5446 | 1.5888 | 1.3287 | 6.5472 |
| Residual UN(3,1) | Gender F | 5000 | 3.5277 | 1.7762 | 1.0206 | 6.8974 |
| Residual UN(3,2) | Gender F | 5000 | 3.3243 | 1.6771 | 0.9705 | 6.6051 |
| Residual UN(3,3) | Gender F | 5000 | 5.0854 | 2.1934 | 1.9174 | 9.1690 |
| Residual UN(4,1) | Gender F | 5000 | 3.5329 | 1.8214 | 0.8502 | 6.9560 |
| Residual UN(4,2) | Gender F | 5000 | 3.3734 | 1.7233 | 0.9427 | 6.8214 |
| Residual UN(4,3) | Gender F | 5000 | 4.4801 | 2.1247 | 1.3600 | 8.5490 |
| Residual UN(4,4) | Gender F | 5000 | 5.4416 | 2.3974 | 2.0595 | 10.0870 |
| Residual UN(1,1) | Gender M | 5000 | 5.7797 | 2.1686 | 2.5756 | 10.0388 |
| Residual UN(2,1) | Gender M | 5000 | 1.9783 | 1.3982 | -0.4409 | 4.7993 |
| Residual UN(2,2) | Gender M | 5000 | 4.4669 | 1.6497 | 1.9516 | 7.6345 |
| Residual UN(3,1) | Gender M | 5000 | 3.1495 | 1.6962 | 0.4838 | 6.6577 |
| Residual UN(3,2) | Gender M | 5000 | 2.0172 | 1.4267 | -0.4760 | 4.9811 |
| Residual UN(3,3) | Gender M | 5000 | 6.3930 | 2.2058 | 3.0217 | 10.8284 |
| Residual UN(4,1) | Gender M | 5000 | 1.4432 | 1.3330 | -0.9448 | 4.2283 |
| Residual UN(4,2) | Gender M | 5000 | 2.4933 | 1.2900 | 0.4163 | 5.1829 |
| Residual UN(4,3) | Gender M | 5000 | 2.8882 | 1.4818 | 0.5280 | 5.9364 |
| Residual UN(4,4) | Gender M | 5000 | 4.2620 | 1.5549 | 1.7624 | 7.2623 |

The posterior estimates of the covariance matrices ([Output 28.4.6](#)) are close to those from the independent model ([Output 28.4.2](#) and [Output 28.4.3](#)), and you are now able to make inferences on Gender-related questions.

The first two **ESTIMATE** statements in the following program compute the intercept and slope differences in gender, a more intuitive specification than in the overparameterized model. The last two **ESTIMATE** statements compute the estimated mean distance for girls and boys at age 11, respectively.

```
proc bglimm data=pr seed=475193 outpost=pr_out;
  class Person Gender Time;
  model Distance = Gender Age*Gender / noint;
  repeated Time / type=un subject=Person group=Gender;
  estimate 'Intercept Difference' Gender 1 -1;
  estimate 'Slope Difference' Gender*Age 1 -1;
  estimate 'Girl at Age 11' Gender 1 0 Gender*Age 11 0;
  estimate 'Boy at Age 11' Gender 0 1 Gender*Age 0 11;
run;
```

The results are shown in [Output 28.4.11](#).

Output 28.4.11 Posterior Summaries and Intervals**The BGLIMM Procedure**

| Results from ESTIMATE Statements | | | | |
|----------------------------------|---------|-----------------------|---------------------|---------|
| Label | Mean | Standard Deviation | 95% HPD Interval | |
| Intercept Difference | 1.4338 | 1.4336 | -1.3417 | 4.2965 |
| Slope Difference | -0.3409 | 0.1243 | -0.5923 | -0.1003 |
| Girl at Age 11 | 22.6688 | 0.5266 | 21.6213 | 23.6831 |
| Boy at Age 11 | 24.9852 | 0.4243 | 24.1859 | 25.8446 |

References

- Breslow, N. E., and Clayton, D. G. (1993). "Approximate Inference in Generalized Linear Mixed Models." *Journal of the American Statistical Association* 88:9–25.
- Clayton, D. G., and Kaldor, J. (1987). "Empirical Bayes Estimates of Age-Standardized Relative Risks for Use in Disease Mapping." *Biometrics* 43:671–681.
- Davidian, M., and Giltinan, D. M. (1995). *Nonlinear Models for Repeated Measurement Data*. New York: Chapman & Hall.
- Eilers, P. H. C., and Marx, B. D. (1996). "Flexible Smoothing with *B*-Splines and Penalties." *Statistical Science* 11:89–121. With discussion.
- Fuller, W. A. (1976). *Introduction to Statistical Time Series*. New York: John Wiley & Sons.
- Gamerman, D. (1997). "Sampling from the Posterior Distribution in Generalized Linear Models." *Statistics and Computing* 7:57–68.
- Gelman, A., and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge: Cambridge University Press.
- Hoffman, M. D., and Gelman, A. (2014). "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo." *Journal of Machine Learning Research* 15:1351–1381.
- Huynh, H., and Feldt, L. S. (1970). "Conditions Under Which Mean Square Ratios in Repeated Measurements Designs Have Exact F-Distributions." *Journal of the American Statistical Association* 65:1582–1589.
- Jennrich, R. I., and Schluchter, M. D. (1986). "Unbalanced Repeated-Measures Models with Structured Covariance Matrices." *Biometrics* 42:805–820.
- Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R. M. (1998). "Markov Chain Monte Carlo in Practice: A Roundtable Discussion." *American Statistician* 52:93–100.
- Kenward, M. G. (1987). "A Method for Comparing Profiles of Repeated Measurements." *Journal of the Royal Statistical Society, Series C* 36:296–308.
- Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., and Schabenberger, O. (2006). *SAS for Mixed Models*. 2nd ed. Cary, NC: SAS Institute Inc.

- Little, R. J. A., and Rubin, D. B. (2002). *Statistical Analysis with Missing Data*. 2nd ed. Hoboken, NJ: John Wiley & Sons.
- Macchiavelli, R. E., and Arnold, S. F. (1994). "Variable Order Ante-dependence Models." *Communications in Statistics—Theory and Methods* 23:2683–2699.
- McCullagh, P., and Nelder, J. A. (1989). *Generalized Linear Models*. 2nd ed. London: Chapman & Hall.
- Neal, R. M. (2011). "MCMC Using Hamiltonian Dynamics." In *Handbook of Markov Chain Monte Carlo*, edited by S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, 113–161. Boca Raton, FL: CRC Press.
- O'Malley, A. J., and Zaslavsky, A. M. (2008). "Domain-Level Covariance Analysis for Multilevel Survey Data with Structured Nonresponse." *Journal of the American Statistical Association* 103:1405–1418.
- Patel, H. I. (1991). "Analysis of Incomplete Data from a Clinical Trial with Repeated Measurements." *Biometrika* 78:609–619.
- Pothoff, R. F., and Roy, S. N. (1964). "A Generalized Multivariate Analysis of Variance Model Useful Especially for Growth Curve Problems." *Biometrika* 51:313–326.
- Rubin, D. B. (1976). "Inference and Missing Data." *Biometrika* 63:581–592.
- Vonesh, E. F., Chinchilli, V. M., and Pu, K. (1996). "Goodness-of-Fit in Generalized Nonlinear Mixed-Effects Models." *Biometrics* 52:572–587.

Subject Index

- ANTE(1) structure
 - BGLIMM procedure, 1243, 1244
- antependence structure
 - BGLIMM procedure, 1243, 1244
- AR(1) structure
 - BGLIMM procedure, 1243, 1244
- autoregressive moving average structure
 - BGLIMM procedure, 1243, 1245
- autoregressive structure
 - BGLIMM procedure, 1243, 1244
- banded Toeplitz structure
 - BGLIMM procedure, 1243, 1246
- BGLIMM procedure, 1206
 - ANTE(1) structure, 1243, 1244
 - antependence structure, 1243, 1244
 - AR(1) structure, 1243, 1244
 - ARMA structure, 1243
 - autoregressive moving average structure, 1243, 1245
 - autoregressive structure, 1243, 1244
 - banded Toeplitz structure, 1243, 1246
 - binomial distribution, 1254
 - class level, 1222, 1269
 - compound symmetry structure, 1243, 1245
 - conjugate sampling, 1258
 - constant priors for fixed effects, 1270
 - correlation estimates, 1251
 - covariance parameters, 1209
 - covariance structure, 1243, 1247
 - dispersion parameter, 1256
 - displayed output, 1269
 - distribution function, 1235
 - estimability, 1230
 - estimate specification, 1229
 - estimates, 1229
 - events/trials format for response, 1254
 - fixed effects, 1207
 - G matrix, 1238, 1240, 1241
 - Gamerman algorithm, 1259
 - gamma distribution, 1253
 - general linear covariance structure, 1243
 - generalized linear mixed models, 1251
 - geometric distribution, 1253
 - group effect, 1241
 - heterogeneity, 1250
 - heterogeneous AR(1) structure, 1243, 1244
 - heterogeneous autoregressive structure, 1244
 - heterogeneous compound symmetry structure, 1243, 1245
 - heterogeneous Toeplitz structure, 1243, 1246
 - Huynh-Feldt covariance structure, 1246
 - independent normal priors for fixed effects, 1270
 - initial values, 1265
 - input data sets, 1219
 - inverse Gaussian distribution, 1253
 - link function, 1207, 1236
 - missing data, 1266
 - model information, 1269
 - model options summary, 1232
 - negative binomial distribution, 1254
 - normal distribution, 1253
 - normal priors for fixed effects, 1270
 - notation, 1207
 - number of observations, 1269
 - ODS Graphics, 1274
 - ODS table names, 1272
 - parameters initial value, 1270
 - Poisson distribution, 1254
 - posterior samples data set, 1222
 - prior distributions, 1261
 - priors for scale and covariance parameters, 1270
 - R matrix, 1248, 1250
 - random effects, 1207, 1238
 - random-effects parameters, 1241
 - repeated measures, 1248
 - response profile, 1269
 - response variable options, 1232
 - response-level ordering, 1232
 - sampling methods, 1258, 1260
 - scale parameter, 1209, 1252, 1256, 1286
 - subject effect, 1243, 1251
 - Toeplitz structure, 1243, 1246
 - unstructured correlations, 1243
 - unstructured covariance, 1246
 - unstructured covariance matrix, 1243
 - variance components, 1243
- BGLIMM procedure, MODEL statement
 - initial values, 1236
- binomial distribution
 - BGLIMM procedure, 1254
- class level
 - BGLIMM procedure, 1222, 1269
- compound symmetry structure
 - BGLIMM procedure, 1243, 1245

- conjugate sampling
 - BGLIMM procedure, 1258
- constant priors for fixed effects
 - BGLIMM procedure, 1270
- correlation
 - estimates (BGLIMM), 1251
- covariance
 - parameters (BGLIMM), 1209
- covariance structure
 - antedependence (BGLIMM), 1244
 - autoregressive (BGLIMM), 1209, 1244
 - autoregressive moving average (BGLIMM), 1245
 - banded (BGLIMM), 1246
 - BGLIMM procedure, 1243
 - compound symmetry (BGLIMM), 1245
 - equicorrelation (BGLIMM), 1245
 - examples (BGLIMM), 1247
 - factor-analytic (BGLIMM), 1245
 - G-side (BGLIMM), 1243
 - general (BGLIMM), 1207
 - heterogeneous autoregressive (BGLIMM), 1244
 - heterogeneous compound symmetry (BGLIMM), 1245
 - heterogeneous Toeplitz (BGLIMM), 1246
 - Huynh-Feldt (BGLIMM), 1246
 - R-side (BGLIMM), 1243
 - Toeplitz (BGLIMM), 1246
 - unstructured (BGLIMM), 1246
 - variance components (BGLIMM), 1247
- dispersion parameter
 - BGLIMM procedure, 1256
- displayed output
 - BGLIMM procedure, 1269
- distribution function
 - BGLIMM procedure, 1235
- estimability
 - BGLIMM procedure, 1230
- estimate specification
 - BGLIMM procedure, 1229
- estimates
 - BGLIMM procedure, 1229
- events/trials format for response
 - BGLIMM procedure, 1254
- examples, BGLIMM
 - covariance structure, 1247
 - estimate, with groups, 1231
 - full-rank parameterization, 1303
 - getting started, 1210
 - group option in estimate, 1231
 - logistic regression with random intercepts, 1210
 - Pothoff and Roy growth measurements, 1296
 - Scottish lip cancer data, 1290
- fixed effects
 - BGLIMM procedure, 1207
- G matrix
 - BGLIMM procedure, 1238, 1240, 1241
- Gamerman algorithm
 - BGLIMM procedure, 1259
- gamma distribution
 - BGLIMM procedure, 1253
- general linear covariance structure
 - BGLIMM procedure, 1243
- generalized linear mixed model, *see also* BGLIMM procedure
- generalized linear mixed model (BGLIMM)
 - theory, 1251
- generalized linear model, *see also* BGLIMM procedure
- geometric distribution
 - BGLIMM procedure, 1253
- GLM, *see also* BGLIMM procedure
- GLMM, *see also* BGLIMM procedure
- group effect
 - BGLIMM procedure, 1241
- Hamiltonian Monte Carlo sampler
 - sampling methods, 1260
- heterogeneity
 - BGLIMM procedure, 1250
- heterogeneous
 - AR(1) structure (BGLIMM), 1243
 - compound symmetry structure (BGLIMM), 1243
 - Toeplitz structure (BGLIMM), 1243
- heterogeneous AR(1) structure
 - BGLIMM procedure, 1244
- heterogeneous autoregressive structure
 - BGLIMM procedure, 1244
- heterogeneous compound symmetry structure
 - BGLIMM procedure, 1245
- heterogeneous Toeplitz structure
 - BGLIMM procedure, 1246
- Huynh-Feldt
 - structure (BGLIMM), 1243, 1246
- independent normal priors for fixed effects
 - BGLIMM procedure, 1270
- initial values
 - BGLIMM procedure, 1265
 - BGLIMM procedure, MODEL statement, 1236
- inverse Gaussian distribution
 - BGLIMM procedure, 1253
- link function
 - BGLIMM procedure, 1207, 1236
- missing data
 - BGLIMM procedure, 1266

- model information
 - BGLIMM procedure, [1269](#)
- negative binomial distribution
 - BGLIMM procedure, [1254](#)
- normal distribution
 - BGLIMM procedure, [1253](#)
- normal priors for fixed effects
 - BGLIMM procedure, [1270](#)
- notation
 - BGLIMM procedure, [1207](#)
- number of observations
 - BGLIMM procedure, [1269](#)
- ODS Graphics
 - BGLIMM procedure, [1274](#)
- ODS table names
 - BGLIMM procedure, [1272](#)
- options summary
 - PROC BGLIMM statement, [1218](#)
 - RANDOM statement (BGLIMM), [1238](#)
 - REPEATED statement (BGLIMM), [1248](#)
- parameters initial value
 - BGLIMM procedure, [1270](#)
- Poisson distribution
 - BGLIMM procedure, [1254](#)
- posterior samples data set
 - BGLIMM procedure, [1222](#)
- prior
 - prior for the fixed-effects coefficients, [1261](#)
 - prior for the G-side covariance, [1262](#)
 - prior for the R-side covariance, [1264](#)
 - prior for the random-effects coefficients, [1262](#)
 - prior for the scale parameter, [1264](#)
- prior distributions
 - BGLIMM procedure, [1261](#)
- priors for scale and covariance parameters
 - BGLIMM procedure, [1270](#)
- R matrix
 - BGLIMM procedure, [1248](#), [1250](#)
- random effects
 - BGLIMM procedure, [1207](#), [1238](#)
- random-effects parameters
 - BGLIMM procedure, [1241](#)
- repeated measures
 - BGLIMM procedure, [1248](#)
- response profile
 - BGLIMM procedure, [1269](#)
- response variable options
 - BGLIMM procedure, [1232](#)
- response-level ordering
 - BGLIMM procedure, [1232](#)
- reverse response-level ordering
 - BGLIMM procedure, [1232](#)
- sampling methods
 - BGLIMM procedure, [1258](#), [1260](#)
 - Hamiltonian Monte Carlo sampler, [1260](#)
- scale parameter
 - BGLIMM procedure, [1209](#), [1252](#), [1286](#)
 - BGLIMM procedure compared to GLIMMIX and GENMOD procedures, [1256](#)
- subject effect
 - BGLIMM procedure, [1243](#), [1251](#)
- Toeplitz structure
 - BGLIMM procedure, [1243](#), [1246](#)
- unstructured correlations
 - BGLIMM procedure, [1243](#)
- unstructured covariance
 - BGLIMM procedure, [1246](#)
- unstructured covariance matrix
 - BGLIMM procedure, [1243](#)
- variance components
 - BGLIMM procedure, [1243](#)

Syntax Index

- BGLIMM procedure, 1218
 - CLASS statement, 1227
 - ESTIMATE statement, 1229
 - MODEL statement, 1231
 - PROC BGLIMM statement, 1218
 - RANDOM statement, 1238
 - syntax, 1218
- BGLIMM procedure, BY statement, 1227
- BGLIMM procedure, CLASS statement, 1227
 - CPREFIX= option, 1227
 - DESCENDING option, 1227
 - LPREFIX= option, 1228
 - MISSING option, 1228
 - ORDER= option, 1228
 - PARAM= option, 1228
 - REF= option, 1229
 - TRUNCATE option, 1229
- BGLIMM procedure, ESTIMATE statement, 1229
 - DIVISOR= option, 1231
 - E option, 1231
 - GROUP option, 1231
 - SUBJECT option, 1231
- BGLIMM procedure, MODEL statement, 1231
 - COEFFPRIOR= option, 1234
 - DESCENDING option, 1232
 - DISTRIBUTION= option, 1235
 - INIT= option, 1236
 - LINK= option, 1236
 - NOINT option, 1237
 - NOOUTPOST option, 1237
 - OFFSET= option, 1237
 - ORDER= option, 1233
 - SCALEPRIOR option, 1237
- BGLIMM procedure, PROC BGLIMM statement, 1218
 - DATA= option, 1219
 - DIAG= option, 1219
 - DIAGNOSTICS= option, 1219
 - DIC option, 1221
 - LOGPOST option, 1221
 - MISSING= option, 1221
 - NBI= option, 1221
 - NMC= option, 1222
 - NOCLPRINT option, 1222
 - NOPRINT option, 1222
 - NTHREADS= option, 1222
 - OUTPOST= option, 1222
 - PLOTS= option, 1222
 - SAMEBYSEED option, 1225
 - SEED= option, 1225
 - SINGCHOL= option, 1225
 - SINGULAR= option, 1225
 - STATISTICS= option, 1225
 - STATS= option, 1225
 - THIN= option, 1226
- BGLIMM procedure, RANDOM statement, 1238
 - COVPRIOR= option, 1239
 - G option, 1240, 1241
 - GROUP= option, 1241
 - MONITOR option, 1241
 - NOOUTPOST option, 1242
 - NUTS option, 1242
 - RESIDUAL option, 1242
 - RSIDE option, 1242
 - SUBJECT= option, 1243
 - TYPE= option, 1243
- BGLIMM procedure, REPEATED statement, 1248
 - COVPRIOR= option, 1249
 - GROUP= option, 1250
 - NUTS option, 1250
 - R option, 1250
 - RCORR option, 1251
 - SUBJECT= option, 1251
 - TYPE= option, 1251
- BY statement
 - BGLIMM procedure, 1227
- CLASS statement
 - BGLIMM procedure, 1227
- COEFFPRIOR= option
 - MODEL statement (BGLIMM), 1234
- COVPRIOR= option
 - RANDOM statement (BGLIMM), 1239
 - REPEATED statement (BGLIMM), 1249
- CPREFIX= option
 - CLASS statement (BGLIMM), 1227
- DATA= option
 - PROC BGLIMM statement, 1219
- DESCENDING option
 - CLASS statement (BGLIMM), 1227
 - MODEL statement (BGLIMM), 1232
- DIAG= option
 - PROC BGLIMM statement, 1219
- DIAGNOSTICS= option
 - PROC BGLIMM statement, 1219
- DIC option

PROC BGLIMM statement, [1221](#)
 DISTRIBUTION= option
 MODEL statement (BGLIMM), [1235](#)
 DIVISOR= option
 ESTIMATE statement (BGLIMM), [1231](#)

 E option
 ESTIMATE statement (BGLIMM), [1231](#)
 ESTIMATE statement
 BGLIMM procedure, [1229](#)

 G option
 RANDOM statement (BGLIMM), [1240](#), [1241](#)
 GROUP option
 ESTIMATE statement (BGLIMM), [1231](#)
 GROUP= option
 RANDOM statement (BGLIMM), [1241](#)
 REPEATED statement (BGLIMM), [1250](#)

 INIT= option
 MODEL statement (BGLIMM), [1236](#)

 LINK= option
 MODEL statement (BGLIMM), [1236](#)
 LOGPOST option
 PROC BGLIMM statement, [1221](#)
 LPREFIX= option
 CLASS statement (BGLIMM), [1228](#)

 MISSING option
 CLASS statement (BGLIMM), [1228](#)
 MISSING= option
 PROC BGLIMM statement, [1221](#)
 MODEL statement
 BGLIMM procedure, [1231](#)
 MONITOR option
 RANDOM statement (BGLIMM), [1241](#)

 NBI= option
 PROC BGLIMM statement, [1221](#)
 NMC= option
 PROC BGLIMM statement, [1222](#)
 NOCLPRINT option
 PROC BGLIMM statement, [1222](#)
 NOINT option
 MODEL statement (BGLIMM), [1237](#)
 NOOUTPOST option
 MODEL statement (BGLIMM), [1237](#)
 RANDOM statement (BGLIMM), [1242](#)
 NOPRINT option
 PROC BGLIMM statement, [1222](#)
 NTHREADS= option
 PROC BGLIMM statement, [1222](#)
 NUTS option
 RANDOM statement (BGLIMM), [1242](#)

 REPEATED statement (BGLIMM), [1250](#)

 OFFSET= option
 MODEL statement (BGLIMM), [1237](#)
 ORDER= option
 CLASS statement (BGLIMM), [1228](#)
 MODEL statement (BGLIMM), [1233](#)
 OUTPOST= option
 PROC BGLIMM statement, [1222](#)

 PARAM= option
 CLASS statement (BGLIMM), [1228](#)
 PLOTS= option
 PROC BGLIMM statement, [1222](#)
 PROC BGLIMM statement, *see* BGLIMM procedure

 R option
 REPEATED statement (BGLIMM), [1250](#)
 RANDOM statement
 BGLIMM procedure, [1238](#)
 RCORR option
 REPEATED statement (BGLIMM), [1251](#)
 REF= option
 CLASS statement (BGLIMM), [1229](#)
 REPEATED statement
 BGLIMM procedure, [1248](#)
 RESIDUAL option
 RANDOM statement (BGLIMM), [1242](#)
 RSIDE option
 RANDOM statement (BGLIMM), [1242](#)

 SAMEBYSEED option
 PROC BGLIMM statement, [1225](#)
 SCALEPRIOR option
 MODEL statement, [1237](#)
 SEED= option
 PROC BGLIMM statement, [1225](#)
 SINGCHOL= option
 PROC BGLIMM statement, [1225](#)
 SINGULAR= option
 PROC BGLIMM statement, [1225](#)
 STATISTICS= option
 PROC BGLIMM statement, [1225](#)
 STATS= option
 PROC BGLIMM statement, [1225](#)
 SUBJECT option
 ESTIMATE statement (BGLIMM), [1231](#)
 SUBJECT= option
 RANDOM statement (BGLIMM), [1243](#)
 REPEATED statement (BGLIMM), [1251](#)
 syntax
 BGLIMM procedure, [1218](#)

 THIN= option
 PROC BGLIMM statement, [1226](#)

TRUNCATE option

CLASS statement (BGLIMM), [1229](#)

TYPE= option

RANDOM statement (BGLIMM), [1243](#)

REPEATED statement (BGLIMM), [1251](#)