SAS/STAT® 14.2 User's Guide
The BCHOICE Procedure

This document is an individual chapter from *SAS/STAT® 14.2 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS/STAT® 14.2 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 14.2 User's Guide

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

November 2016

Chapter 27
The BCHOICE Procedure

Contents

Overview: BCHOICE Procedure

The BCHOICE (Bayesian choice) procedure performs Bayesian analysis for discrete choice models. Discrete choice models are used in marketing research to model decision makers' choices among alternative products and services. The decision maker might be people, households, companies and so on, and the alternatives might be products, services, actions, or any other options or items about which choices must be made (Train 2009). The collection of alternatives that are available to the decision makers is called a choice set.

Discrete choice models are derived under the assumption of utility-maximizing behavior by decision makers. When individuals are asked to make one choice among a set of alternatives, they usually determine the level of utility that each alternative offers. The utility that individual i obtains from alternative j among J alternatives is denoted as

$$u_{ij} = v_{ij} + \epsilon_{ij}, \quad i = 1, \ldots, N \text{ and } j = 1, \ldots, J$$

where the subscript i is an index for the individuals, the subscript j is an index for the alternatives in a choice set, v_{ij} is a nonstochastic utility function that relates observed factors to the utility, and ϵ_{ij} is the error component that captures the unobserved characteristics of the utility. In discrete choice models, the observed part of the utility function is assumed to be linear in the parameters,

$$v_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta}$$

where $\mathbf{x}_{ij}$ is a p-dimensional design vector of observed attribute levels that relate to alternative j and $\boldsymbol{\beta}$ is the corresponding vector of fixed regression coefficients that indicate the utilities or part-worths of the attribute levels.

Decision makers choose the alternative that gives them the greatest utility. Let $\mathbf{y}_i$ be the multinomial response vector for the ith individual. The value y_{ij} takes 1 if the jth component of $\mathbf{u}_i = (u_{i1}, \ldots, u_{iJ})$ is the largest, and 0 otherwise:

$$\begin{aligned} u_{ij} &= \mathbf{x}'_{ij}\boldsymbol{\beta} + \epsilon_{ij} \\ y_{ij} &= \begin{cases} 1 & \text{if } u_{ij} \geq \max(\mathbf{u}_i) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Different specifications about the density of the error vector $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \ldots, \epsilon_{iJ})$ result in different types of choice models: logit, nested logit, and probit, as detailed in the section "Types of Choice Models" on page 1065. Logit and nested logit models have closed-form likelihood, whereas a probit model does not.

The past 15 years have seen a dramatic increase in using a Bayesian approach to develop new methods of analysis and models of consumer behavior. The milestone breakthroughs are Albert and Chib (1993) and McCulloch and Rossi (1994) for choice probit models, and Allenby and Lenk (1994) and Allenby (1997) for logit models that have normally distributed random effects (these models are called mixed logit models). Train (2009) extends the Bayesian procedure for mixed logit models to include nonnormal distributions, such as lognormal, uniform, and triangular distributions.

An issue in choice models is that the quantity of relevant data at the individual level is very limited. Respondents frequently become fatigued after answering 15 to 20 questions in a survey. Rossi, McCulloch, and Allenby (1996) and Allenby and Rossi (1999) show how to obtain information about individual-level

parameters within a model by using random taste variation. The lack of data at the individual level, combined with the desire to account for individual differences instead of treating all respondents alike, presents challenges in marketing research. Bayesian methods are ideally suited for analyzing limited data.

Bayesian methods have several advantages. First, Bayesian methods do not require optimization of any function. For probit models and logit models that have random effects, optimization of the likelihood function can be numerically difficult. Different starting values might lead to different maximization results. The problem of a local maximum versus a global maximum is another issue, because convergence is not guaranteed to find the global maximum. Second, Bayesian procedures enable consistency and efficiency to be achieved under more relaxed conditions. When the likelihood function does not have a closed form or there are too many parameters (as in models with random effects), simulation can be used to estimate the likelihood function. Maximization that is based on such a simulated likelihood function is consistent only if the number of draws in simulation rises with the sample size, and it is efficient only if the number of draws in simulation increases faster than the square root of the sample size. On the other hand, Bayesian methods are consistent for a fixed number of draws and are efficient if the number of draws goes up at any rate with the sample size.

For a short introduction to Bayesian analysis and related basic concepts, see Chapter 7, "Introduction to Bayesian Analysis Procedures." Also see the section "A Bayesian Reading List" on page 153 for a guide to Bayesian textbooks of varying degrees of difficulty. It follows from Bayes' theorem that a posterior distribution is proportional to the product of the likelihood function and the prior distribution of the parameter. When it is difficult to obtain the posterior distribution analytically, Bayesian methods often rely on simulations to generate samples from the posterior distribution, and they use the simulated draws to approximate the distribution and to make the inferences.

To use the BCHOICE procedure, you need to specify the type of model for the data. You can also supply a prior distribution for the parameters if you want something other than the default noninformative prior. PROC BCHOICE obtains samples from the corresponding posterior distributions, produces summary and diagnostic statistics, and saves the posterior samples in an output data set that can be used for further analysis. The procedure derives inferences from simulation rather than through analytic or numerical methods. You should expect slightly different answers from each run for the same problem, unless you use the same random number seed.

PROC BCHOICE Compared with Other SAS Procedures

The underlying structure of PROC BCHOICE is similar to the structure of PROC MCMC in that both procedures obtain samples from the posterior distributions and produce summary and diagnostic statistics when you specify the model or the priors or both. However, they differ in that PROC MCMC is a general-purpose Markov chain Monte Carlo (MCMC) simulation procedure that is designed to fit a wide range of Bayesian models, whereas PROC BCHOICE is designed specifically for discrete choice models. You can call PROC MCMC to analyze data that have any likelihood, prior, or hyperprior, as long as these functions can be programmed by using the SAS DATA step functions. For example, you can fit choice logit and nested logit models in PROC MCMC by using some SAS coding to specify the likelihood. PROC BCHOICE works only with choice models, but it is customized to fit special characteristics and features in a choice model. The syntax is quite different from PROC MCMC's syntax. PROC BCHOICE provides a CLASS statement to handle categorical variables, and it requires less complicated SAS coding for choice models. The default sampling method for choice logit models when direct sampling is not available is the Metropolis-Hastings method, which is based on the Gamerman approach, which in turn has proved to be often more efficient than

the random walk Metropolis algorithm that PROC MCMC uses. In addition, it is difficult to fit choice probit models in PROC MCMC.

For a standard logit choice model, you can use the TIES=BRESLOW option in the PHREG procedure. The approach has the same likelihood as PROC BCHOICE of fitting the data. You use the STRATA statement in PROC PHREG to specify how to define the choice set. However, this is a frequentist approach, and it does not work for a choice model with random effects.

Getting Started: BCHOICE Procedure

The first example in this section is a simple logit conjoint analysis that illustrates some basic features of PROC BCHOICE. The second example discusses a mixed logit model with individual-level random effects, which has become increasingly popular.

A Simple Logit Model Example

This example uses a standard logit model in which the error component, ϵ_{ij} $(j = 1, \ldots, J)$, is independently and identically distributed (iid) with the Type I extreme-value distribution, $\exp(-\exp(-\epsilon_{ij}))$. This assumption provides a convenient logit form for the choice probability (McFadden 1974):

$$P(y_{ij} = 1) = \frac{\exp(\mathbf{x}_{ij}'\boldsymbol{\beta})}{\sum_{k=1}^{J} \exp(\mathbf{x}_{ik}'\boldsymbol{\beta})}, \quad i = 1, \ldots, N \text{ and } j = 1, \ldots, J$$

The likelihood is formed by the product of the N independent multinomial distributions:

$$p(\mathbf{Y}|\boldsymbol{\beta}) = \prod_{i=1}^{N} \prod_{j=1}^{J} P(y_{ij} = 1)^{y_{ij}}$$

Suppose $\boldsymbol{\beta}$ has a normal prior

$$\pi(\boldsymbol{\beta}) = \mathrm{N}(\mathbf{0}, c\mathbf{I})$$

where $\mathbf{I}$ is the identity matrix and c is a scalar. c is often set to be large for a noninformative prior.

The posterior density of the parameter $\boldsymbol{\beta}$ is

$$p(\boldsymbol{\beta}|\mathbf{Y}) \propto p(\mathbf{Y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta})$$

PROC BCHOICE obtains samples from the posterior distribution, produces summary and diagnostic statistics, and saves the posterior samples in an output data set that can be used for further analysis.

In this example (Kuhfeld 2010), each of 10 subjects is presented with eight different chocolate candies and asked to choose one. The eight candies consist of the 2^3 combinations of dark or milk chocolate, soft or chewy center, and nuts or no nuts. Each subject sees all eight alternatives and makes one choice. Experimental choice data such as these are usually analyzed by using a multinomial logit model.

The following statements read the data:

```
title 'Conjoint Analysis of Chocolate Candies';

data Chocs;
   input Subj Choice Dark Soft Nuts;
   datalines;
1 0 0 0 0
1 0 0 0 1
1 0 0 1 0
1 0 0 1 1
1 1 1 0 0
1 0 1 0 1

   ... more lines ...

10 0 1 0 0
10 1 1 0 1
10 0 1 1 0
10 0 1 1 1
;

proc print data=Chocs (obs=16);
run;
```

The data for the first two subjects are shown in Figure 27.1.

Figure 27.1 Data for the First Two Subjects

Conjoint Analysis of Chocolate Candies

Obs	Subj	Choice	Dark	Soft	Nuts
1	1	0	0	0	0
2	1	0	0	0	1
3	1	0	0	1	0
4	1	0	0	1	1
5	1	1	1	0	0
6	1	0	1	0	1
7	1	0	1	1	0
8	1	0	1	1	1
9	2	0	0	0	0
10	2	0	0	0	1
11	2	0	0	1	0
12	2	0	0	1	1
13	2	0	1	0	0
14	2	1	1	0	1
15	2	0	1	1	0
16	2	0	1	1	1

The data set contains 10 subjects and 80 observation lines. Each line of the data represents one alternative in the choice set for each subject. It is required that the response variable, which is Choice in this study, indicate the chosen alternative by the value 1 and the unchosen alternatives by the value 0. Dark is 1 for dark

chocolate and 0 for milk chocolate; Soft is 1 for soft center and 0 for chewy center; Nuts is 1 if the candy contains nuts and 0 if it does not contain nuts. In this example, subject 1 chose the fifth alternative (Choice=1, which is Dark/Chewy/No Nuts) among the eight alternatives in the choice set. All the chosen and unchosen alternatives must appear in the data set. If you have choice data in which all alternatives appear on one line, then you must rearrange the data in the correct form (that is, the data must contain one observation line for each alternative of each choice set for each subject).

The following statements fit a multinomial logit model:

```
ods graphics on;
proc bchoice data=Chocs outpost=Bsamp nmc=10000 thin=2 diag=(AutoCorr
     ESS MCSE) seed=124;
   class Dark(ref='0') Soft(ref='0') Nuts(ref='0') Subj;
   model Choice = Dark Soft Nuts / choiceset=(Subj) cprior=normal(var=1000);
run;
```

The ODS GRAPHICS ON statement invokes the ODS Graphics environment and displays the diagnostic plots, such as the trace and autocorrelation function plots of the posterior samples. For more information about ODS, see Chapter 21, "Statistical Graphics Using ODS."

The PROC BCHOICE statement invokes the procedure, and the DATA= option specifies the input data set Chocs. The OUTPOST= option requests an output data set called Bsamp to contain all the posterior samples. 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 to keep one of every two samples. Thinning is often used to reduce correlation among posterior sample draws. In this example, 5,000 simulated values are saved in the Bsamp data set. The DIAG=(AUTOCORR ESS MCSE) option requests that three convergence diagnostics be output to help determine whether the chain has converged: the autocorrelations, effective sample sizes, and Monte Carlo standard errors. The SEED= option specifies a seed for the random number generator, which guarantees the reproducibility of the random stream.

The CLASS statement names the classification variables to be used in the model. The CLASS statement must precede the MODEL statement (as it must in most other SAS procedures). The REF= option specifies the reference level.

The MODEL statement is required; it defines the dependent variable (y_{ij}) and independent variables (x_{ij}). To the left of the equal sign in the MODEL statement, you specify the dependent variable that indicates which alternatives are chosen and which ones are not chosen. The dependent variable Choice has the value 1 (chosen) or 0 (unchosen). You specify the independent variables after the equal sign. The independent variables often indicate attributes or characteristics of the alternatives in the choice set, such as Dark, Soft, and Nuts in this example. The CHOICESET= option specifies how a choice set is defined. In this example, CHOICESET=(Subj) because there is one and only one choice set per subject. The variable that you specify in the CHOICESET= option must be a classification variable that appears in the CLASS statement. Always use the CHOICESET= option to define the choice set.

The first table that PROC BCHOICE produces is the "Model Information" table, as shown in Figure 27.2. This table displays basic information about the analysis, such as the name of the input data set, response variable, model type, choice response type, 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. Here the response type is binary, where the response takes either 1 (chosen) or 0 (unchosen). Other types of choice response include MaxDiff and allocation, where the response can take other values.

Figure 27.2 Model Information

The BCHOICE Procedure

Model Information	
Data Set	WORK.CHOCS
Response Variable	Choice
Type of Model	Logit
Type of Choice Response	Binary
Fixed Effects Included	Yes
Random Effects Included	No
Sampling Algorithm	Gamerman Metropolis
Burn-In Size	500
Simulation Size	10000
Thinning	2
Random Number Seed	124
Number of Threads	1

The "Choice Sets Summary" table is displayed by default; you can use it to check the data entry. In this case, there are 10 choice sets, one for each subject. All 10 choice sets display the same pattern: they have a total of eight alternatives, one of which is chosen and seven of which are unchosen. Sometimes there can be more than one pattern among the choice sets (for example, some choice sets have nine alternatives). Logit models can have different patterns as long as each choice set has at least two alternatives in total and only one chosen alternative. If more than one alternative is chosen or no alternative is chosen, the choice set is invalid for the binary response. PROC BCHOICE deletes all the invalid choice sets and produces a warning message.

Figure 27.3 Choice Sets Summary

Choice Sets Summary				
Pattern	Choice Sets	Total Alternatives	Chosen Alternatives	Not Chosen
1	10	8	1	7

The next table is the "Number of Observations" table, as shown in Figure 27.4. This table lists the number of observations that are read from the DATA= data set and the number of nonmissing and valid observations that are used in the analysis. PROC BCHOICE does not impute missing values. If any missing value is encountered in a row of the DATA= data set, PROC BCHOICE skips that row and moves on to read the next row. If the missing value causes the corresponding choice set to be invalid, the entire choice set is discarded from analysis. In this example, all 80 observations are used.

Figure 27.4 Number of Observations

Number of Observations	
Number of Observations Read	80
Number of Observations Used	80

PROC BCHOICE reports posterior summary statistics (posterior means, standard deviations, and highest posterior density (HPD) intervals) for each parameter, as shown in Figure 27.5. For more information about posterior statistics, see the section "Summary Statistics" on page 150 in Chapter 7, "Introduction to Bayesian Analysis Procedures."

Figure 27.5 PROC BCHOICE Posterior Summary Statistics

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Dark 1	5000	1.5308	0.7943	0.1848	3.2412
Soft 1	5000	-2.4312	0.9792	-4.4882	-0.8125
Nuts 1	5000	0.9671	0.7454	-0.3255	2.6675

Before you examine the posterior summary statistics and try to draw any conclusions, you might want to verify that the simulation has converged. Convergence diagnostics are essential to inferring from simulations that are based on Markov chains. If the Markov chain has not converged, all conclusions that are based on the samples might be misleading. PROC BCHOICE computes the effective sample size by default. There are a number of other convergence diagnostics to help you determine whether the chain has converged: the Monte Carlo standard errors, the autocorrelations at selected lags, and so on. These statistics are shown in Figure 27.6. For details and interpretations of these diagnostics, see the section "Assessing Markov Chain Convergence" on page 136 in Chapter 7, "Introduction to Bayesian Analysis Procedures."

The "Posterior Autocorrelations" table shows that the autocorrelations among posterior samples reduce quickly. The "Effective Sample Sizes" table reports the number of effective sample sizes of the Markov chain. The "Monte Carlo Standard Errors" table indicates that the standard errors of the mean estimates for each of the variables are relatively small with respect to the posterior standard deviations. The values in the MCSE/SD column (ratios of the standard errors and the standard deviations) are small. This means that only a fraction of the posterior variability is caused by the simulation.

Figure 27.6 PROC BCHOICE Convergence Diagnostics

Posterior Autocorrelations

Parameter	Lag 1	Lag 5	Lag 10	Lag 50
Dark 1	0.4495	0.1427	0.0603	-0.0095
Soft 1	0.5389	0.1518	0.0730	0.0449
Nuts 1	0.4136	0.1473	0.0259	-0.0264

Effective Sample Sizes

Parameter	ESS	Autocorrelation Time	Efficiency
Dark 1	1078.4	4.6365	0.2157
Soft 1	904.9	5.5253	0.1810
Nuts 1	1161.6	4.3043	0.2323

Monte Carlo Standard Errors

Parameter	MCSE	Standard Deviation	MCSE/SD
Dark 1	0.0242	0.7943	0.0305
Soft 1	0.0326	0.9792	0.0332
Nuts 1	0.0219	0.7454	0.0293

PROC BCHOICE produces a number of graphs, shown in Figure 27.7, which also aid convergence diagnostic checks. The trace plots have two important aspects to examine. First, you want to check whether the mean of the Markov chain has stabilized and appears constant over the graph. Second, you want to check whether the chain has good mixing and is "dense," in the sense that it quickly traverses the support of the distribution to explore both the tails and the mode areas efficiently. The plots show that the chains appear to have reached their stationary distributions.

Next, you want to examine the autocorrelation plots, which indicate the degree of autocorrelation for each of the posterior samples. High correlations usually imply slow mixing. Finally, the kernel density plots estimate the posterior marginal distributions for each parameter.

Figure 27.7 PROC BCHOICE Diagnostic Plots

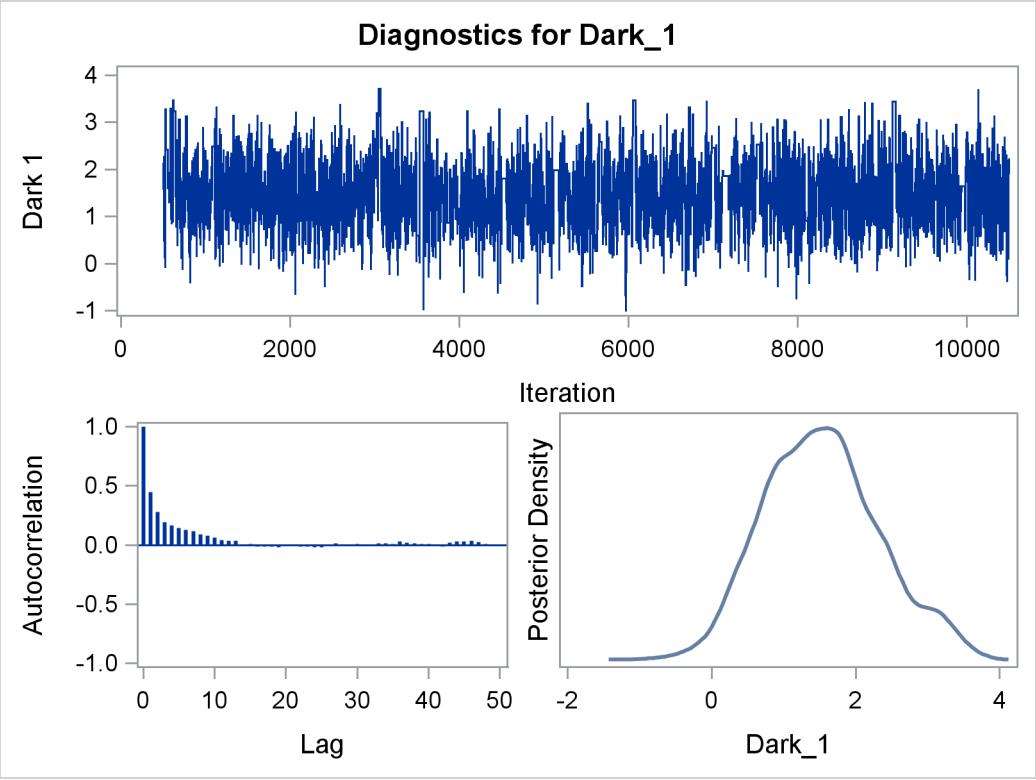
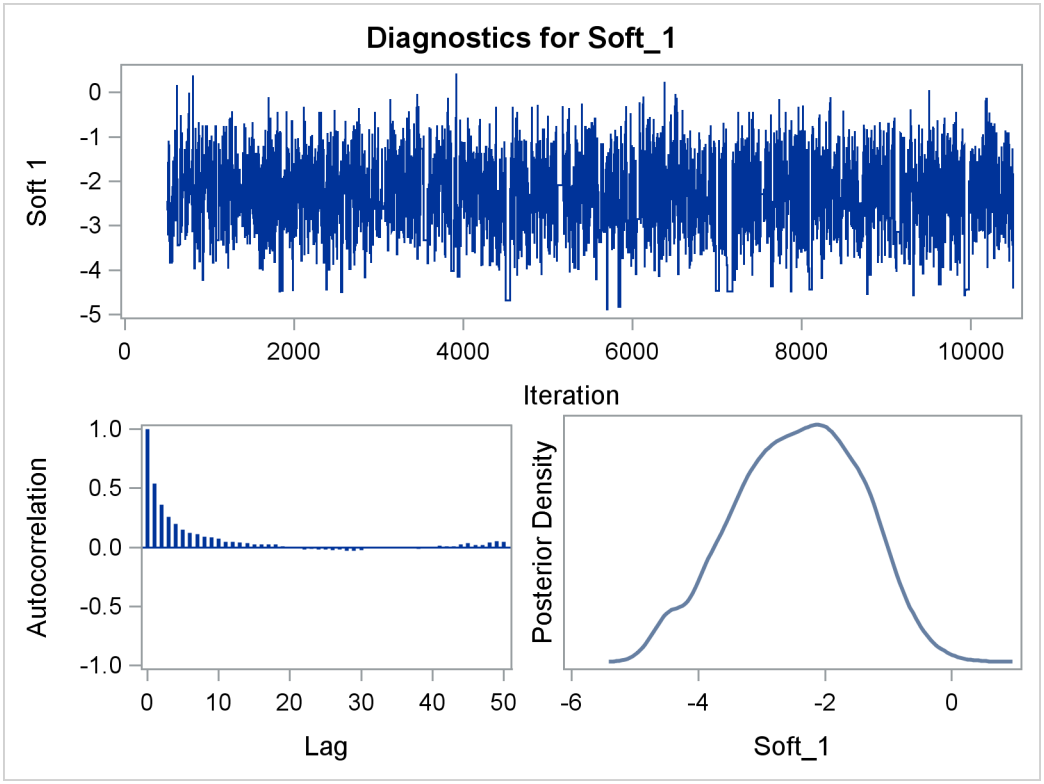
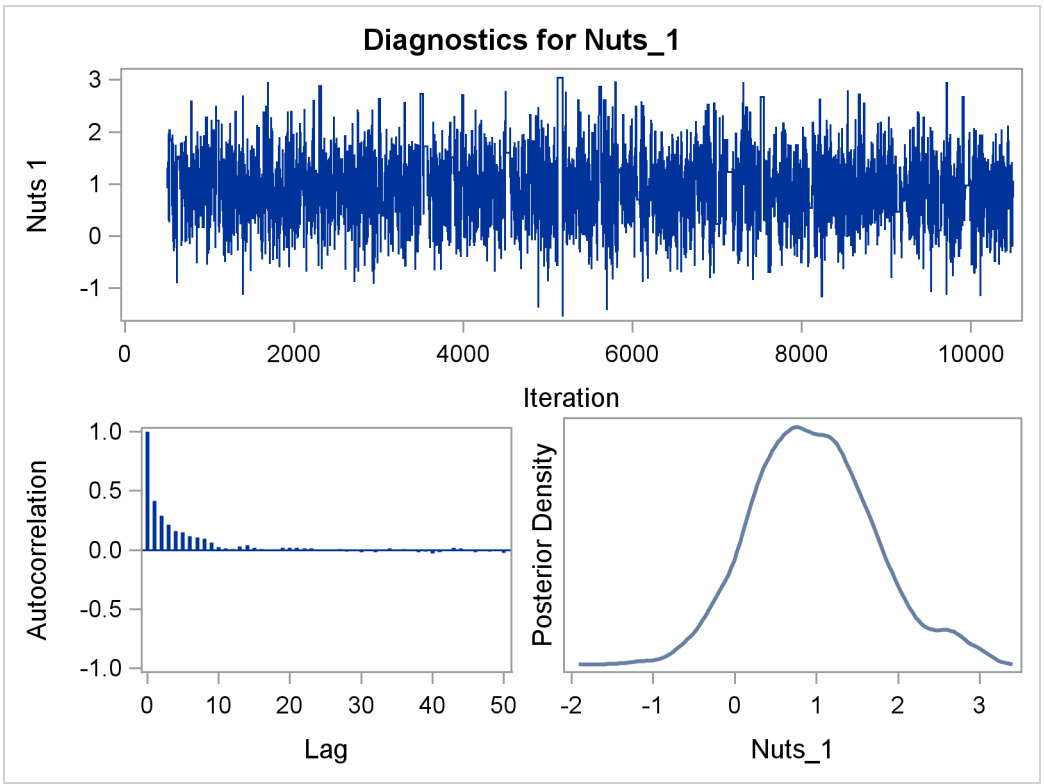


Figure 27.7 *continued*





Because the chains seem to have reached their stationary distributions, you can go back to the posterior summary table for results and conclusions. As seen in Figure 27.5, the part-worth for dark chocolate is 1.5 and the part-worth for milk chocolate (base category) is structural 0; the part-worth for soft center is –2.4 and the part-worth for chewy center is structural 0; the part-worth for containing nuts is 1.0 and the part-worth for no nuts is structural 0. A positive part-worth implies being more favorable. Hence, dark chocolate is preferred over milk chocolate, soft centers are less popular than chewy centers, and candies with nuts are more popular than candies without nuts.

A Logit Model Example with Random Effects

Choice models that have random effects (or random coefficients) provide solutions to create individual-level or group-specific utilities. Because people have different preferences, it can be misleading to roll the whole sample together into a single set of utilities. The desire to account for individual differences, instead of treating all respondents alike, provides challenges in marketing research. For logit models that have random effects, using frequentist methods to optimize of the likelihood function can be numerically difficult. Bayesian methods are ideally suited for analysis with random effects.

Choice models that have random effects generalize the standard choice models to incorporate individual-level effects. Let the utility that individual i obtains from alternative j in choice situation t ($t = 1, \ldots, T$) be

$$\begin{aligned} u_{ijt} &= \mathbf{x}'_{ijt}\boldsymbol{\beta} + \mathbf{z}'_{ijt}\boldsymbol{\gamma}_i + \epsilon_{ijt} \\ y_{ijt} &= 1 \text{ if } u_{ijt} \geq \max(u_{i1t}, u_{i2t}, \ldots, u_{iJt}) \\ &= 0 \text{ otherwise} \end{aligned}$$

where y_{ijt} is the observed choice for individual i and alternative j in choice situation t; $\mathbf{x}_{ijt}$ is the fixed design vector for individual i and alternative j in choice situation t; $\boldsymbol{\beta}$ are the fixed coefficients; $\mathbf{z}_{ijt}$ is the random design vector for individual i and alternative j in choice situation t; and $\boldsymbol{\gamma}_i$ are the random coefficients for individual i corresponding to $\mathbf{z}_{ijt}$.

It is assumed that each $\boldsymbol{\gamma}_i$ is drawn from a superpopulation and that this superpopulation is normal, $\boldsymbol{\gamma}_i \sim$ iid $\mathrm{N}(\mathbf{0}, \boldsymbol{\Omega}_{\boldsymbol{\gamma}})$. An additional stage is added to the model in which a prior for $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$ is specified:

$$\begin{aligned} \pi(\boldsymbol{\gamma}_i) &= \mathrm{N}(\mathbf{0}, \boldsymbol{\Omega}_{\boldsymbol{\gamma}}) \\ \pi(\boldsymbol{\Omega}_{\boldsymbol{\gamma}}) &= \text{inverse Wishart}(\nu_0, \mathbf{V}_0) \end{aligned}$$

The covariance matrix $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$ characterizes the extent of heterogeneity among individuals. Large diagonal elements of $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$ indicate substantial heterogeneity in part-worths. Off-diagonal elements indicate patterns in the evaluation of attribute levels in pairs.

Consider a study that estimates the market demand for kitchen trash cans (Rossi 2013). There are four attributes, and each has two levels: touchless opening (Yes/No), material (Steel/Plastic), automatic trash bag replacement (Yes/No), and price (80/40). The number of all possible hypothetical types of trash cans is $2^4 = 16$. Including more attributes and more levels can easily become unmanageable. The study uses a fractional factorial design, in which the first three factors are set up to be a full factorial design and the fourth is generated as the product of the first three. This design confounds the three-way interaction with the effect of the fourth factor, shown in Table 27.1.

Table 27.1 Design for the Trash Can Study

Obs	Touchless	Steel	AutoBag	Price80
1	–1	–1	–1	–1
2	–1	–1	1	1
3	–1	1	–1	1
4	–1	1	1	–1
5	1	–1	–1	1
6	1	–1	1	–1
7	1	1	–1	–1
8	1	1	1	1

In Table 27.1, 1 means "Yes" and –1 means "No." This is a balanced design, in which each level appears the same number of times. This study assigns only two alternatives to a choice set by randomly sampling two rows from the previous table and giving each individual 10 choice sets (or choice tasks) to pick from. For more information about how to design a choice model efficiently, see Kuhfeld (2010).

Data were obtained by enrolling 104 people and assigning 10 choice tasks to each of them: for each task, the participants stated their preference between two types of trash cans. The following steps read in the data:

```
data Trashcan;
   input ID Task Choice Index Touchless Steel AutoBag Price80 @@;
   datalines;
1 1 1 1 0 1 1 0 1 1 0 2 1 1 0 0 1 2 0 1 0 0 0 0 1 2 1 2 1 1 1 1 1 3 0 1
0 0 0 0 1 3 1 2 1 1 0 0 1 4 0 1 0 1 0 1 1 4 1 2 1 0 0 1 1 5 1 1 0 1 1 0
1 5 0 2 1 0 0 1 1 6 0 1 0 0 1 1 1 6 1 2 1 1 1 1 1 7 0 1 1 0 0 1 1 7 1 2
1 1 0 0 1 8 0 1 0 0 1 1 1 8 1 2 0 1 1 0 1 9 1 1 0 0 1 1 1 9 0 2 1 0 0 1
1 10 0 1 0 1 1 0 1 10 1 2 1 0 1 0 2 1 1 1 0 1 1 0 2 1 0 2 1 1 0 0 2 2 0
1 0 0 0 0 2 2 1 2 1 1 1 1 2 3 0 1 0 0 0 0 2 3 1 2 1 1 0 0 2 4 0 1 0 1 0
1 2 4 1 2 1 0 0 1 2 5 1 1 0 1 1 0 2 5 0 2 1 0 0 1 2 6 1 1 0 0 1 1 2 6 0
2 1 1 1 1 2 7 1 1 1 0 0 1 2 7 0 2 1 1 0 0 2 8 1 1 0 0 1 1 2 8 0 2 0 1 1
0 2 9 1 1 0 0 1 1 2 9 0 2 1 0 0 1 2 10 1 1 0 1 1 0 2 10 0 2 1 0 1 0 3 1

   ... more lines ...

2 1 2 1 1 1 1 104 3 0 1 0 0 0 0 104 3 1 2 1 1 0 0 104 4 0 1 0 1 0 1 104
4 1 2 1 0 0 1 104 5 1 1 0 1 1 0 104 5 0 2 1 0 0 1 104 6 0 1 0 0 1 1 104
6 1 2 1 1 1 1 104 7 0 1 1 0 0 1 104 7 1 2 1 1 0 0 104 8 0 1 0 0 1 1 104
8 1 2 0 1 1 0 104 9 0 1 0 0 1 1 104 9 1 2 1 0 0 1 104 10 0 1 0 1 1 0 104
10 1 2 1 0 1 0
;

proc print data=Trashcan (obs=8);
run;
```

The data for the first four choice tasks are shown in Figure 27.8.

Figure 27.8 Data for the First Four Choice Tasks

Obs	ID	Task	Choice	Index	Touchless	Steel	AutoBag	Price80
1	1	1	1	1	0	1	1	0
2	1	1	0	2	1	1	0	0
3	1	2	0	1	0	0	0	0
4	1	2	1	2	1	1	1	1
5	1	3	0	1	0	0	0	0
6	1	3	1	2	1	1	0	0
7	1	4	0	1	0	1	0	1
8	1	4	1	2	1	0	0	1

In the data, ID is the individual's ID number, and Task indexes the number of choice tasks. The response is Choice, which states each individual's choice for each choice task. Touchless, Steel, AutoBag, and Price80 are the attribute variables; for each of them, 1 means "Yes" and 0 means "No." In the data, 0 replaces the –1 values that are shown in the design matrix in Table 27.1.

The following statements fit a logit model with random effects:

```
proc bchoice data=Trashcan seed=1 nmc=30000 thin=2 nthreads=4;
   class ID Task;
   model Choice = Touchless Steel AutoBag Price80 / choiceset=(ID Task);
   random Touchless Steel AutoBag Price80 / sub=ID monitor=(1 to 5) type=un;
run;
```

The NTHREADS option in the PROC BCHOICE statement specifies the number of threads to be used for running analytic computations and simulation simultaneously. Using four threads at the same time enhances the efficiency and reduces the run time. If you do not specify the NTHREADS option, the default number is 1. The maximum number of threads should not exceed the total number of CPUs on the host where the analytic computations execute.

The choice set is specified by ID (which identifies the participants) and by Task (which identifies each of the 10 choice tasks that are assigned to each participant). The variables ID and Task are needed in the CLASS statement because they define the choice set in the MODEL statement.

In addition to the MODEL statement for fixed effects, the RANDOM statement is added for random effects. Note that Touchless, Steel, AutoBag, and Price80 are listed as both fixed and random effects, so that their average part-worth values in the population are estimated via fixed effects and the deviation from the overall mean for each individual is presented through random effects. The SUB=ID argument in the RANDOM statement defines ID as a subject index for the random effects grouping, so that each person with a different ID has his own random effects. The MONITOR=(1 to 5) option requests to display the summary and diagnostics statistics of the random-effects parameter estimates for the first five subjects. By default, PROC BCHOICE does not print the summary and diagnostics statistics for any individual-level random-effects to save time and space. In models that have a large number of individual random effects (for example, tens of thousands individuals), it may take a long time to display the summary, diagnostics statistics, and plots for all the individual-level parameters, so be cautious when using the MONITOR option without providing a subset of the random-effects parameters.

The TYPE=UN option in the RANDOM statement specifies an unstructured covariance matrix for the random effects. The unstructured type provides a mechanism for estimating the correlation between the random effects. The TYPE=VC (variance components) option, which is a diagonal matrix and is the default structure, models a different variance component for each random effect.

Summary statistics for the fixed coefficients ($\boldsymbol{\beta}$), the covariance of the random coefficients ($\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$), and the random coefficients ($\boldsymbol{\gamma}_i$) for the first five individuals are shown in Figure 27.9.

Figure 27.9 Posterior Summary Statistics

The BCHOICE Procedure

Posterior Summaries and Intervals

Parameter	Subject	N	Mean	Standard Deviation	95% HPD Interval	
Touchless		15000	1.7043	0.2709	1.1645	2.2322
Steel		15000	1.0516	0.2608	0.5376	1.5691
AutoBag		15000	2.1722	0.3583	1.4886	2.8948
Price80		15000	-4.6321	0.6720	-5.9688	-3.4046
RECov Touchless, Touchless		15000	3.1145	1.0613	1.3441	5.2576
RECov Steel, Touchless		15000	-0.4727	0.8796	-2.1884	1.2334
RECov Steel, Steel		15000	2.6312	0.9565	1.1035	4.5817
RECov AutoBag, Touchless		15000	-0.6473	0.8811	-2.3803	1.1974
RECov AutoBag, Steel		15000	0.0595	0.7078	-1.4318	1.3748
RECov AutoBag, AutoBag		15000	3.5547	1.5837	0.9611	6.7248
RECov Price80, Touchless		15000	-1.3016	1.1857	-3.8046	0.7154
RECov Price80, Steel		15000	-1.5935	1.2052	-4.1428	0.6170
RECov Price80, AutoBag		15000	-2.2190	1.7156	-5.7073	0.7349
RECov Price80, Price80		15000	7.7452	3.4362	2.2287	14.3546
Touchless	ID 1	15000	0.5244	1.0525	-1.3366	2.7890
Steel	ID 1	15000	-0.3298	1.0799	-2.3057	1.9751
AutoBag	ID 1	15000	1.6216	1.3842	-0.9207	4.4860
Price80	ID 1	15000	-0.5394	2.2018	-5.2251	3.4324
Touchless	ID 2	15000	-1.7498	0.9341	-3.5965	0.0610
Steel	ID 2	15000	-1.4343	0.8572	-3.2176	0.2048
AutoBag	ID 2	15000	0.4767	1.3208	-2.0239	3.1743
Price80	ID 2	15000	4.6873	1.4955	1.9223	7.7535
Touchless	ID 3	15000	-0.6191	0.9695	-2.4024	1.4490
Steel	ID 3	15000	0.5779	0.9524	-1.2792	2.4187
AutoBag	ID 3	15000	-1.8737	1.2107	-4.2878	0.4407
Price80	ID 3	15000	3.7507	1.6747	0.4795	7.0672
Touchless	ID 4	15000	-0.5921	0.9679	-2.5222	1.2694
Steel	ID 4	15000	0.3152	1.0416	-1.7134	2.3286
AutoBag	ID 4	15000	0.9770	1.4250	-1.7103	3.9276
Price80	ID 4	15000	-1.9068	2.3717	-6.4256	2.8233
Touchless	ID 5	15000	-1.2689	1.1491	-3.4720	0.9828
Steel	ID 5	15000	1.6581	1.2220	-0.4731	4.2409
AutoBag	ID 5	15000	1.2695	1.5708	-1.7506	4.4762
Price80	ID 5	15000	-0.3706	2.3810	-5.0593	4.2134

The fixed effects (Touchless, Steel, AutoBag, and Price80) are shown in the first four rows. Across all the respondents in the data, the average part-worths for touchless opening, steel material, and automatic trash bag replacement are all positive, indicating that most people favor those features; the average part-worth for having to pay USD80 for a trash can instead of USD40 is negative (–4.6), which is very intuitive, because spending more money is usually unfavorable.

The covariance estimate of the random coefficients ($\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$) is displayed by the parameters whose label begins with "RECov":

$$\hat{\boldsymbol{\Omega}}_{\boldsymbol{\gamma}} = \begin{pmatrix} 3.11 & . & . & . \\ -0.47 & 2.63 & . & . \\ -0.65 & 0.06 & 3.55 & . \\ -1.30 & -1.59 & -2.22 & 7.75 \end{pmatrix}$$

where the dots refer to the corresponding elements in the lower part of the symmetric covariance matrix. The covariance estimate of the random coefficients ($\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$) characterizes the variability of part-worths across respondents. Some of the diagonal elements of the matrix are large. For example, the variance for price (labeled "RECov Price80, Price80") is quite large, indicating substantial unexplained difference in response to price. Off-diagonal elements of the matrix illustrate attribute levels that tend to be evaluated similarly (positive covariance) or differently (negative covariance) across all the respondents. The covariances between each of the attributes (Touchless, Steel, and AutoBag) and Price80 are all negative, implying that the respondents who prefer some of the new features are those who are also unwilling to pay a higher price for the trash can. Therefore, offering a discounted price might be a particularly effective method of introducing the new features to customers.

The next set of parameters that are displayed are the estimates for the individual-level random effects for the first five respondents (see Figure 27.9). These estimates are the deviation from the overall means (which are estimated via the fixed effects). The part-worth for touchless opening for the first respondent (who is labeled "ID 1" in the Subject column) is 1.7 + 0.5 = 2.2.

Allenby and Rossi (1999) and Rossi, Allenby, and McCulloch (2005) propose a hierarchical Bayesian random-effects model that is set up in a different way such that there are no fixed effects but only random effects. For more information about this type of model, see the section "Random Effects" on page 1070 and a follow-up example in "Example 27.4: A Random-Effects-Only Logit Model" on page 1097.

Syntax: BCHOICE Procedure

The following statements are available in the BCHOICE procedure. Items within < > are optional.

PROC BCHOICE < *options* > **;**
BY *variables* **;**
CLASS *variable* < **(***options***)** > < *. . . variable* < **(***options***)** > > < */ options* > **;**
MODEL *response* < **(***response-options***)** > **=** < *fixed-effects* > < */ model-options* > **;**
RANDOM *random-effects* < */ options* > **;**
PREDDIST OUTPRED=*SAS-data-set* < *options* > **;**

The PROC BCHOICE and MODEL statements are required. The CLASS statement, if present, must precede the MODEL statement.

The rest of this section provides detailed syntax information for each statement, beginning with the PROC BCHOICE statement. The remaining statements are presented in alphabetical order.

PROC BCHOICE Statement

PROC BCHOICE <*options*> **;**

The PROC BCHOICE statement invokes the BCHOICE procedure.

Table 27.2 summarizes the *options* available in the PROC BCHOICE statement.

Table 27.2 PROC BCHOICE Statement Options

Option	Description
Basic Options	
DATA=	Names the 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
NTU=	Specifies the number of tuning iterations for the random walk sampler
OUTPOST=	Names the output data set to contain posterior samples of parameters
SEED=	Specifies the random seed for simulation
THIN=	Specifies the thinning rate
Sampling, Summary, Diagnostics, and Plotting options	
ALGORITHM=	Specifies the algorithm to use to sample the posterior distribution
DIAGNOSTICS=	Controls the convergence diagnostics
DIC	Computes the deviance information criterion (DIC)
HITPROB	Outputs the average of estimated probabilities of chosen alternatives in the input data
PLOTS=	Controls plotting
STATISTICS=	Controls posterior statistics

Table 27.2 (continued)

Option	Description
Other Options	
ACCEPTTOL=	Specifies the acceptance rate tolerance for the random walk sampler
INF=	Specifies the machine numerical limit for infinity
LOGPOST	Calculates the logarithm of the posterior density and likelihood
MAXTUNE=	Specifies the maximum number of tuning loops for the random walk sampler
MCHISTORY=	Displays Markov chain sampling history
MINTUNE=	Specifies the minimum number of tuning loops for the random walk sampler
NOCLPRINT	Suppresses the "Class Level Information" table completely or partially
SCALE=	Specifies the initial scale applied to the proposal distribution for the random walk sampler
TARGACCEPT=	Specifies the target acceptance rate for the random walk sampler
TUNEWT=	Specifies the weight used in covariance updating for the random walk sampler

You can specify the following *options*.

ACCEPTTOL=*n*

specifies a tolerance for acceptance probabilities for the random walk sampler. By default, ACCEPTTOL=0.075. You can specify this option for logit models.

ALGORITHM=GAMERMAN | RWM | LATENT

ALG=GAMERMAN | RWM | LATENT

specifies the algorithm to use to sample the posterior distribution for the regression coefficients. You can specify the following algorithms:

GAMERMAN

uses the Metropolis-Hastings approach of Gamerman (1997). This is the default for logit models.

RWM

uses the random walk Metropolis algorithm along with the normal proposal, as suggested in Rossi, Allenby, and McCulloch (2005).

LATENT

uses the latent variables via the data augmentation method as proposed in McCulloch and Rossi (1994). This approach avoids direct evaluation of the likelihood. This is the default for probit models. This option is ignored for logit models.

When possible, PROC BCHOICE samples directly from the full conditional distribution. Otherwise, the default sampling algorithm is the Gamerman algorithm for standard logit models, the random walk Metropolis algorithm for nested logit models, and the latent variables via the data augmentation method for probit models. Standard logit models can also use random walk Metropolis sampling algorithm if specified, while the other two types of models, nested logit models and probit models, have only their own default sampling algorithm.

DATA=*SAS-data-set*
specifies the input data set.

DIAGNOSTICS=NONE | (*keyword-list***)**
DIAG=NONE | (*keyword-list***)**
specifies options for convergence diagnostics. By default, PROC BCHOICE computes the effective sample sizes. The sample autocorrelations, Monte Carlo errors, Geweke test, Raftery-Lewis test, and Heidelberger-Welch test are also available. For more information about convergence diagnostics, see the section "Assessing Markov Chain Convergence" on page 136 in Chapter 7, "Introduction to Bayesian Analysis Procedures." 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 the option ALL 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* must take 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. For more information, see the section "Effective Sample Size" on page 149 in Chapter 7, "Introduction to Bayesian Analysis Procedures."

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. For more information, see the section "Geweke Diagnostics" on page 143 in Chapter 7, "Introduction to Bayesian Analysis Procedures." 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 Heidelberger-Welch diagnostic (which consists of a stationarity test and a half-width 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 half-width test is then carried out. For more information, see the section "Heidelberger and Welch Diagnostics" on page 145 in Chapter 7, "Introduction to Bayesian Analysis Procedures."

You can also specify suboptions, such as DIAGNOSTICS=HEIDELBERGER(EPS=0.05), as follows:

EPS=*value*

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

HALPHA=*value*

specifies the α level ($0 < \alpha < 1$) for the half-width 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=*n*

specifies the maximum number of autocorrelation lags to use to compute the effective sample size; for more information, see the section "Effective Sample Size" on page 149 in Chapter 7, "Introduction to Bayesian Analysis Procedures." The value of *n* is also used in the calculation of the Monte Carlo standard error; see the section "Standard Error of the Mean Estimate" on page 150 in Chapter 7, "Introduction to Bayesian Analysis Procedures." By default, MAXLAG=MIN(500, MCsample/4), where MCsample is the Markov chain sample size that is kept after thinning—that is, MCsample $= \left[\frac{\text{NMC}}{\text{NTHIN}}\right]$. If *n* 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 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, $\Pr(\text{Q}-\text{R} \leq \hat{\text{P}}_Q \leq \text{Q}+\text{R}) = \text{S}$. For more information, see the section "Raftery and Lewis Diagnostics" on page 146 in Chapter 7, "Introduction to Bayesian Analysis Procedures."

You can specify Q, R, S, and a precision level ϵ for a stationary 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 stationary 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. For more information, see the section "Deviance Information Criterion (DIC)" on page 152 in Chapter 7, "Introduction to Bayesian Analysis Procedures."

HITPROB

HITPROBABILITY

calculates the average of estimated probabilities of all chosen alternatives in the input data set. You can use this average as a measure of goodness of fit.

INF=*value*

specifies the numerical definition of infinity in PROC BCHOICE. By default, INF=1E15. For example, PROC BCHOICE considers 1E16 to be outside the support of the normal distribution and assigns a missing value to the log density evaluation. The minimum *value* that is allowed is 1E10, and the maximum *value* that is allowed is 1E27.

LOGPOST

calculates the logarithm of the posterior density of the parameters and the likelihood at each iteration. As a result, the OUTPOST= data set will contain the LOGLIKE and LOGPOST variables. You can specify this option for logit models and nested logit models, but not probit models.

MAXTUNE=*n*

specifies an upper limit for the number of proposal tuning loops for the random walk sampler. You can specify this option for logit models; it is ignored for other models. By default, MAXTUNE=6.

MCHISTORY=BRIEF | DETAILED | NONE

MCHIST=BRIEF | DETAILED | NONE

controls the display of the Metropolis sampling history. This option is ignored for nested logit and probit models. You can specify the following values:

BRIEF
produces a summary output for the tuning, burn-in, and sampling history tables. No tuning history table is produced if there is no tuning stage. The tables show the following when applicable:

- Scale shows the scale, or the range of the scales, that is used in each random walk Metropolis block that has a normal distribution.
- Acceptance Rate shows the acceptance rate, or the range of the acceptance rates, for each Metropolis block.

DETAILED
produces detailed output of the tuning, burn-in, and sampling history tables, including scale values, acceptance probabilities, and blocking information. No tuning history table is produced if there is no tuning stage. Use this option with caution, especially in random-effects models that have a large number of random-effects groups, because it can produce copious output.

NONE
produces none of the tuning history, burn-in history, and sampling history tables.

By default, MCHISTORY=BRIEF.

MINTUNE=*n*
specifies a lower limit for the number of proposal tuning loops for the random walk sampler. You can specify this option for logit models; it is ignored for other models. By default, MINTUNE=2.

NBI=*n*
specifies the number of burn-in iterations to perform before beginning to save parameter estimate chains. By default, NBI=500. For more information, see the section "Burn-In, Thinning, and Markov Chain Samples" on page 135 in Chapter 7, "Introduction to Bayesian Analysis Procedures."

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

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

NTHREADS=*n*
NTHREAD=*n*
specifies the number of threads(CPUs) for analytic computations and simulation to run simultaneously on. Multi-threading is the concurrent execution of threads. When multi-threading is possible, you can realize substantial performance gains compared to the performance you get from sequential (single-threaded) execution. The more threads there are, the faster the computation runs. But do not specify a number that is more than the number of CPUs on the host where the analytic computations execute.

PROC BCHOICE performs two types of threading. In sampling fixed-effects parameters, PROC BCHOICE 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. If you do not specify the NTHREADS= option or if you specify NTHREADS=0, the default number of threads is 1.

NTU=*n*
specifies the number of iterations to use in each proposal tuning phase for the random walk sampler. You can specify this option for logit models; it is ignored for other models. By default, NTU=500.

OUTPOST=*SAS-data-set*

OUT=*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-options***)> <= ***plot-request* **<(***options***)>>**

PLOTS <(*global-plot-options***)> <= (***plot-request* **<(***options***)> <... ***plot-request* **<(***options***)>>)>**
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 but you do not specify the PLOTS= option, then PROC BCHOICE 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:

TYPE
groups the plots by type.

PARAMETER
groups the plots by parameter.

LAGS=*n*
specifies the number of autocorrelation lags that are used 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 AUTOCORR 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(X3)	Autocorr(X2) 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)

SCALE=*value*
controls the initial multiplicative scale to the covariance matrix of the proposal distribution for the random walk–based Metropolis algorithm for logit models. By default, SCALE=2.93. For more information, see the section "Scale Tuning" on page 1080.

SEED=*n*
specifies the random number seed. By default, SEED=0, and PROC BCHOICE gets a random number seed from the system clock. Negative seed values are treated as the default. The largest possible value for the seed is $2^{31} - 1$. Analyses that use the same nonzero seed are reproducible. The seed value is reported in the "Model Information" table.

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

STATS <(*global-stats-options***)> = NONE | ALL |** *stats-request*
specifies options for posterior statistics. By default, PROC BCHOICE 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. For more information, see the section "Summary Statistics" on page 150 in Chapter 7, "Introduction to Bayesian Analysis Procedures." 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. 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 ALPHA= option to request 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. For more information, see the sections "Equal-Tail Credible Interval" on page 151 and "Highest Posterior Density (HPD) Interval" on page 151 in Chapter 7, "Introduction to Bayesian Analysis Procedures." By default, ALPHA=0.05, but you can use the global ALPHA= option to request 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 PERCENT= option to request specific percentile points.

TARGACCEPT=*value*

specifies the target acceptance rate for the random walk–based Metropolis algorithm for logit models. For more information, see the section "Metropolis and Metropolis-Hastings Algorithms" on page 130 in Chapter 7, "Introduction to Bayesian Analysis Procedures." The numeric *value* must be between 0.01 and 0.99. By default, TARGACCEPT=0.45 for one parameter; TARGACCEPT=0.35 for two, three, or four parameters; and TARGACCEPT=0.234 for more than four parameters (Roberts, Gelman, and Gilks 1997; Roberts and Rosenthal 2001).

THIN=*n*

NTHIN=*n*

controls the thinning rate of the simulation. PROC BCHOICE 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. For more information, see the section "Burn-In, Thinning, and Markov Chain Samples" on page 135 in Chapter 7, "Introduction to Bayesian Analysis Procedures."

TUNEWT=*value*

specifies the multiplicative weight used in updating the covariance matrix of the proposal distribution for the random walk sampler for logit models. The numeric *value* must be between 0 and 1. By default, TUNEWT=0.75. For more information, see the section "Covariance Tuning" on page 1081.

BY Statement

BY *variables* **;**

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

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

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the BCHOICE 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 specified in the CLASS statement. If you specify more than one CLASS statement, the *global-options* 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 both the DESCENDING and ORDER= options are specified, PROC BCHOICE 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 for 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 the ORDER= 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 BCHOICE 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=REF. 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 *keyword* 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.

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*. The default is REF=LAST.

FIRST designates the first ordered level as reference.

LAST designates the last ordered level as reference.

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.

MODEL Statement

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

The MODEL statement is required; it defines the response (dependent) variable and the fixed effects. The response variable indicates the chosen alternative in a choice set by the value 1 and the unchosen alternatives by the value 0 when the response is binary. In the MaxDiff choice experiment, a respondent can pick one best item and one worst item in each choice set, where there are three valid integer values: 1 for the best alternative, –1 for the worst alternative, and 0 for the ones in the middle. In the allocation choice setting, the response takes a percent of preference for each alternative and the sum of all percents in each choice set is 100%. Any value of the response variable that is valid, including missing, will be replace by zero. The

fixed-effects determine the **X** matrix of the model. The specification of effects is the same as in the GLIMMIX and MIXED procedures, where you do not specify random effects in the MODEL statement.

Table 27.3 summarizes the options available in the MODEL statement. These are subsequently discussed in detail in alphabetical order by option category.

Table 27.3 MODEL Statement Options

Option	Description
ALTERIDX=	Specifies the order of all alternatives in a choice set
CHOICESET=	Specifies the variables for defining a choice set
CHOICETYPE=	Specifies what values the choice response variable takes
COEFFPRIOR=	Specifies the prior of the regression coefficients
COVPRIOR=	Specifies the prior of the covariance parameter for a probit model
COVTYPE=	Specifies the structure of the covariance matrix of the error difference for a probit model
INIT=	Controls the generation of initial values of the regression coefficients
LAMBDAPRIOR=	Specifies the prior of the log-sum coefficients for a nested logit model
NEST=	Defines the nonoverlapping nests for a nested logit model
SAMELAMBDA	Constrains the log-sum coefficients to be the same for all the nests in a nested logit model
TYPE=	Specifies the type of the model

ALTERIDX=*variable*

specifies the order of all alternatives in a choice set for a nested logit or probit model. Specify this option to make sure the alternatives appear in the same order in all choice sets. When you do not specify this option, PROC BCHOICE assumes that the alternatives are in the same order. The *variable* should take increasing positive integers starting from 1 within each choice set. You need to specify this option only for a nested logit or probit model; PROC BCHOICE ignores it for a logit model.

CHOICESET=(*variables***)**

specifies one or more *variables* for defining the choice sets. You must specify how the choice sets are constructed, and you can use more than one variable. PROC BCHOICE does not sort by the values of the choice set variable; rather, it considers the data to be from a new choice set whenever the value of the choice set variable changes from the previous observation.

CHOICETYPE=*choice-type*

specifies what values the choice outcome variable takes. You can specify the following *choice-types*:

BINARY

specifies that the choice response variable has a binary integer value: 1 for a chosen alternative and 0 for an unchosen alternative. Each choice set has at least two alternatives and only one chosen alternative.

MAXDIFF

specifies that the choice response variable takes one of the possible three integer values: 1 for the best alternative, –1 for the worst alternative, and 0 for the ones in the middle. Each respondent is shown a set of the possible items to be evaluated and chooses one best item and one worst item in each choice set.

ALLOCATION < (WEIGHT=*n***) >**

specifies that the choice response variable takes a percentage of preference for each alternative and that the sum of percentages in each choice set is 100%. You can specify the following suboption:

WEIGHT=*n*

specifies a positive integer to indicate how many choice sets the allocated percentages should apply to. For example, if the respondent were asked to imagine the next 20 purchase occasions and to allocate a percentage of those 20 purchases for each alternative, you would use a weight of 20. By default, WEIGHT=10.

By default, CHOICETYPE=BINARY.

COEFFPRIOR=NORMAL < (*options***) >**

CPRIOR=NORMAL < (*options***) >**

specifies the prior distribution for the regression coefficients. The default is the normal prior $N(\mathbf{0}, 10^2\mathbf{I})$, where $\mathbf{I}$ is the identity matrix. You can specify the following *options*, 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 covariances 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 BCHOICE 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 with _TYPE_='VAR'; alternatively, the precisions (inverse of the variances) can be specified with _TYPE_='PRECISION'.

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

COVPRIOR=IWISHART < (*options*) **>**
specifies an inverse Wishart prior distribution, **IWISHART**(*a*,*b*), for the covariance matrix for the vector of error differences. For models that do not have a covariance matrix for the error differences (the logit and nested logit models), this option is ignored.

You can specify the following *options*, enclosed in parentheses:

DF=*a*
specifies the degrees of freedom of the inverse Wishart distribution. The default is the number of alternatives in the choice set plus 2, which is equivalent to the dimension of the covariance matrix of the error differences plus 3.

SCALE=*b*
specifies $b\mathbf{I}$ for the scale parameter of the inverse Wishart distribution, where $\mathbf{I}$ is the identity matrix. The default is the number of alternatives in the choice set plus 2.

COVTYPE=UN | VC
specifies the covariance structure of the error difference vector for a probit model. Although a variety of structures are available, most applications call for either COVTYPE=VC or COVTYPE=UN for the error difference vector. The COVTYPE=VC (variance components) models a different variance component for each error term. The TYPE=UN (unstructured) specifies a full structured covariance matrix. The unstructured form accommodates any pattern of correlation in addition to fitting a different variance component for each error difference term.

INIT=*keyword-list* **|** (*numeric-list*)
INITIAL=*keyword-list* **|** (*numeric-list*)
specifies options for generating the initial values for the coefficients parameters that are specified as *fixed-effects* in the MODEL statement. By default, INIT=POSTMODE for logit models and INIT=PRIORMODE for probit models. You can specify the following *keywords*:

LIST=*numeric-list*
assigns the numbers to be the initial values of the fixed effects in the corresponding list order. The length of the 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 / choiceset=(ID Index) init=(list=(1 2 3) pinit);
```

If the length of the list is less than the number of fixed effects, the initial value of each remaining parameter will be replaced by the corresponding default initial value. For example, the corresponding mode of the posterior density is used for a logit model. If the length of the list is greater than the number of fixed effects, the extra ones are ignored.

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

POSTMODE
uses the mode of the posterior density as the initial value of the parameter, if you do not provide one. If the mode does not exist or if it is on the boundary of the support of the density, the mean value is used. If you specify POSTMODE for a probit model, where the posterior density is difficult to obtain, PROC BCHOICE resets it to PRIORMODE.

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

LAMBDAPRIOR=SEMIFLAT <(*options*)>
LPRIOR=SEMIFLAT <(*options*)>
specifies a semi-flat prior distribution (Lahiri and Gao 2002) for the log-sum coefficient, λ, for each nest in a nested logit model. For models that are not nested logit, this option is ignored.

You can specify the following *option*, enclosed in parentheses:

PHI=*a*
specifies the parameter ϕ of the semi-flat prior. By default, $\phi = 0.8$.

NEST=(*numeric-list***)**
defines the nonoverlapping nests for a nested logit model. For a nested logit model, you must specify the nests for all the alternatives in the choice set. Otherwise, the standard logit model is assumed. The number of values in the list should match the number of alternatives in the choice sets, and each of the actual values represents the nest that the particular alternative goes to. For example, NEST=(1 2 1 1 2) arranges the first, third, and fourth alternatives in the first nest and the second and fifth alternatives in the second nest. Currently, this option can accommodate only two-level nested logit models.

SAMELAMBDA
constrains the log-sum coefficients to be the same for all the nests in a nested logit model.

TYPE=*keyword*
specifies the type of the model. You can specify the following *keywords*:

LOGIT
specifies a standard logit model.

NLOGIT
specifies a nested logit model. If you do not also specify the NEST= option to define the nests, this option is ignored, and a standard logit model is fit.

PROBIT
specifies a probit model.

PREDDIST Statement

PREDDIST OUTPRED=*SAS-data-set* <**COVARIATES=***SAS-data-set*> **;**

The PREDDIST statement creates a new SAS data set that contains random samples from the posterior predictive distribution of the choice probabilities. It enables you to get the expected choice probabilities of all the alternatives in a choice set.

The posterior predictive distribution is the distribution of unobserved observations (prediction) conditional on the observed data. Let $\mathbf{Y}$ be the observed data, $\mathbf{X}$ be the covariates, $\boldsymbol{\theta}$ be the parameter, and $\mathbf{Y}_{\text{pred}}$ be the unobserved data. The posterior predictive distribution is defined as follows:

$$\begin{aligned} p(\mathbf{Y}_{\text{pred}}|\mathbf{Y},\mathbf{X}) &= \int p(\mathbf{Y}_{\text{pred}},\boldsymbol{\theta}|\mathbf{Y},\mathbf{X})d\boldsymbol{\theta} \\ &= \int p(\mathbf{Y}_{\text{pred}}|\boldsymbol{\theta},\mathbf{Y},\mathbf{X})p(\boldsymbol{\theta}|\mathbf{Y},\mathbf{X})d\boldsymbol{\theta} \end{aligned}$$

Assuming that the observed and unobserved data are conditionally independent given $\boldsymbol{\theta}$, the posterior predictive distribution can be further simplified as follows:

$$p(\mathbf{Y}_{\text{pred}}|\mathbf{Y},\mathbf{X}) = \int p(\mathbf{Y}_{\text{pred}}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{Y},\mathbf{X})d\boldsymbol{\theta}$$

The posterior predictive distribution is an integral of the likelihood function $p(\mathbf{Y}_{\text{pred}}|\boldsymbol{\theta})$ with respect to the posterior distribution $p(\boldsymbol{\theta}|\mathbf{Y})$. The PREDDIST statement generates samples from a posterior predictive distribution based on draws from the posterior distribution of $\boldsymbol{\theta}$.

You can specify the following options:

COVARIATES=*SAS-data-set*
: names the SAS data set to contain the sets of explanatory variable values for which the predictions are established. This data set must contain data that has the same variables used in the model. If you omit the COVARIATES= option, the DATA= data set that is specified in the PROC BCHOICE statement is used instead.

NALTER=*n*

NALTERNATIVE=*n*
: specifies the number of alternatives in a choice set in the COVARIATES= data set. All choice sets in the data must have the same number of alternatives. You must specify this option if a COVARIATES= data set is given.

OUTPRED=*SAS-data-set*
: creates an output data set to contain the samples from the posterior predictive distribution of the choice probability that each alternative is chosen from a choice set. The output data set are in the order of either the COVARIATES= data set or the DATA= data set specified in the PROC statement. Multi-threading and data deletion might cause the order to change.

RANDOM Statement

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

The RANDOM statement defines the random effects in the choice model. You can use this statement to specify traditional variance component models and to specify random coefficients, which can be classification or continuous. You can specify only one RANDOM statement. Include the TYPE= option to define the covariance structure. PROC BCHOICE does not include the intercept in the RANDOM statement.

Table 27.4 summarizes the *options* available in the RANDOM statement. All *options* are subsequently discussed in alphabetical order.

Table 27.4 Summary of RANDOM Statement Options

Option	Description
COVPRIOR=	Specifies the prior distribution for the covariance of the random effects
PRINTCOV=	requests to always output the summary, diagnostics statistics and plots for the covariance of the random effects.
MONITOR	Displays the summary, diagnostics statistics and plots of individual random effects
NOOUTPOST	Suppresses saving of the posterior samples of individual random effects to the OUTPOST= data set
REMEAN	Models the prior mean of the random effects
SUBJECT=	Identifies the subjects in the model
TYPE=	Specifies the covariance structure

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

COVPRIOR=IWISHART <(*options*)>
specifies an inverse Wishart prior distribution, **IWISHART**(*a*,*b*), for the covariance matrix of the random effects.

You can specify the following *options*, enclosed in parentheses:

DF=*a*
specifies the degrees of freedom of the inverse Wishart distribution. The default is the dimension of the covariance matrix of the random effects plus 3.

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

PRINTCOV
requests to always output the summary, diagnostics statistics and plots for the parameters in the covariance matrix of the random effects. By default, PROC BCHOICE produces the summary, diagnostics statistics and plots for the parameters in the covariance matrix of the random effects. However, if there are too many parameters, greater than 36, in the covariance matrix of random effects, then the summary, diagnostics statistics and plots are not displayed. You can use this option to have them printed if you would like to.

MONITOR

MONITOR=(*numeric-list*)

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

You can monitor a subset of the random-effects parameters. You can provide a numeric list of the SUBJECT indexes, or PROC BCHOICE 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);
```

PROC BCHOICE outputs results of random effects for subjects 1, 3, 5, 23, and 57. PROC BCHOICE 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));
```

PROC BCHOICE 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 BCHOICE statement.

When you specify the MONITOR option, it uses the specification of the STATISTICS= and PLOTS= options in the PROC BCHOICE statement.

NOOUTPOST

suppresses storing the posterior samples of individual random-effects parameters in the OUTPOST= data set. By default, PROC BCHOICE outputs the posterior samples of all random-effects parameters to the OUTPOST= data set. You can use the NOOUTPOST option to not save the random-effects parameters. In models that have a large number of individual random-effects (for example, tens of thousands individuals), PROC BCHOICE 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 BCHOICE outputs the list of variables that are monitored.

There is a limit on the maximum number of variables that can be saved 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 BCHOICE 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.

REMEAN

requests to estimate the mean of the individual random effects, so that the mean of the random effects, $\bar{\boldsymbol{\gamma}}$, is not assumed to be zero and is estimated. This option is recommended when there are no fixed effects specified in the MODEL statement.

You can also model the mean of the random effects as a function of individual characteristics, such as demographic variables. Rossi, McCulloch, and Allenby (1996) and Rossi, Allenby, and McCulloch (2005) propose adding another layer of flexibility to the random-effects-only model by allowing heterogeneity that is driven by observable (demographic) characteristics of the individuals. Effects specified in the REMEAN= option are not supposed to change within each SUBJECT= level in the RANDOM statement. PROC BCHOICE takes only the first value of all effects in the REMEAN= option within each SUBJECT= level.

The following REMEAN=(AGE GENDER) option in the RANDOM statement estimates the mean of the random effect, X, and models the mean as a function of Age and Gender:

```
random X / subject=Index remean=(Age Gender);
```

SUBJECT=*effect*

SUB=*effect*

identifies the subjects in the model for the random effects. A set of random effects will be estimated for each subject. PROC BCHOICE assumes complete independence across subjects; thus, for the RANDOM statement, the SUBJECT= option produces a block-diagonal structure that has identical blocks. Specifying a subject effect is equivalent to nesting all other effects in the RANDOM statement within the subject effect.

The *effect* in the RANDOM statement must be specified in the CLASS list. The *effect* can be continuous variables.

PROC BCHOICE does not sort by the values of the SUBJECT= variable; rather, it considers the data to be from a new subject or group whenever the value of the variable changes from the previous observation.

TYPE=UN | VC

specifies the covariance structure. Although a variety of structures are available, most applications call for either TYPE=VC or TYPE=UN. The TYPE=VC (variance components) option, which is the default structure, models a different variance component for each random effect. The TYPE=UN (unstructured) specifies a full structured covariance matrix for the random effects. The unstructured form accommodates any pattern of correlation between the random effects in addition to fitting a different variance component for each random effect.

RESTRICT Statement

RESTRICT < *'label'* > *fixed-effect operand operator* < *value* > **;**

RESTRICT < *'label'* > *fixed-effect constraint-list* **;**

The RESTRICT statement enables you to specify restrictions on a *fixed-effect* that is specified in a preceding MODEL statement for a logit model. You can specify either a boundary requirement or order constraints on the *fixed-effect*. Specify one RESTRICT statement for each boundary requirement or order constraint. You must specify all boundary requirements before any order constraints. PROC BCHOICE incorporates the specified restrictions in the analysis. The RESTRICT statement is not available for a nested logit or probit model, nor for any random effects.

For either type of restriction, you can specify an optional *label*, surrounded by single quotation marks, and you must specify the following argument:

fixed-effect

specifies a *fixed-effect* that matches the one specified in the preceding MODEL statement. You can specify only one *fixed-effect*. Use multiple RESTRICT statements if you need to specify multiple restrictions for a single *fixed-effect* or if you need to specify restrictions on multiple *fixed-effects*.

Use the following syntax to specify a boundary requirement:

RESTRICT < *'label'* > *fixed-effect operand operator* < *value* > **;**

You must specify the following arguments in addition to the *fixed-effect*:

operand
specifies what the operator applies to. For a continuous *fixed-effect*, specify 1; for a classification *fixed-effect*, specify a string of 0's and one 1, where 1 indicates the level of the *fixed-effect* that the operator applies to.

operator
specifies one of the following mathematical operators: **>**, **<**, **>=**, **<=**.

You can also specify the following optional argument:

value
specifies a *value* to which the operator applies. The default is 0.

The following statements specify three boundary requirements:

```
proc bchoice;
   class Brand Subj;
   model choice = Brand Price / choiceset = (Subj);
   restrict Brand 1 0 0 0 > 0;
   restrict Brand 0 1 0 0 < 1;
   restrict Price 1 < -1;
run;
```

The first RESTRICT statement restricts the part-worth (regression coefficient) that is associated with the first level of the variable Brand to be positive. The second RESTRICT statement restricts the part-worth that is associated with the second level of the variable Brand to be less than 1. The third RESTRICT statement restricts the part-worth that is associated with the variable Price to be less than –1. You can specify a boundary restriction for only one parameter in one RESTRICT statement. If you want to specify both an upper bound and a lower bound for a parameter, use two RESTRICT statements.

Use the following syntax to specify an order constraint:

RESTRICT < *'label'* > *fixed-effect constraint-list* **;**

You must specify the following argument in addition to the *fixed-effect*:

constraint-list
specifies a list of order constraints for the classification variable that is specified in the preceding CLASS statement. The *constraint-list* can contain integers 0 through the total number of estimable levels for the classification variable, where 0 indicates a level that is ignored, 1 indicates the level that has the smallest part-worth (regression coefficient), 2 indicates the level that has the second-smallest part-worth, and so on.

In the following statements, the RESTRICT statement specifies that the first level of the variable Brand has the second-smallest part-worth, the second level has the smallest part-worth, the third level is ignored, and the fourth level has the largest part-worth:

```
proc bchoice;
   class Brand Subj;
   model choice = Brand Price / choiceset = (Subj);
   restrict Brand 2 1 0 3;
run;
```

Because reference coding is used, the fifth level of the variable is the reference level and is normalized to 0. No order index is needed for the reference level.

When you specify a constraint for a *fixed-effect*, the corresponding parameter is transformed to an unconstrained variable behind the scenes. The transformation is based on the constraint.

For example, if a *fixed-effect* β is specified to have a lower bound a, then β is transformed to an unbounded variable α:

$$\alpha = \log(\beta - a)$$

The inverse of the lower-bound transformation maps an unbounded variable α to a variable β that is bounded below by a:

$$\beta = \exp(\alpha) + a$$

The normal prior distribution is assumed on the unconstrained parameters. PROC BCHOICE samples the unconstrained parameters first and then exponentiates the unconstrained parameters when they enter the utility function.

Suppose you specify the following order constraint on a set of *fixed-effects*:

$$\beta_1 < \beta_2 < \ldots < \beta_K$$

PROC BCHOICE maps them to an unconstrained set of variables $(\alpha_1, \alpha_2, \ldots, \alpha_K)$ by setting

$$\alpha_k = \begin{cases} \beta_1 & \text{if } k = 1 \\ \log(\beta_k - \beta_{k-1}) & \text{if } 1 < k \leq K \end{cases}$$

Then, the inverse of this transformation is defined by the recursion:

$$\beta_k = \begin{cases} \alpha_1 & \text{if } k = 1 \\ \exp(\alpha_k) + \beta_{k-1} & \text{if } 1 < k \leq K \end{cases}$$

Details: BCHOICE Procedure

Discrete Choice Models

Discrete choice models are used in marketing research to model decision makers' choices among alternative products and services. The decision makers might be people, households, companies and so on, and the alternatives might be products, services, actions, or any other options or items about which choices must be made (Train 2009). The collection of alternatives that are available to the decision makers is called a choice set.

Discrete choice models are derived under the assumption of utility-maximizing behavior by the decision maker. When individuals are asked to make one choice among a set of alternatives, they usually determine

the level of utility that each alternative offers. The utility that individual i obtains from alternative j among J alternatives is denoted as

$$u_{ij} = v_{ij} + \epsilon_{ij}, \quad i = 1, \ldots, N, \text{ and } j = 1, \ldots, J$$

where the subscript i is an index for the individuals, the subscript j is an index for the alternatives in a choice set, v_{ij} is a nonstochastic utility function that relates those observed factors to the utility, and ϵ_{ij} is the error component that captures the unobserved characteristics of the utility. In discrete choice models, the observed part of the utility function is assumed to be linear in the parameters,

$$v_{ij} = \mathbf{x}_{ij}'\boldsymbol{\beta}$$

where $\mathbf{x}_{ij}$ is a p-dimensional design vector of observed attribute levels that relate to alternative j and $\boldsymbol{\beta}$ is the corresponding vector of fixed regression coefficients that indicate the utilities or part-worths of the attribute levels.

Decision makers choose the alternative that gives them the greatest utility. Let $\mathbf{y}_i$ be the multinomial response vector for the ith individual. The value y_{ij} takes 1 if the jth component of $\mathbf{u}_i = (u_{i1}, \ldots, u_{iJ})$ is the largest, and 0 otherwise:

$$\begin{aligned} u_{ij} &= \mathbf{x}_{ij}'\boldsymbol{\beta} + \epsilon_{ij} \\ y_{ij} &= \begin{cases} 1 & \text{if } u_{ij} \geq \max(\mathbf{u}_i) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

The probability that the individual i chooses alternative j is

$$\begin{aligned} P(y_{ij} = 1) &= \Pr\left(u_{ij} > u_{ik} \text{ for all } k \neq j\right) \\ &= \Pr\left(v_{ij} + \epsilon_{ij} > v_{ik} + \epsilon_{ik} \text{ for all } k \neq j\right) \\ &= \Pr\left(\epsilon_{ik} - \epsilon_{ij} < v_{ij} - v_{ik} \text{ for all } k \neq j\right) \\ &= \int_{\epsilon} \mathrm{I}(\epsilon_{ik} - \epsilon_{ij} < v_{ij} - v_{ik} \text{ for all } k \neq j) f(\boldsymbol{\epsilon}_i) d\boldsymbol{\epsilon}_i \end{aligned}$$

where I(.) is the indicator function and $f(\boldsymbol{\epsilon}_i)$ denotes the joint density of the error vector $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \ldots, \epsilon_{iJ})$. Different specifications about the density result in different types of choice models, as detailed in the next section. Logit and nested logit models have a closed form for this integral, whereas a probit model does not.

Types of Choice Models

Logit

In the multinomial logit (MNL) model, each ϵ_{ij} $(j = 1, \ldots, J)$ is independently and identically distributed (iid) with the Type I extreme-value distribution, $\exp(-\exp(-\epsilon_{ij}))$, which is also known as the Gumbel distribution. The essential part of the model is that the unobserved parts of utility are uncorrelated among all alternatives, in addition to having the same variance. This assumption provides a convenient logit form for the choice probability (McFadden 1974):

$$P(y_{ij} = 1) = \frac{\exp(\mathbf{x}_{ij}'\boldsymbol{\beta})}{\sum_{k=1}^{J} \exp(\mathbf{x}_{ik}'\boldsymbol{\beta})}, \quad i = 1, \ldots, N \text{ and } j = 1, \ldots, J$$

The MNL likelihood is formed by the product of the N independent multinomial distributions:

$$p(\mathbf{Y}|\boldsymbol{\beta}) = \prod_{i=1}^{N} \prod_{j=1}^{J} P(y_{ij} = 1)^{y_{ij}}$$

The MNL model is by far the easiest and most widely used discrete choice model. It is popular because the formula for the choice probability has a closed form and is readily interpretable. In his Nobel lecture, McFadden (2001) tells a history of the development of this path-breaking model.

Bayesian analysis for the MNL model requires the specification of a prior over the coefficient parameters $\boldsymbol{\beta}$ and computation of the posterior density. It is convenient to use a normal prior on $\boldsymbol{\beta}$, $\pi(\boldsymbol{\beta}) = \mathrm{N}(\bar{\boldsymbol{\beta}}, \boldsymbol{\Omega}_{\boldsymbol{\beta}})$. When the investigator has no strong prior beliefs about the location of the parameters, it is recommended that diffuse, but proper, priors be used.

The posterior density of the parameter $\boldsymbol{\beta}$ is

$$p(\boldsymbol{\beta}|\mathbf{Y}) \propto p(\mathbf{Y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta})$$

Because discrete choice logit models fall under the framework of a generalized linear model (GLM) with a logit link, you can use Metropolis-Hastings sampling with the Gamerman approach for the sampling procedure. This approach extends the usual iterative weighted least squares method to include a sampling step based on the Metropolis-Hastings algorithm for GLMs. For more information about the Gamerman approach, see the section "Gamerman Algorithm" on page 1078. You can also incorporate this methodology for a hierarchical structure that has random effects. For more information, see the section "Logit with Random Effects" on page 1072. The Gamerman approach is the default sampling method when a direct sampling that uses a closed-form posterior distribution cannot be attained. The random walk Metropolis is also available when you specify ALGORITHM=RWM in the PROC BCHOICE statement.

The iid assumption about the error terms of the utilities of the different alternatives imposes the independence from irrelevant alternatives (IIA) property, which implies proportional substitution across alternatives. Sometimes this property might seem restrictive or unrealistic, prompting the need for more flexible models to overcome it.

Nested Logit

You derive the nested logit model by allowing the error components to be identical but nonindependent. Instead of being independent Type I extreme-value errors, the errors are assumed to have a generalized extreme-value (GEV) distribution. This model generalizes the standard logit model to allow for particular patterns of correlation in unobserved utility (McFadden 1978).

In a nested logit model, all the alternatives in a choice set can be partitioned into nests in such a way that the following conditions are true:

- The probability ratio of any two alternatives that are in the same nest is independent of the existence of all other alternatives. Hence, the IIA assumption holds within each nest.
- The probability ratio of any two alternatives that are in different nests is not independent of the existence of other alternatives in the two nests. Hence, the IIA assumption does not hold between different nests.

A nested logit example from Train (2009) best explains how a nested logit model works. Suppose the alternatives for commuting to work are driving an auto alone, carpooling, taking the bus, and taking the train. If you remove any alternative, the probabilities of the other alternatives increase. But by what percentage would each probability increase? Suppose the changes in probabilities occur as shown in Table 27.5.

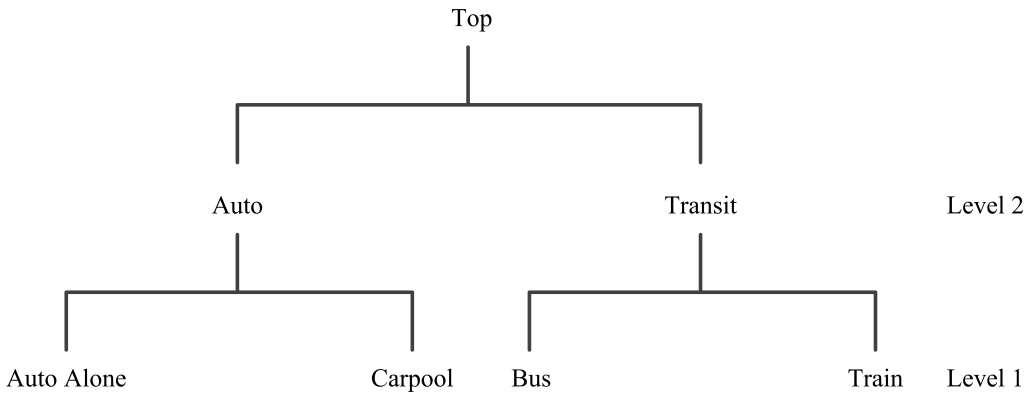
Table 27.5 Example of IIA Holding within Nests of Alternatives

		Probabilities with One Alternative Removed			
Alternative	**Original Prob.**	**Auto Alone**	**Carpool**	**Bus**	**Train**
Auto alone	0.40	–	0.45(+12.5%)	0.52(+30.0%)	0.48(+20.0%)
Carpool	0.10	0.20(+100%)	–	0.13(+30.0%)	0.12(+20.0%)
Bus	0.30	0.48(+60.0%)	0.33(+10.0%)	–	0.40(+33.0%)
Train	0.20	0.32(+60.0%)	0.22(+10.0%)	0.35(+70.0%)	–

In Table 27.5, the probabilities for bus and train increase by the same proportion whenever you remove one of the other alternatives. Therefore, IIA is valid between bus and train, so they can be placed in the same nest called "transit." Similarly, auto alone and carpool are placed in one nest called "auto." IIA does not hold between these two nests.

A convenient way to represent the model and substitution patterns is to draw a tree diagram, where each branch denotes a subset of alternatives within which IIA holds and where each leaf on each branch denotes an alternative. The tree diagram for the commuting-to-work example is shown in Figure 27.10.

Figure 27.10 Two-Level Decision Tree



Although decision trees in nested logit analyses, such as the one shown in Figure 27.10, are often interpreted as implying that the higher-level decisions are made first, followed by decisions at lower levels, no such temporal ordering is necessarily required. A good way to think about nested logit models is that they are appropriate when there are groups of alternatives that are similar to each other in unobserved ways. In other words, nested logit models are appropriate when there is correlation between the alternatives in each nest but no correlation between the alternatives in different nests. Currently, PROC BCHOICE considers only two-level nested logit models.

In mathematical notation, let the set of alternatives be partitioned into K nonoverlapping subsets (nests) that are denoted by $S_1, S_2, \ldots, S_K$. The nested logit model is derived by assuming that the vector of the

unobserved part of the utility, $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \ldots, \epsilon_{iJ})$, has a cumulative distribution

$$\exp(-\sum_{k=1}^{K}(\sum_{j \in S_k} \exp(-\epsilon_{ij}/\lambda_k))^{\lambda_k})$$

This is a type of generalized extreme value (GEV) distribution. For a standard logit model, each ϵ_{ij} is iid with an extreme-value distribution, whereas for a nested logit model, the marginal distribution of ϵ_{ij} is an extreme-value distribution. The ϵ_{ij} are correlated within nests. If alternatives j and m belong to the same nest, then ϵ_{ij} is correlated with ϵ_{im}. But if any two alternatives are in different nests, their unobserved part of utility is still independent. The parameter λ_k measures the degree of independence among alternatives in nest k. The higher the value of λ_k is, the less correlation there is. However, the correlation is actually more complicated than the parameter λ_k. The equation $\lambda_k = 1$ represents no correlation in nest k. If $\lambda_k = 1$ for all nests, the nested logit model reduces to the standard logit model.

The choice probability for alternative $j \in S_k$ has a closed form:

$$P(y_{ij} = 1) = \frac{\exp(\mathbf{x}'_{ij}\boldsymbol{\beta}/\lambda_k)(\sum_{m \in S_k} \exp(\mathbf{x}'_{im}\boldsymbol{\beta}/\lambda_k))^{\lambda_k - 1}}{\sum_{l=1}^{K}(\sum_{m \in S_l} \exp(\mathbf{x}'_{im}\boldsymbol{\beta}/\lambda_l))^{\lambda_l}}$$

From this form, the nested logit likelihood is derived as the product of the N multinomial distributions:

$$p(\mathbf{Y}|\boldsymbol{\beta}, \boldsymbol{\lambda}) = \prod_{i=1}^{N}\prod_{j=1}^{J} P(y_{ij} = 1)^{y_{ij}}$$

A prior for $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$ is needed in addition to a prior for the coefficient parameters $\boldsymbol{\beta}$. The λ_k in nest k is often called the log-sum coefficient. The value of λ_k must be positive for the model to be consistent with utility-maximizing behavior. If $\lambda_k \in [0, 1]$ for all k, the model is consistent with utility maximization for all possible values of the explanatory covariates, whereas the model is consistent only for some range of the covariates but not for all values if $\lambda_k > 1$. Kling and Herriges (1995) and Herriges and Kling (1996) suggest tests of consistency of nested logit with utility maximization when $\lambda_k > 1$. Train, McFadden, and Ben-Akiva (1987) show an example of models for which $\lambda_k > 1$. A negative value of λ_k should be avoided because the model will be inconsistent and imply that consumers are choosing the alternative to minimize their utilities.

For the nested logit model, noninformative priors are not ideal for $\boldsymbol{\lambda}$. Flat priors on different versions of the parameter space can yield different posterior distributions.

Lahiri and Gao (2002) suggest the following semi-flat priors by using the parameter ϕ for each λ:

$$\pi(\lambda) = \begin{cases} 0 & \text{if } \lambda \leq 0 \\ \phi & \text{if } 0 < \lambda < 1 \\ \phi \exp[\frac{\phi}{1-\phi}(1-\lambda)] & \text{if } \lambda \geq 1 \end{cases}$$

PROC BCHOICE uses this prior with a default value of 0.8 for ϕ. You can specify other values for ϕ. Sims' priors are also attractive with various s values as follows:

$$\pi(\lambda) = \begin{cases} 0 & \text{if } \lambda \leq 0 \\ s\lambda^{s-1}\exp(-\lambda^s) & \text{if } \lambda > 0 \end{cases}$$

In addition, a gamma or beta distribution is a good choice for the prior of $\boldsymbol{\lambda}$. However, PROC BCHOICE supports only semi-flat priors in this release.

When the priors for the parameters $(\boldsymbol{\beta}, \boldsymbol{\lambda})$ are specified, the posterior density is as follows:

$$p(\boldsymbol{\beta}, \boldsymbol{\lambda}|\mathbf{Y}) \propto p(\mathbf{Y}|\boldsymbol{\beta}, \boldsymbol{\lambda})\pi(\boldsymbol{\beta})\pi(\boldsymbol{\lambda})$$

Another generalization of the conditional logit model, the heteroscedastic extreme-value (HEV) model, is obtained by allowing independent but nonidentical errors that are distributed with a Type I extreme-value distribution (Bhat 1995). The HEV model permits different variances on the error components of utility across the alternatives. Currently, PROC BCHOICE does not consider this type of model.

Probit

The multinomial probit (MNP) model is derived when the errors, $\boldsymbol{\epsilon}_i' = (\epsilon_{i1}, \epsilon_{i2}, \ldots, \epsilon_{iJ})$, have a multivariate normal (MVN) distribution with a mean vector of $\mathbf{0}$ and a covariance matrix $\boldsymbol{\Sigma}$.

$$\begin{aligned} u_{ij} &= \mathbf{x}_{ij}'\boldsymbol{\beta} + \epsilon_{ij}, \;\; \boldsymbol{\epsilon}_i \sim \mathrm{N}(0, \boldsymbol{\Sigma}), \;\; i = 1, \ldots, N \;\text{ and }\; j = 1, \ldots, J \\ y_{ij} &= \begin{cases} 1 & \text{if } u_{ij} \geq \max(\mathbf{u}_i) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

where $\boldsymbol{\Sigma}$ is the J × J covariance matrix of the error vector. For a full covariance matrix, any pattern of correlation and heteroscedasticity can be accommodated. Thus, this model fits a very general error structure.

However, the choice probability, from which the likelihood is derived, does not have a closed form. McCulloch and Rossi (1994) propose an algorithm that is a multivariate version of the probit regression algorithm of Albert and Chib (1993) and construct a Gibbs sampler that is based on a Markov chain to draw directly from the exact posteriors of the MNP model. This approach avoids direct evaluation of the likelihood. Hence, it avoids problems that are associated with calculating choice probabilities that affect both the standard likelihood and the method of simulated moments approaches.

To avoid the parameter identification problems, a usual practice is to take differences against one particular alternative, because only differences in utility matter. Suppose you take differences against the last alternative in the choice set and you define $w_{ij} = u_{ij} - u_{iJ}$, $\tilde{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \mathbf{x}_{iJ}$, $\tilde{\epsilon}_{ij} = \epsilon_{ij} - \epsilon_{iJ}$, and $\tilde{\boldsymbol{\epsilon}}_i' = (\tilde{\epsilon}_{i1}, \tilde{\epsilon}_{i2}, \ldots, \tilde{\epsilon}_{iJ-1})$.

$$\begin{aligned} w_{ij} &= \tilde{\mathbf{x}}_{ij}'\boldsymbol{\beta} + \tilde{\epsilon}_{ij}, \;\; \tilde{\boldsymbol{\epsilon}}_i \sim \mathrm{N}(0, \tilde{\boldsymbol{\Sigma}}), \;\; i = 1, \ldots, N \;\text{ and }\; j = 1, \ldots, J-1 \\ y_{ij} &= \begin{cases} 1 & \text{if } w_{ij} \geq \max(0, \mathbf{w}_{i,-j}) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

where $\tilde{\boldsymbol{\Sigma}}$ is the $(J-1) \times (J-1)$ covariance matrix of the vector of error differences, $\tilde{\boldsymbol{\epsilon}}_i$, and $\mathbf{w}_{i,-j} = (w_{i1}, \ldots, w_{i(j-1)}, w_{i(j+1)}, \ldots, w_{i(J-1)})$. The dimension has been reduced to $J-1$. McCulloch and Rossi (1994) call w_{ij} the latent variable. The introduction of the latent variables in what is known as the data augmentation step makes the application of a Gibbs sampler feasible. The order in which the alternatives appear in a choice set is important, so you should arrange the alternatives to appear in the same order in all choice sets.

A normal prior is used for $\boldsymbol{\beta}$ and an inverse Wishart prior on $\tilde{\boldsymbol{\Sigma}}$:

$$\begin{aligned} \pi(\boldsymbol{\beta}) &= \mathrm{N}(\bar{\boldsymbol{\beta}}, \boldsymbol{\Omega}_{\boldsymbol{\beta}}) \\ \pi(\tilde{\boldsymbol{\Sigma}}) &= \text{inverse Wishart}(\nu, \mathbf{V}) \end{aligned}$$

Then sampling is carried out consecutively from the following three groups of conditional posterior distributions:

$$(1) \quad (w_{ij}|\mathbf{w}_{i,-j}, \boldsymbol{\beta}, \tilde{\boldsymbol{\Sigma}}, \mathbf{Y}), \quad i = 1, \ldots, N \text{ and } j = 1, \ldots, J-1$$
$$(2) \quad (\boldsymbol{\beta}|\mathbf{W}, \tilde{\boldsymbol{\Sigma}}, \mathbf{Y})$$
$$(3) \quad (\tilde{\boldsymbol{\Sigma}}|\mathbf{W}, \boldsymbol{\beta}, \mathbf{Y})$$

where $\mathbf{W}$ is obtained by stacking all $\mathbf{w}_i$. All three groups of conditional distributions have closed forms that are easily drawn from. Conditional (1) is truncated normals, (2) is a regular normal, and (3) is an inverse Wishart distribution. This sampling avoids the tuning stage that is required by most Metropolis-Hastings algorithms, and its analytically tractable complete data likelihood also makes it easy to embed probit models into more elaborate hierarchical structures, such as random-effects models. For more information, see McCulloch and Rossi (1994).

Originally, the multinomial probit model required burdensome computation compared to a family of multinomial choice models that are derived from the Gumbel distributed utility function, because the computation involves multidimensional integration (with dimension $J-1$) in the estimation process. In addition, the multinomial probit model requires more parameters than other multinomial choice models. However, it fully relaxes the restrictive IIA assumption that is imposed in logit models and allows any pattern of substitution among alternatives. Because of the breakthrough Bayesian methods that Albert and Chib (1993) and McCulloch and Rossi (1994) introduced, the probit model can be estimated without the need to compute the choice probabilities. This advantage has greatly increased the popularity of the probit model.

Random Effects

Choice models that have random effects (or random coefficients) provide solutions to create individual-level or group-specific utilities. They are also referred as "mixed models" or "hybrid models." One of the greatest challenges in marketing research is to account for the diversity of preferences and sensitivities in the marketplace. Heterogeneity in individual preferences is the reason for differentiated product programs and market segmentation. As individual preferences become more diverse, it becomes less appropriate to consider the analyses in an aggregated way. Individual utilities are useful because they make segmentation easy and provide a way to detect groups. Because people have different preferences, it can be misleading to roll the whole sample together into a single set of utilities.

For example, imagine studying the popularity of a new brand. Some participants in the study who are interviewed love the new brand, whereas others dislike it. If you simply aggregate the data and look at the average, the conclusion is that the sample is ambivalent toward the new brand. This would be the least helpful conclusion that could be drawn, because it does not fit for anyone.

Choice models that have random effects generalize the standard choice models to incorporate individual-level effects. Let the utility that individual i obtains from alternative j in choice situation t ($t = 1, \ldots, T$) be

$$u_{ijt} = \mathbf{x}'_{ijt}\boldsymbol{\beta} + \mathbf{z}'_{ijt}\boldsymbol{\gamma}_i + \epsilon_{ijt}$$
$$y_{ijt} = \begin{cases} 1 & \text{if } u_{ijt} \geq \max(u_{i1t}, u_{i2t}, \ldots, u_{iJt}) \\ 0 & \text{otherwise} \end{cases}$$

where y_{ijt} is the observed choice for individual i and alternative j in choice situation t; $\mathbf{x}_{ijt}$ is the fixed design vector for individual i and alternative j in choice situation t; $\boldsymbol{\beta}$ is the vector of fixed coefficients; $\mathbf{z}_{ijt}$ is the

random design vector for individual i and alternative j in choice situation t; and $\boldsymbol{\gamma}_i$ is the vector of random coefficients for individual i that correspond to $\mathbf{z}_{ijt}$. Sometimes the random coefficients might be at a level different from the individual level. For example, it is common to assume that there are random coefficients at the household level or group level of participants. For the convenience of notation, random effects are assumed to be at the individual level.

In the random-effects model, it is assumed that each $\boldsymbol{\gamma}_i$ is drawn from a superpopulation and this superpopulation is normal, $\boldsymbol{\gamma}_i \sim$ iid $\mathrm{N}(\mathbf{0}, \boldsymbol{\Omega}_{\boldsymbol{\gamma}})$. An additional stage is added to the model where a prior for $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$ is specified:

$$\begin{aligned} \pi(\boldsymbol{\gamma}_i) &= \mathrm{N}(\mathbf{0}, \boldsymbol{\Omega}_{\boldsymbol{\gamma}}) \\ \pi(\boldsymbol{\Omega}_{\boldsymbol{\gamma}}) &= \text{inverse Wishart}(\nu_0, \mathbf{V}_0) \end{aligned}$$

The covariance matrix $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$ characterizes the extent of heterogeneity among individuals. Large diagonal elements of $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$ indicate substantial heterogeneity in part-worths. Off-diagonal elements indicate patterns in the evaluation of attribute levels. For example, positive covariances specify pairs of attribute levels that tend to be evaluated similarly across respondents. Product offerings that consist of these attribute levels are more strongly preferred or disliked by certain individuals.

In this setup, the prior mean is 0 for the random effects, meaning that the random effects either are truly around 0 or have been centered by the fixed effects. For random effects whose mean is not around 0, you can follow the usual practice of specifying them in the fixed effects. For example, if one random effect is price and you do not think that the population mean of price is around 0, then you should add price as a fixed effect as follows:

```
proc bchoice data=randeffdata;
   class subj set;
   model y = price / choiceset=(subj set);
   random  price / subject=subj;
run;
```

Thus, you obtain the estimate for the population mean of the price effect through the fixed effect, and you obtain the deviation from the population mean for each individual through random effects.

Allenby and Rossi (1999) and Rossi, Allenby, and McCulloch (2005) propose a hierarchical Bayesian random-effects model that is set up in a different way. In their model, there are no fixed effects but only random effects. This model, which is referred to as the random-effects-only model in the rest of this chapter, is as follows:

$$\begin{aligned} u_{ijt} &= \mathbf{z}'_{ijt}\boldsymbol{\gamma}_i + \epsilon_{ijt} \\ y_{ijt} &= \begin{cases} 1 & \text{if } u_{ijt} \geq \max(u_{i1t}, u_{i2t}, \ldots, u_{iJt}) \\ 0 & \text{otherwise} \end{cases} \\ \pi(\boldsymbol{\gamma}_i) &= \mathrm{N}(\bar{\boldsymbol{\gamma}}, \boldsymbol{\Omega}_{\boldsymbol{\gamma}}) \\ \pi(\boldsymbol{\Omega}_{\boldsymbol{\gamma}}) &= \text{inverse Wishart}(\nu_0, \mathbf{V}_0) \end{aligned}$$

where $\bar{\boldsymbol{\gamma}}$ is a mean vector of regression coefficients, which models the central location of distribution of the random coefficients; $\bar{\boldsymbol{\gamma}}$ represents the average part-worths across the respondents in the data. If you want to use this setup, specify the REMEAN option in the RANDOM statement to request estimation on $\bar{\boldsymbol{\gamma}}$ and do not specify any fixed effects in the MODEL statement.

Rossi, McCulloch, and Allenby (1996) and Rossi, Allenby, and McCulloch (2005) add another layer of flexibility to the random-effects-only model by allowing heterogeneity that is driven by observable (demographic) characteristics of the individuals. They model the prior mean of the random coefficients as a function of the individual's demographic variables (such as age and gender),

$$\begin{aligned} \pi(\boldsymbol{\gamma}_i) &= \mathrm{N}(\boldsymbol{\Gamma} d_i, \boldsymbol{\Omega}_{\boldsymbol{\gamma}}) \\ \pi(\boldsymbol{\Omega}_{\boldsymbol{\gamma}}) &= \text{inverse Wishart}(\nu_0, \mathbf{V}_0) \end{aligned}$$

where d_i is a vector that consists of an intercept and some observable demographic variables. $\boldsymbol{\Gamma}$ is a matrix of regression coefficients, which affects the location of distribution of the random coefficients. $\boldsymbol{\Gamma}$ should be useful for identifying respondents who have part-worths that are different from those in the rest of the sample if some individual-level characteristics are included in d_i. This specification allows the preferences or intercepts to vary by both demographic variables and the slopes. If d_i consists of only an intercept, this model reduces to the previous one. For more information about how to sample $\boldsymbol{\Gamma}$, see Rossi, McCulloch, and Allenby (1996), Rossi, Allenby, and McCulloch (2005) (Section 2.12 in Chapter 2), and Rossi (2012).

Logit with Random Effects

The logit model with random effects consists of the fixed-coefficients parameters $\boldsymbol{\beta}$, the random-coefficients parameters $\boldsymbol{\gamma}_i$, and the covariance parameters for the random coefficients $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$. You can use the Metropolis-Hastings sampling with Gamerman approach to draw samples through the following three conditional posterior distributions:

(1) $(\boldsymbol{\beta}|\boldsymbol{\gamma}_i, \mathbf{Y})$
(2) $(\boldsymbol{\gamma}_i|\boldsymbol{\beta}, \boldsymbol{\Omega}_{\boldsymbol{\gamma}}, \mathbf{Y}) \quad i = 1, \ldots, N$
(3) $(\boldsymbol{\Omega}_{\boldsymbol{\gamma}}|\boldsymbol{\gamma}_i, \mathbf{Y})$

All chains are initialized with random effects that are set to 0 and a covariance matrix that is set to an identity matrix. Updating is done first for the fixed effects, $\boldsymbol{\beta}$, as a block to position the chain in the correct region of the parameter space. Then the random effects are updated, and finally the covariance of the random effects is updated. For more information, see Gamerman (1997) and the section "Gamerman Algorithm" on page 1078.

The hierarchical Bayesian random-effects-only model as proposed in Allenby and Rossi (1999) and Rossi, Allenby, and McCulloch (2005) contains the random-coefficients parameters $\boldsymbol{\gamma}_i$, the population mean of the random-coefficients parameters $\bar{\boldsymbol{\gamma}}$, and the covariance parameters for the random coefficients $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$. The sampling can be carried out by the following conditional posteriors:

(1) $(\boldsymbol{\gamma}_i|\bar{\boldsymbol{\gamma}}, \boldsymbol{\Omega}_{\boldsymbol{\gamma}}, \mathbf{Y}) \quad i = 1, \ldots, N$
(2) $(\bar{\boldsymbol{\gamma}}|\boldsymbol{\gamma}_i, \boldsymbol{\Omega}_{\boldsymbol{\gamma}})$
(3) $(\boldsymbol{\Omega}_{\boldsymbol{\gamma}}|\boldsymbol{\gamma}_i, \bar{\boldsymbol{\gamma}})$

The second and third conditional posteriors are easy to draw because they have direct sampling distributions: the second has a normal distribution with a mean of $\sum_{i=1}^{N} \boldsymbol{\gamma}_i / N$ and a covariance of $\boldsymbol{\Omega}_{\boldsymbol{\gamma}} / N$; the third is an inverse Wishart$(\nu_0 + N, \mathbf{V}_0 + S)$, where $S = \sum_{i=1}^{N} (\boldsymbol{\gamma}_i - \bar{\boldsymbol{\gamma}})(\boldsymbol{\gamma}_i - \bar{\boldsymbol{\gamma}})' / N$. There is no closed form for the first conditional posterior. The Metropolis-Hastings sampling with Gamerman approach is the default sampling algorithm for it.

You can also use random walk Metropolis sampling when direct sampling is not an option, such as for the fixed-coefficients parameters $\boldsymbol{\beta}$ and random-coefficients parameters $\boldsymbol{\gamma}_i$. You can specify ALGORITHM=RWM to

choose random walk Metropolis sampling. For the random-coefficients parameters $\boldsymbol{\gamma}_i$, Rossi, McCulloch, and Allenby (1996) and Rossi, Allenby, and McCulloch (2005) suggest random walk Metropolis sampling in which increments have covariance $s^2\boldsymbol{\Omega}_{\boldsymbol{\gamma}}^t$, where s is a scaling constant whose value is usually set at $\frac{2.93}{\sqrt{\dim(\boldsymbol{\gamma}_i)}}$ and $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}^t$ is the current draw of $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$.

The sampling for the random-effects-only setup is often faster, because the first conditional posterior for each i involves only the data for individual i and because the second and third conditional posteriors depend not on the data directly but only on the draws of $\boldsymbol{\gamma}_i$.

Probit with Random Effects

The probit model with random effects has the following parameters: the fixed-coefficients parameters $\boldsymbol{\beta}$, the covariance parameters for the error differences $\tilde{\boldsymbol{\Sigma}}$, the random-coefficients parameters $\boldsymbol{\gamma}_i$, and the covariance parameters for the random coefficients $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$. It has extra parameters $(\boldsymbol{\gamma}_i, \boldsymbol{\Omega}_{\boldsymbol{\gamma}})$ in addition to $(\boldsymbol{\beta}, \tilde{\boldsymbol{\Sigma}})$ in a fixed-effects-only model. You can conveniently adapt the Gibbs sampler proposed in McCulloch and Rossi (1994) to handle the model by appending the new parameters to the set of parameters that would be drawn for a probit model without random coefficients. The sampling can be carried out by the following conditional posteriors:

(1) $(w_{ij}|\mathbf{w}_{i,-j}, \boldsymbol{\beta}, \tilde{\boldsymbol{\Sigma}}, \boldsymbol{\gamma}_i, \mathbf{Y}) \quad i = 1, \ldots, N \text{ and } j = 1, \ldots, J-1$

(2) $(\boldsymbol{\beta}|\mathbf{W}, \tilde{\boldsymbol{\Sigma}}, \boldsymbol{\gamma}_i, \mathbf{Y})$

(3) $(\boldsymbol{\gamma}_i|\mathbf{W}, \boldsymbol{\beta}, \tilde{\boldsymbol{\Sigma}}, \boldsymbol{\Omega}_{\boldsymbol{\gamma}}, \mathbf{Y}) \quad i = 1, \ldots, N$

(4) $(\tilde{\boldsymbol{\Sigma}}|\mathbf{W}, \boldsymbol{\beta}, \mathbf{Y})$

(5) $(\boldsymbol{\Omega}_{\boldsymbol{\gamma}}|\mathbf{W}, \boldsymbol{\beta}, \tilde{\boldsymbol{\Sigma}}, \boldsymbol{\gamma}_i, \mathbf{Y})$

All the groups of conditional distributions have closed forms that are easily drawn from. Conditional (1) is truncated normal, (2) and (3) are normal, and (4) and (5) are inverse Wishart distribution. For more information, see McCulloch and Rossi (1994).

Other Types of Choice Response

So far, the focus has been on one type of choice response, where the response takes binary integer values: 1 for a chosen alternative and 0 for an unchosen alternative. Each choice set should have at least two alternatives and only one chosen alternative. This is the default choice-type.

However, there are other types of choice response that PROC BCHOICE supports.

MaxDiff

Louviere (1991) and Finn and Louviere (1992) developed a technique in which a person was shown a set or subset of the possible items and asked to indicate the least preferred item in a choice set, in addition to the traditional most preferred item. This approach is now called MaxDiff (maximum difference) or best-worst scaling (BWS).

A respondent can pick one best and one worst item in each choice set. The choice response variable takes one of the three integer values: 1 for the best alternative, –1 for the worst alternative, and 0 for the ones in the middle. MaxDiff questionnaires are relatively easy and straightforward. People are often much

better at judging items at extremes than discriminating among items of medium importance or preference (Louviere 1991). MaxDiff is free from scale bias because the responses involve choosing items rather than expressing strength of preference. MaxDiff forces trade-offs of items and provides better discrimination among alternatives and between groups of the alternatives.

Following random utility theory, the MaxDiff experiments treat the choices of best alternatives as utility-maximizing behavior and the choices of worst alternatives as utility-minimizing behavior. Each choice set in a MaxDiff data set is coded as two separate sets: one for the best choice and one for the worst choice. In the first set, the only change is to replace the response value –1 with 0 for the worst choice. In the second set, all elements in the attribute design matrix are negated (multiplied by –1), the response value is changed from 1 to 0 for the best choice, and the response value is changed from –1 to 1 for the worst choice. The observed part of the utility function for the set of the best choice remains the same,

$$v_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta}$$

whereas the observed part of the utility function for the set of the worst choice has changed the sign,

$$v_{ij} = -\mathbf{x}'_{ij}\boldsymbol{\beta}$$

Then, you concatenate these two sets and estimate parameters by using a single model.

For MaxDiff data (where CHOICETYPE=MAXDIFF is specified), PROC BCHOICE expands each choice set to two separate sets: one for the best choice and one for the worst choice. You can do the data expansion by yourself and then treat the data as regular binary type of choice data without specifying CHOICETYPE=MAXDIFF. We recommend the former, although these two approaches are equivalent. If there is non-estimable parameters due to singularity when a classification covariate is specified, PROC BCHOICE is able to detect it; whereas if you expand the data as described above before calling PROC BCHOICE, those parameters (columns), which were singular before data expansion, are not singular any more (but they should be deleted).

Allocation

Binary choice data closely mimic actual purchase situations, but researchers sometimes want more information than just determining the best choice: for example, they might ask respondents to answer more fully by allocating a percentage to each alternative in a choice set. Alternatively, the respondents might be asked to think about the next 10 buying occasions and to state how many times they will pick each alternative in total. This type of choice experiment is called an allocation choice model.

In an allocation choice model, the choice takes a percentage of preference for each alternative, and the sum of percentages in each choice set is 100%. An allocation choice model can provide more information than choices alone, but it has its own disadvantages. One shortcoming is that it takes respondents longer to answer each question. Another is that respondents' mental processes of assigning percentages are not clear.

When the choice response is of the allocation type, you can specify the weight suboptions to indicate how many choice sets the allocated percentage should apply to. For example, if the respondent were asked to imagine the next 20 purchase occasions and to allocate a percentage of those 20 purchases for each alternative, you would specify WEIGHT=20. By default, WEIGHT=10.

Identification and Specification

Two aspects of identification in choice models play an important role in model specification: location shift and scale shift. For more information, see Chapter 2 in Train (2009).

Location Shift

Decision makers choose the alternative that gives them the greatest utility. If a constant is added to the utilities of all alternatives, the location of the utilities will change, but the decision maker will choose the same alternative.

The probability that the individual i chooses alternative j is

$$\begin{aligned} P(y_{ij}=1) &= \Pr\left(u_{ij} > u_{ik} \text{ for all } k \neq j\right) \\ &= \Pr\left(u_{ij} - u_{ik} > 0 \text{ for all } k \neq j\right) \end{aligned}$$

Only the difference in utilities matters, not the absolute values (Train 2009). This explains why there is no overall intercept term in a choice model.

Alternative-Specific Main Effects

People often want to observe the main effect that is related to each of the alternatives. For example,

$$u_{ij} = \alpha_j + \mathbf{x}_{ij}'\boldsymbol{\beta} + \epsilon_{ij} \text{ for all } j$$

where α_j is the main effect that is related to alternative j, $\mathbf{x}_{ij}$ is the design variable vector that is specific to alternative j, and $\boldsymbol{\beta}$ are the corresponding coefficients of $\mathbf{x}_{ij}$. The main effect α_j implies the average effect for alternative j on utility of all factors that are not included in the model.

Consider a person who can take either a car or a bus to work. The form of utilities for taking a car or bus can be specified as follows:

$$\begin{aligned} u_{\text{car}} &= \alpha^1_{\text{car}} + \mathbf{x}_{\text{car}}'\boldsymbol{\beta} + \epsilon_{\text{car}} \\ u_{\text{bus}} &= \alpha^1_{\text{bus}} + \mathbf{x}_{\text{bus}}'\boldsymbol{\beta} + \epsilon_{\text{bus}} \end{aligned}$$

The preceding form is equivalent to the form

$$\begin{aligned} u_{\text{car}} &= \alpha^2_{\text{car}} + \mathbf{x}_{\text{car}}'\boldsymbol{\beta} + \epsilon_{\text{car}} \\ u_{\text{bus}} &= \alpha^2_{\text{bus}} + \mathbf{x}_{\text{bus}}'\boldsymbol{\beta} + \epsilon_{\text{bus}} \end{aligned}$$

as long as $\alpha^1_{\text{car}} - \alpha^1_{\text{bus}} = \alpha^2_{\text{car}} - \alpha^2_{\text{bus}}$. There are infinitely many values that will result in the same choice probabilities. The two main effects are not estimable.

To avoid this problem, you must normalize one of the main effects to 0. If you set α_{car} to 0, then α_{bus} is interpreted as the average effect on the utility of bus relative to car. PROC BCHOICE automatically normalizes for you by setting one of the main effects to 0 if you specify the alternative specific main effects in the CLASS statement. If you want to designate which alternative to set to 0, use the REF= option in the CLASS statement.

Individual-Specific Effects

Attributes of the alternatives vary, so they are also called alternative-specific effects. However, demographic variables, such as age, gender, and race, do not change among alternatives for an individual. These are individual-specific effects. They can enter the model only if they are specified in ways that create differences in utilities among alternatives.

Suppose you want to examine the effect of an individual's age on choosing car or bus:

$$\begin{aligned} u_{\text{car}} &= \mathbf{x}'_{\text{car}}\boldsymbol{\beta} + \theta_{\text{car}}\text{Age} + \epsilon_{\text{car}} \\ u_{\text{bus}} &= \alpha_{\text{bus}} + \mathbf{x}'_{\text{bus}}\boldsymbol{\beta} + \theta_{\text{bus}}\text{Age} + \epsilon_{\text{bus}} \end{aligned}$$

Because only differences in utility matter, you cannot estimate both θ_{car} and θ_{bus}. You need to set one of them to 0. You can do this by interacting Age with the alternative-specific main effects, one of which has been normalized to 0:

$$\begin{aligned} u_{\text{car}} &= \mathbf{x}'_{\text{car}}\boldsymbol{\beta} + \epsilon_{\text{car}} \\ u_{\text{bus}} &= \alpha_{\text{bus}} + \mathbf{x}'_{\text{bus}}\boldsymbol{\beta} + \theta_{\text{bus}}\text{Age} + \epsilon_{\text{bus}} \end{aligned}$$

where θ_{bus} is interpreted as the differential effect of age on the utility of bus versus car.

You need to make sure that individual-specific effects have interacted with some alternative-specific effects, so that there is no identification problem. PROC BCHOICE checks whether there is any variable that does not change among alternatives; it issues a warning message if any such variable is found.

Scale Shift

The scale of utility is also irrelevant for choice models. The alternative that has the largest utility is the largest, regardless of how the utilities are scaled. Multiplying the utilities by a constant does not change the choice probabilities.

Scale Normalization for Logit Models

Consider the utility function

$$u^1_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \epsilon^1_{ij} \quad \text{for all } j$$

where $\text{Var}(\epsilon^1_{ij}) = 1$. There is another utility function

$$u^2_{ij} = \mathbf{x}'_{ij}(\boldsymbol{\beta} * \sigma) + \epsilon^2_{ij} \quad \text{for all } j$$

where $\text{Var}(\epsilon^2_{ij}) = \sigma^2$. These two utility functions are equivalent. The coefficients $(\boldsymbol{\beta} * \sigma)$ reflect the effect of the observed variables relative to the standard deviation of the unobserved factors (Train 2009).

In both standard and nested logit models, the error variance is equal to $\pi^2/6$, which is around 1.6. So the coefficients are larger by a factor of $\sqrt{1.6}$ than the coefficients of a model that has an error variance of 1.

Scale Normalization for Probit Models

Parameters in logit models are automatically normalized because of the specific variance assumption on the error term. In probit models, the covariance matrix of the error is estimated together with the $\boldsymbol{\beta}$ coefficients; therefore, the normalization is not done automatically. The researcher has to be careful in interpreting the results.

For notational convenience, use a four-alternative model. The error vector is $\boldsymbol{\epsilon}' = (\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4)$, which is assumed to have a normal distribution with 0 mean and a covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ . & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ . & . & \sigma_{33} & \sigma_{34} \\ . & . & . & \sigma_{44} \end{pmatrix}$$

where the dots refer to the corresponding elements in the upper part of the symmetric matrix. There are J(J+1)/2 elements in the covariance matrix, but not all of them are estimable, because only the difference in the utilities matters.

A typical way of normalizing the covariance matrix is to reduce it by differencing the utilities with respect to the last alternative; some prefer the first alternative (Train 2009). You can choose which alternative to subtract by rearranging the data in the right order. Define the error differences as $(\tilde{\epsilon}_{14}, \tilde{\epsilon}_{24}, \tilde{\epsilon}_{34})$; the subscripts mean that the error difference is taken against the fourth one. The covariance matrix for the transformed new vector of error differences is of the form

$$\tilde{\boldsymbol{\Sigma}} = \begin{pmatrix} \tilde{\sigma}_{11} & \tilde{\sigma}_{12} & \tilde{\sigma}_{13} \\ . & \tilde{\sigma}_{22} & \tilde{\sigma}_{23} \\ . & . & \tilde{\sigma}_{33} \end{pmatrix}$$

where

$$\begin{aligned} \tilde{\sigma}_{11} &= \sigma_{11} + \sigma_{44} - 2\sigma_{14} \\ \tilde{\sigma}_{22} &= \sigma_{22} + \sigma_{44} - 2\sigma_{24} \\ \tilde{\sigma}_{33} &= \sigma_{33} + \sigma_{44} - 2\sigma_{34} \\ \tilde{\sigma}_{12} &= \sigma_{12} + \sigma_{44} - \sigma_{14} - \sigma_{24} \\ \tilde{\sigma}_{13} &= \sigma_{13} + \sigma_{44} - \sigma_{14} - \sigma_{34} \\ \tilde{\sigma}_{23} &= \sigma_{23} + \sigma_{44} - \sigma_{24} - \sigma_{34} \end{aligned}$$

Furthermore, the top left entry on the diagonal is set to 1:

$$\tilde{\tilde{\boldsymbol{\Sigma}}} = \begin{pmatrix} 1 & \tilde{\sigma}_{12}/\tilde{\sigma}_{11} & \tilde{\sigma}_{13}/\tilde{\sigma}_{11} \\ . & \tilde{\sigma}_{22}/\tilde{\sigma}_{11} & \tilde{\sigma}_{23}/\tilde{\sigma}_{11} \\ . & . & \tilde{\sigma}_{33}/\tilde{\sigma}_{11} \end{pmatrix}$$

PROC BCHOICE outputs estimates of $\tilde{\tilde{\boldsymbol{\Sigma}}}$.

In the multinomial logit (MNL) model, each ϵ_{ij} is independently and identically distributed (iid) with the Type I extreme-value distribution, and the covariance matrix is $\pi^2/6I$. The normalized covariance matrix of the vector of error differences would be

$$\begin{pmatrix} 1 & 0.5 & .5 \\ . & 1 & .5 \\ . & . & 1 \end{pmatrix}$$

This provides a method of model selection: you might want to fit a probit model first, check the normalized covariance matrix estimation to see how close it is to the preceding matrix, and then decide whether a simpler logit model would be appropriate.

Gamerman Algorithm

Discrete choice logit models fall in the framework of a generalized linear model (GLM) with a logit link. The Metropolis-Hastings sampling approach of Gamerman (1997) is well suited to this type of model.

In the GLM setting, the data are assumed to be independent with exponential family density

$$f(\mathbf{y}_i|\boldsymbol{\theta}_i) = \exp\left[(\mathbf{y}_i\boldsymbol{\theta}_i - b(\boldsymbol{\theta}_i))/\phi_i\right]c(\mathbf{y}_i, \phi_i)$$

The means that $\boldsymbol{\mu}_i = E(\mathbf{y}_i|\boldsymbol{\theta}_i)$ are related to the canonical parameters $\boldsymbol{\theta}_i$ via $\boldsymbol{\mu}_i = b'(\boldsymbol{\theta}_i)$ and to the regression coefficients via the link function

$$g(\boldsymbol{\mu}_i) = \eta_i = \mathbf{X}_i\boldsymbol{\beta}$$

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

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

and define the corresponding weights as

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

Suppose a normal prior is specified on $\boldsymbol{\beta}$, $\pi(\boldsymbol{\beta}) = \mathrm{N}(\bar{\boldsymbol{\beta}}, \boldsymbol{\Omega}_{\boldsymbol{\beta}})$. The posterior density is as follows:

$$\begin{aligned} p(\boldsymbol{\beta}|\mathbf{Y}) &\propto p(\mathbf{Y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta}) \\ &\propto \exp\{\sum_{i=1}^{N}\frac{\mathbf{y}_i\boldsymbol{\theta}_i - b(\boldsymbol{\theta}_i)}{\phi_i} - \frac{1}{2}(\boldsymbol{\beta}-\bar{\boldsymbol{\beta}})'\boldsymbol{\Omega}_{\boldsymbol{\beta}}^{-1}(\boldsymbol{\beta}-\bar{\boldsymbol{\beta}})\} \end{aligned}$$

Gamerman (1997) proposes that Metropolis-Hastings sampling be combined with iterative weighted least squares 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 $\boldsymbol{\beta}^*$ with probability

$$\alpha(\boldsymbol{\beta}_{(t-1)}, \boldsymbol{\beta}^*) = \min[1, \frac{p(\boldsymbol{\beta}^*|\mathbf{Y})q(\boldsymbol{\beta}^*, \boldsymbol{\beta}^{(t-1)})}{p(\boldsymbol{\beta}^{(t-1)}|\mathbf{Y})q(\boldsymbol{\beta}^{(t-1)}, \boldsymbol{\beta}^*)}]$$

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

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

You can extend this methodology to logit models that have random effects. If there are random effects, the link function is extended to

$$g(\boldsymbol{\mu}_i) = \eta_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\boldsymbol{\gamma}_i$$

where the random effects are assumed to have a normal distribution, $\pi(\boldsymbol{\gamma}_i) = \mathrm{N}(\mathbf{0}, \boldsymbol{\Omega}_{\boldsymbol{\gamma}})$, and $\pi(\boldsymbol{\Omega}_{\boldsymbol{\gamma}}) =$ inverse Wishart$(\nu_0, \mathbf{V}_0)$. The posterior density is

$$p(\boldsymbol{\beta}, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_N, \boldsymbol{\Omega}_{\boldsymbol{\gamma}}|\mathbf{Y}) \propto p(\mathbf{Y}|\boldsymbol{\beta}, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_N, \boldsymbol{\Omega}_{\boldsymbol{\gamma}})\pi(\boldsymbol{\beta}) \prod_{i=1}^{N} \pi(\boldsymbol{\gamma}_i)\pi(\boldsymbol{\Omega}_{\boldsymbol{\gamma}})$$

The parameters are divided into blocks, $\boldsymbol{\beta}, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_N$, and $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$. For the fixed-effects $\boldsymbol{\beta}$ block, the conditional posterior has the same form, but the link changes to include $\mathbf{Z}_i\boldsymbol{\gamma}_i, i = 1, \ldots, N$, which are taken as known constants (offsets) at each iteration. The only change that is needed is to replace the transformed response $\tilde{\mathbf{y}}_i(\boldsymbol{\beta}^{(t-1)})$ with $\tilde{\mathbf{y}}_i(\boldsymbol{\beta}^{(t-1)}) - \mathbf{Z}_i\boldsymbol{\gamma}_i$ in step 2 of the previous Gamerman procedure.

For the random-effects $\boldsymbol{\gamma}_i$ block, the same Metropolis-Hastings sampling with the least square proposal can apply. The conditional posterior is

$$p(\boldsymbol{\gamma}_i|\mathbf{Y}, \boldsymbol{\beta}, \boldsymbol{\Omega}_{\boldsymbol{\gamma}}) \propto \exp\{\frac{\mathbf{y}_i\boldsymbol{\theta}_i - b(\boldsymbol{\theta}_i)}{\phi_i} - \frac{1}{2}\boldsymbol{\gamma}_i'\boldsymbol{\Omega}_{\boldsymbol{\gamma}}^{-1}\boldsymbol{\gamma}_i\}$$

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

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

Finally, for the covariance matrix $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$ block, direct sampling from an inverse Wishart$(\nu_0 + N, \mathbf{V}_0 + S)$ is used, where $S = \sum_{i=1}^{N} \boldsymbol{\gamma}_i\boldsymbol{\gamma}_i'/N$.

The chain is initialized with random effects set to 0 and the covariance set to the identity matrix. Updating is done first for the fixed effects, $\boldsymbol{\beta}$, as a block to position the chain in the correct region of the parameter space. Then the random effects are updated, and finally the covariance of the random effects is updated. For more information about this algorithm, see Gamerman (1997).

Tuning the Random Walk Metropolis in Logit Models

When you use a random walk Metropolis for a logit model, you want to find a good proposal distribution for each sampling block (for a fixed-effects block or for each individual random-effects block). This process is called tuning. The tuning phase consists of a number of loops. The minimum number of loops is controlled by the MINTUNE= option, which has a default value of 2. The MAXTUNE= option controls the maximum number of tuning loops, and it has a default value of 6. The number of iterations in each loop is controlled by the NTU= option, which has a default value of 500. At the end of every loop, PROC BCHOICE examines the acceptance probability for each block. The acceptance probability is the percentage of proposals (one from each iteration) that have been accepted. If the probability falls within the acceptance tolerance range (see the section "Scale Tuning" on page 1080), the current configuration of the scale and covariance matrix is kept. Otherwise, the scale and covariance matrix are modified before the next tuning loop.

A good proposal distribution should resemble the actual posterior distribution of the parameters. Large-sample theory states that the posterior distribution of the parameters approaches a multivariate normal distribution (Gelman et al. 2004, Appendix B; Schervish 1995, Section 7.4). That is why a normal proposal distribution often works well in practice. The default proposal distribution in PROC BCHOICE is the normal distribution:

$$q_j(\theta_{\text{new}}|\theta^t) = \text{MVN}(\theta_{\text{new}}|\theta^t, c^2\boldsymbol{\Sigma})$$

Scale Tuning

The acceptance rate is closely related to the sampling efficiency of a Metropolis chain. For a random walk Metropolis, a high acceptance rate means that most new samples occur very close to the current data point. Their frequent acceptance means that the Markov chain is moving rather slowly and not exploring the parameter space fully. On the other hand, a low acceptance rate means that the proposed samples are often rejected; hence the chain is not moving much. An efficient Metropolis sampler has an acceptance rate that is neither too high nor too low. The scale c in the proposal distribution $q(\cdot|\cdot)$ effectively controls this acceptance probability. Roberts, Gelman, and Gilks (1997) showed that if both the target and proposal densities are normal, the optimal acceptance probability for the Markov chain should be around 0.45 in a single-dimensional problem, and asymptotically approaches 0.234 in higher dimensions.

The nature of stochastic simulations makes it impossible to fine-tune a set of variables such that the Metropolis chain has the exact desired acceptance rate. In addition, Roberts and Rosenthal (2001) empirically demonstrate that an acceptance rate between 0.15 and 0.5 is at least 80% efficient, so there is really no need to fine-tune the algorithms to reach an acceptance probability that is within small tolerance of the optimal values. PROC BCHOICE works with a probability range, which is $a \pm b$, where a and b are the values of the TARGACCEPT and ACCEPTTOL options, respectively, in the BCHOICE statement. The default value of TARGACCEPT is a function of the number of parameters in the model, as outlined in Roberts, Gelman, and Gilks (1997). By default, ACCEPTTOL=0.075. If the observed acceptance rate in a particular tuning loop is less than the lower bound of the range, the scale is reduced; if the observed acceptance rate is greater than the upper bound of the range, the scale is increased. During the tuning phase, a scale parameter in the normal distribution is adjusted as a function of the observed acceptance rate and the target acceptance rate. PROC BCHOICE uses the updating scheme

$$c_{\text{new}} = \frac{c_{\text{cur}} \cdot \Phi^{-1}(p_{\text{opt}}/2)}{\Phi^{-1}(p_{\text{cur}}/2)}$$

where c_{cur} is the current scale, p_{cur} is the current acceptance rate, and p_{opt} is the optimal acceptance probability.[1]

Covariance Tuning

To tune a covariance matrix, PROC BCHOICE takes a weighted average of the old proposal covariance matrix and the recently observed covariance matrix, based on a number of samples in the current loop that is specified in NTU= option. The TUNEWT=*w* option determines how much weight is put on the recently observed covariance matrix. The following formula is used to update the covariance matrix:

$$\text{COV}_{\text{new}} = w\,\text{COV}_{\text{cur}} + (1-w)\text{COV}_{\text{old}}$$

For fixed effects:

- By default, PROC BCHOICE uses the mode of the posterior density as the initial value of the fixed parameter and uses a numerical optimization routine (such as the quasi-Newton method) to find a starting covariance matrix. The optimization is performed on the joint posterior distribution, and the covariance matrix is a quadratic approximation at the posterior mode. In some cases, this is a better and more efficient way of initializing the covariance matrix. However, there are cases (such as when the number of parameters is large) where the optimization could fail to find a matrix that is positive definite. In that case, the tuning covariance matrix is reset to the identity matrix.
- If you specify INIT=PRIORMODE or INIT=(LIST=*numeric-list*) to assign some initial values for the fixed effects in the MODEL statement, no numerical optimization is carried out and the proposal covariance matrix is set to the identity matrix divided by the number of parameters in the fixed effects sampling block. It can take a number of tuning phases before the proposal distribution is tuned to its optimal stage, because the Markov chain needs to spend time learning about the posterior covariance structure. If the posterior variances of your parameters vary by more than a few orders of magnitude, if the variances of your parameters are much different from 1, or if the posterior correlations are high, then the proposal tuning algorithm might have difficulty with forming an acceptable proposal distribution.

For random effects, the initial values are set to 0 and the proposal covariance matrix is set to the identity matrix divided by the number of parameters in each individual-level random-effects sampling block.

Autocall Macros for Postprocessing

Although PROC BCHOICE 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 options in advance. 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 27.6.

[1] Roberts, Gelman, and Gilks (1997) and Roberts and Rosenthal (2001) demonstrate that the relationship between acceptance probability and scale in a random walk Metropolis is $p = 2\Phi\left(-\sqrt{I}c/2\right)$, where c is the scale, p is the acceptance rate, Φ is the CDF of a standard normal, and $I \equiv E_f[(f'(x)/f(x))^2]$, $f(x)$ is the density function of samples. This relationship determines the updating scheme, with I being replaced by the identity matrix to simplify calculation.

Table 27.6 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 that used in PROC BCHOICE. As a result, there might be numerical differences in some cases, especially when the sample size is small.

Table 27.7 lists options that are shared by all postprocessing autocall macros. For macro-specific options, see Table 27.8.

Table 27.7 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 the data set that contains posterior samples is called Post and the variables of interest are defined in the macro variable &PARMS. The following statements call the %ESS macro and calculate 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 27.8.

Table 27.8 Macro-Specific Options

Macro	Option	Description
%ESS	AUTOCORLAG=*numeric*	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=*numeric*	Specifies the earlier portion of the Markov chain used in the test. By default, FRAC1=0.1.
	FRAC2=*numeric*	Specifies the latter portion of the Markov chain used in the test. By default, FRAC2=0.5.
%HEIDEL	SALPHA=*numeric*	Specifies the α level for the stationarity test. By default, SALPHA=0.05.
	HALPHA=*numeric*	Specifies the α level for the halfwidth test. By default, HALPHA=0.05.
	EPS=*numeric*	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=*numeric*	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(numeric-list)*	Specifies which autocorrelation lags to calculate. The default values are 1, 5, 10, and 50.
%POSTINT	ALPHA=*value*	Specifies the α level ($0 < \alpha < 1$) for the interval estimates. By default, ALPHA=0.05.
%RAFTERY	Q=*numeric*	Specifies the order of the quantile of interest. By default, Q=0.025.
	R=*numeric*	Specifies the margin of error for measuring the accuracy of estimation of the quantile. By default, R=0.005.
	S=*numeric*	Specifies the probability of attaining the accuracy of the estimation of the quantile. By default, S=0.95.
	EPS=*numeric*	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 BCHOICE 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.BCHOICE.Graphics.TraceAutocorrDensity** template is not feasible. The easiest way to regenerate the same graph is to use the %TADPLOT autocall macro. The %TADPLOT macro requires you to specify an input data set (which is the output data set from a previous PROC BCHOICE 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 statements to regenerate the graphs:

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

Displayed Output

This section describes the displayed output from PROC BCHOICE. For a quick reference of all ODS table names, see the section "ODS Table Names" on page 1087. ODS tables are arranged in four groups, listed in the following sections: "Model- and Data-Related ODS Tables" on page 1084, "Sampling-Related ODS Tables" on page 1085, "ODS Tables Related to Posterior Statistics" on page 1085, and "ODS Tables Related to Convergence Diagnostics" on page 1086.

Model- and Data-Related ODS Tables

Model Information

The "Model Information" table (ODS table name ModelInfo) displays basic information about the model, such as the name of the input data set, response variable, model type, sampling algorithm, burn-in size, simulation size, thinning, and random number seed. The "Model Information" table is produced irrespective of the sampling technique.

The "Model Information" table also displays the random number seed that is used to initialize the random number generators. If you repeat the analysis and specify this seed value in the SEED= option in the PROC BCHOICE statement, you get an identical stream of random numbers. This table is displayed by default.

Class Level Information

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

Choice Sets Summary

The "Choice Sets Summary" table (ODS table name ChoiceSummary) is displayed by default and should be used to check the data entry. It shows the numbers of choice sets, the number of alternatives, and the number of chosen and unchosen alternatives. You should always check this summary table to ensure that the data are arrayed correctly. This table is displayed by default.

Number of Observations

The "NObs" table (ODS table name NOBS) shows the number of observations in the data set and the number of observations that are used in the analysis. By default, observations that have missing values are not used. This table is displayed by default.

Sampling-Related ODS Tables

Burn-In History

The "Burn-In History" table (ODS table name BurnInHistory) shows the scales and acceptance rates for each parameter block in the burn-in phase for logit models. The table is not displayed by default, but you can request it by specifying MCHISTORY=BRIEF or MCHISTORY=DETAILED in the PROC BCHOICE statement.

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, but you can request it by specifying the option INIT=PINIT in the MODEL statement.

Prior for Fixed Effects

The "Prior for Fixed Effects" table (ODS table name CoeffPrior) shows the prior distribution for the regression coefficients. The table is displayed by default for a model with fixed effects.

Sampling History

The "Sampling History" table (ODS table name SamplingHistory) shows the scales and acceptance rates for each parameter block in the main sampling phase for logit models. The table is not displayed by default, but you can request it by specifying MCHISTORY=BRIEF or MCHISTORY=DETAILED in the PROC BCHOICE statement.

Tuning History

The "Tuning History" table (ODS table name TuningHistory) shows the number of tuning phases used in establishing the proposal distribution. The table also displays the scales and acceptance rates for each parameter block at each of the tuning phases for logit models using random walk Metropolis algorithm. For more information about the self-adapting proposal tuning algorithm, see the section "Tuning the Random Walk Metropolis in Logit Models" on page 1080. The table is not displayed by default, but you can request it by specifying MCHISTORY=BRIEF or MCHISTORY=DETAILED in the PROC BCHOICE statement.

ODS Tables Related to Posterior Statistics

PROC BCHOICE 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 BCHOICE statement. For more information about the calculations, see the section "Summary Statistics" on page 150 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, but you can request it by specifying the option STATISTICS=SUM.

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, but you can request it by specifying the option STATISTICS=CORR.

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, but you can request it by specifying the option STATISTICS=COV.

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, but you can request it by specifying the option DIC. For more information about the calculations, see the section "Deviance Information Criterion (DIC)" on page 152 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. This table is not displayed by default, but you can request it by specifying the option STATISTICS=INT.

ODS Tables Related to Convergence Diagnostics

PROC BCHOICE provides convergence diagnostic tests that check for Markov chain convergence. The procedure produces a number of ODS tables that you can request and save individually. For information about calculation, see the section "Statistical Diagnostic Tests" on page 141 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 BCHOICE displays lag 1, 5, 10, and 50 estimates of the autocorrelations. You can request different autocorrelations by using the DIAGNOSTICS=AUTOCORR(LAGS=) option. 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 149 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 150 in Chapter 7, "Introduction to Bayesian Analysis Procedures." This table is not displayed by default, but you can request it by specifying the option DIAGNOSTICS=MCSE.

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 143 in Chapter 7, "Introduction to Bayesian Analysis Procedures." This table is not displayed by default, but you can request it by specifying DIAGNOSTICS=GEWEKE.

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 stationary test and a half-width test. For more information, see the section "Heidelberger and Welch Diagnostics" on page 145 in Chapter 7, "Introduction to Bayesian Analysis Procedures." This table is not displayed by default, but you can request it by specifying DIAGNOSTICS=HEIDEL.

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 146 in Chapter 7, "Introduction to Bayesian Analysis Procedures." This table is not displayed by default, but you can request it by specifying DIAGNOSTICS=RAFTERY.

ODS Table Names

PROC BCHOICE assigns a name to each table that it creates. You can use these names to refer to the tables when you use the output delivery system (ODS) to select tables and create output data sets. These names are listed in Table 27.9. For more information about ODS, see Chapter 21, "Statistical Graphics Using ODS."

Table 27.9 ODS Tables Produced in PROC BCHOICE

ODS Table Name	Description	Statement or Option
AutoCorr	Autocorrelation statistics for each parameter	DIAG=AUTOCORR
BurnInHistory	History of burn-in phase sampling	MCHISTORY=BRIEF | DETAILED
ChoiceSummary	Choice sets summary	Default
ClassLevels	Class level information	Default
CoeffPrior	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

Table 27.9 (continued)

ODS Table Name	Description	Statement or Option
ESS	Effective sample size for each parameter	Default
Geweke	Geweke diagnostics for each parameter	DIAG=GEWEKE
Heidelberger	Heidelberger-Welch diagnostics for each parameter	DIAG=HEIDEL
MCSE	Monte Carlo standard error for each parameter	STATS=MCSE
ModelInfo	Model information	Default
NObs	Number of observations	Default
ParametersInit	Parameter initial values	INIT=PINIT
PostSumInt	Brief posterior statistics for each parameter, including sample size, mean, standard deviation, and HPD intervals	Default
PostIntervals	Equal-tail and HPD intervals for each parameter	STATS=INT
PostSummaries	Basic posterior statistics for each parameter, including sample size, mean, standard deviation, and percentiles	STATS=SUM
Raftery	Raftery-Lewis diagnostics for each parameter	DIAG=RAFTERY
SamplingHistory	History of main phase sampling	MCHISTORY=BRIEF | DETAILED
TuningHistory	History of proposal distribution tuning	MCHISTORY=BRIEF | DETAILED

ODS Graphics

You can refer by name to every graph that is produced through ODS Graphics. The names of the graphs that PROC BCHOICE generates are listed in Table 27.10.

Table 27.10 Graphs Produced by PROC BCHOICE

ODS Graph Name	Plot Description	Statement and Option
ADPanel	Autocorrelation function and density panel	PLOTS=(AUTOCORR DENSITY)
AutocorrPanel	Autocorrelation function panel	PLOTS=AUTOCORR
AutocorrPlot	Autocorrelation function plot	PLOTS(UNPACK)=AUTOCORR

Table 27.10 *continued*

ODS Graph Name	Plot Description	Statement and Option
DensityPanel	Density panel	PLOTS=DENSITY
DensityPlot	Density plot	PLOTS(UNPACK)=DENSITY
TAPanel	Trace and autocorrelation function panel	PLOTS=(TRACE AUTOCORR)
TADPanel	Trace, density, and autocorrelation function panel	PLOTS=(TRACE AUTOCORR DENSITY)
TDPanel	Trace and density panel	PLOTS=(TRACE DENSITY)
TracePanel	Trace panel	PLOTS=TRACE
TracePlot	Trace plot	PLOTS(UNPACK)=TRACE

Examples: BCHOICE Procedure

Example 27.1: Alternative-Specific and Individual-Specific Effects

In many situations, a choice model that includes characteristics of both the alternatives and the individuals is needed for investigating consumer choice.

Consider an example of travel demand. People are asked to choose among travel by auto, plane, or public transit (bus or train). The following SAS statements create the data set Travel. The variables AutoTime, PlanTime, and TranTime represent the total travel time that is required to get to a destination by using auto, plane, or public transit, respectively. The variable Age represents the age of each individual who is surveyed, and the variable Chosen contains each individual's choice of travel mode.

```
data Travel;
   input AutoTime PlanTime TranTime Age Chosen $;
   AgeCtr=Age-34;
   datalines;
10.0 4.5 10.5 32 Plane
5.5 4.0 7.5 13 Auto
4.5 6.0 5.5 41 Transit
3.5 2.0 5.0 41 Transit
1.5 4.5 4.0 47 Auto
10.5 3.0 10.5 24 Plane
7.0 3.0 9.0 27 Auto
9.0 3.5 9.0 21 Plane
4.0 5.0 5.5 23 Auto
22.0 4.5 22.5 30 Plane
7.5 5.5 10.0 58 Plane
11.5 3.5 11.5 36 Transit
3.5 4.5 4.5 43 Auto
12.0 3.0 11.0 33 Plane
18.0 5.5 20.0 30 Plane
```

```
23.0 5.5 21.5 28 Plane
4.0 3.0 4.5 44 Plane
5.0 2.5 7.0 37 Transit
3.5 2.0 7.0 45 Auto
12.5 3.5 15.5 35 Plane
1.5 4.0 2.0 22 Auto
;
```

In this example, the AutoTime, PlanTime, and TranTime variables apply to the alternatives, whereas Age is a characteristic of the individuals. AgeCtr, a centered version of Age, is created by subtracting the sample's mean age from each individual's age. To study how the choice depends on both the travel time and age of the individuals, you need to incorporate both types of variables.

Before you invoke PROC BCHOICE to fit a choice logit model, you must arrange your data in such a way that there is one observation for each combination of individual and alternative. In this example, let Subject identify the individuals, let TravTime represent the travel time for each mode of transportation, and let Choice have the value 1 if the alternative is chosen and 0 otherwise. The following SAS statements rearrange the data set Travel into a new data set, Travel2, and display the first nine observations:

```
data Travel2(keep=Subject Mode TravTime Age AgeCtr Choice);
   array Times[3] AutoTime PlanTime TranTime;
   array Allmodes[3] $ _temporary_ ('Auto' 'Plane' 'Transit');
   set Travel;
   Subject = _n_;
   do i = 1 to 3;
      Mode = Allmodes[i];
      TravTime = Times[i];
      Choice = (Chosen eq Mode);
      output;
   end;
run;

proc print data=Travel2 (obs=20);
   by Subject;
   id Subject;
run;
```

The data for the first nine observations is shown in Output 27.1.1.

Output 27.1.1 Data for the First Nine Observations

Subject	Age	AgeCtr	Mode	TravTime	Choice
1	32	-2	Auto	10.0	0
	32	-2	Plane	4.5	1
	32	-2	Transit	10.5	0

Subject	Age	AgeCtr	Mode	TravTime	Choice
2	13	-21	Auto	5.5	1
	13	-21	Plane	4.0	0
	13	-21	Transit	7.5	0

Output 27.1.1 *continued*

Subject	Age	AgeCtr	Mode	TravTime	Choice
3	41	7	Auto	4.5	0
	41	7	Plane	6.0	0
	41	7	Transit	5.5	1

Notice that each subject in the data set Travel corresponds to a block of three observations in the data set Travel2, one for each travel alternative. The response variable Choice indicates the chosen alternative by the value 1 and the unchosen alternative by the value 0; exactly one alternative is chosen. The following SAS statements invoke PROC BCHOICE to fit the choice logit model:

```
proc bchoice data=Travel2 seed=124;
   class Mode Subject / param=ref order=data;
   model Choice = Mode TravTime / choiceset=(Subject);
run;
```

The "Choice Sets Summary" table shows that there are 21 choice sets and that each consists of three alternatives and one chosen alternative (each subject chooses one out of the three travel modes). It seems that the data are arrayed correctly.

Output 27.1.2 Choice Sets Summary

The BCHOICE Procedure

Choice Sets Summary

Pattern	Choice Sets	Total Alternatives	Chosen Alternatives	Not Chosen
1	21	3	1	2

Summary statistics are shown in Output 27.1.3.

Output 27.1.3 PROC BCHOICE Posterior Summary Statistics

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Mode Auto	5000	-0.1678	0.7440	-1.7017	1.2396
Mode Plane	5000	-1.8794	1.2683	-4.6055	0.3801
TravTime	5000	-0.5695	0.2047	-0.9943	-0.2328

When Transit is the reference mode (normalized to 0), the part-worth (posterior mean) of Auto, which is negative, might reflect that driving is more inconvenient than traveling by bus or train, and the negative part-worth of Plane might reflect that traveling by plane is more expensive than traveling by bus or train. However, both are only suggestive, because the 95% HPD intervals have 0 in them. The posterior mean of TravTime is negative, which makes sense because having to spend more time en route is often unfavorable.

To study the relationship between the choice of transportation and the age of people who make the choice, you need to create an interaction between AgeCtr and Mode. AgeCtr is not estimable by itself, because it is the same throughout a choice set for an individual. The following statements fit a model including the interaction between AgeCtr and Mode:

```
proc bchoice data=Travel2 seed=124;
   class Mode Subject / param=ref order=data;
   model Choice = Mode Mode*AgeCtr TravTime / choiceset=(Subject);
run;
```

Output 27.1.4 PROC BCHOICE Posterior Summary Statistics

The BCHOICE Procedure

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Mode Auto	5000	-0.2634	0.7883	-1.8072	1.1395
Mode Plane	5000	-2.8210	1.5370	-5.9228	-0.0686
AgeCtr*Mode Auto	5000	-0.0986	0.0678	-0.2182	0.0350
AgeCtr*Mode Plane	5000	0.0251	0.0775	-0.1268	0.1618
TravTime	5000	-0.7608	0.2564	-1.2943	-0.3473

The parameter estimate for Mode Auto reflects the part-worth of Auto for an individual of mean age (34 years old), whereas the parameter estimate for Mode Plane is the part-worth of Plane for an individual of mean age. There are two interaction effects: the first corresponds to the effect of a one-unit change in age on the probability of choosing Auto over Transit, and the second corresponds to the effect of a one-unit change in age on the probability of choosing Plane over Transit.

Example 27.2: Nested Logit Modeling

The standard logit model imposes the restriction of the independence from irrelevant alternatives (IIA) property, which implies proportional substitution across alternatives. When the IIA assumption does not hold, models with more flexibility are needed.

One of the most widely used models is called the nested logit. In a nested logit model, all the alternatives in a choice set can be partitioned into nests in such a way that the following conditions are true:

- The ratio of any two alternatives that are in the same nest is independent of the existence of all other alternatives. Hence, the IIA assumption holds within each nest.
- The ratio of any two alternatives that are in different nests is not independent of the existence of other alternatives in the two nests. Hence, the IIA assumption does not hold between different nests.

For more information about nested logit models, see the section "Nested Logit" on page 1066.

In the previous example of travel demand data, people are asked to choose among travel by auto, plane, or public transit. It seems that auto and public transit are more similar to each other than either of them are to plane, because the probability of choosing auto and public transit might rise by about the same proportion whenever the option of taking a plane is unavailable. A nested logit model that places auto and public transit in one nest and plane in another nest might seem more reasonable than the standard logit model.

You specify a nested logit model by using the TYPE=NLOGIT option, and you allocate the nests by using the NEST= option. In the following code, NEST=(1 2 1) implies that travel alternatives 1 (auto) and 3 (public transit) are in the first nest and travel alternative 2 (plane) is in the second nest. There are two nests in total.

The deviance information criterion (DIC) is used to select the model. The "Deviance Information Criterion" table (ODS table name DIC) contains the DIC of the model. The table is not displayed by default, but you can request it by specifying the DIC option in the PROC BCHOICE statement. For more information about the calculations, see the section "Deviance Information Criterion (DIC)" on page 152 in Chapter 7, "Introduction to Bayesian Analysis Procedures." The DIC option requests the calculation of DIC.

```
proc bchoice data=Travel2 seed=124 nmc=20000 nthin=2 dic;
   class Mode Subject / param=ref order=data;
   model Choice = Mode TravTime / choiceset=(Subject) type=nlogit nest=(1 2 1);
run;
```

Output 27.2.1 PROC BCHOICE Posterior Summary Statistics and DIC

The BCHOICE Procedure

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Mode Auto	10000	-0.2398	0.6574	-1.5033	1.1475
Mode Plane	10000	-2.0295	1.1816	-4.2725	0.3157
TravTime	10000	-0.5638	0.2042	-0.9638	-0.1872
Lambda 1	10000	0.8884	0.3540	0.2338	1.5335

Deviance Information Criterion

Dbar (Posterior Mean of Deviance)	32.946
Dmean (Deviance Evaluated at Posterior Mean)	30.311
pD (Effective Number of Parameters)	2.635
DIC (Smaller is Better)	35.581

There are two alternatives in the first nest and one alternative in the second nest. A nest that has only one alternative is degenerate; therefore its λ is not estimable. This explains why there is only one λ estimate in the output.

The following statements revisit the standard logit model (the default type) that is discussed in the previous example and request that the calculation of DIC be displayed:

```
proc bchoice data=Travel2 seed=124 nmc=20000 nthin=2 dic;
   class Mode Subject / param=ref order=data;
   model Choice = Mode TravTime / choiceset=(Subject);
run;
```

Output 27.2.2 PROC BCHOICE Posterior Summary Statistics and DIC

The BCHOICE Procedure

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Mode Auto	10000	-0.1999	0.7341	-1.6891	1.1612
Mode Plane	10000	-2.0470	1.3376	-4.6183	0.5479
TravTime	10000	-0.6000	0.2333	-1.0716	-0.1986

Output 27.2.2 *continued*

Deviance Information Criterion	
Dbar (Posterior Mean of Deviance)	33.356
Dmean (Deviance Evaluated at Posterior Mean)	30.550
pD (Effective Number of Parameters)	2.806
DIC (Smaller is Better)	36.161

The DIC values are very close. A smaller DIC value indicates a better fit to the data; hence, the nested logit model might be a little better.

Example 27.3: Probit Modeling

A more general model is the probit model, which is derived under the assumption of jointly normal random utility components, where the error vector has a multivariate normal distribution with a mean vector of $\mathbf{0}$ and a covariance matrix $\mathbf{\Sigma}$. For a full covariance matrix, PROC BCHOICE can accommodate any pattern of correlation and heteroscedasticity. Thus, this model fits a very general error structure. For more information about probit models, see the section "Probit" on page 1069.

In Train (2009), a project team collected the following data from commuters about four available travel modes for their trips to work: car alone, carpool, bus, and railway. The time and cost of travel for each mode were determined for each commuter, based on the location of the commuter's home and workplace.

```
data Commuter;
   input Subject Mode Choice Cost Time @@;
   datalines;
1 1 1 1.51 18.5 1 2 0 2.34 26.34 1 3 0 1.8 20.87 1 4 0 2.36 30.03 2 1 0
6.06 31.31 2 2 0 2.9 34.26 2 3 0 2.24 67.18 2 4 1 1.86 60.29 3 1 1 5.79
22.55 3 2 0 2.14 23.26 3 3 0 2.58 63.31 3 4 0 2.75 49.17 4 1 1 1.87
26.09 4 2 0 2.57 29.9 4 3 0 1.9 19.75 4 4 0 2.27 13.47 5 1 1 2.5 4.7 5 2
0 1.72 12.41 5 3 0 2.69 43.09 5 4 0 2.97 39.74 6 1 1 4.73 3.07 6 2 0
0.62 9.22 6 3 0 1.85 12.83 6 4 0 2.31 43.54 7 1 1 4.73 13.14 7 2 0 0.6
17.77 7 3 0 2.43 54.09 7 4 0 2 42.22 8 1 1 5.35 52.9 8 2 0 2.91 48.78 8
3 0 2.61 69.16 8 4 0 2.78 53.25 9 1 0 4.41 61.06 9 2 0 1.59 62.13 9 3 1

   ... more lines ...

450 1 1 4.59 29.44 450 2 0 2.89 33.73 450 3 0 1.9 66.12 450 4 0 1.79
39.84 451 1 1 3.24 16.35 451 2 0 1.21 18.98 451 3 0 1.75 23.39 451 4 0
2.02 43.3 452 1 0 6.93 65.42 452 2 0 1.17 60.48 452 3 1 2.46 52.4 452 4
0 2.61 48.37 453 1 0 6.53 59.57 453 2 1 1.41 55.14 453 3 0 2.21 67.82
453 4 0 1.86 73.45
;

proc print data=Commuter(obs=20);
run;
```

The variable Mode has the value 1 for car alone, 2 for carpool, 3 for bus, and 4 for railway. The variable Choice is the response variable that represents the decision among the four travel modes for each commuter. The order in which the alternatives appear in a choice set is important: the variable Mode should have the value 1 in the first alternative, 2 in the second alternative, then 3 in the third alternative, and finally 4 in the last alternative. The data for the first five commuters are shown in Output 27.3.1.

Output 27.3.1 Data for the First Five Commuters

Obs	Subject	Mode	Choice	Cost	Time
1	1	1	1	1.51	18.50
2	1	2	0	2.34	26.34
3	1	3	0	1.80	20.87
4	1	4	0	2.36	30.03
5	2	1	0	6.06	31.31
6	2	2	0	2.90	34.26
7	2	3	0	2.24	67.18
8	2	4	1	1.86	60.29
9	3	1	1	5.79	22.55
10	3	2	0	2.14	23.26
11	3	3	0	2.58	63.31
12	3	4	0	2.75	49.17
13	4	1	1	1.87	26.09
14	4	2	0	2.57	29.90
15	4	3	0	1.90	19.75
16	4	4	0	2.27	13.47
17	5	1	1	2.50	4.70
18	5	2	0	1.72	12.41
19	5	3	0	2.69	43.09
20	5	4	0	2.97	39.74

The following statements fit a probit model by specifying TYPE=PROBIT. The BCHOICE procedure's implementation of the Gibbs sampler for the probit model exhibits a higher autocorrelation than that for the logit model. High autocorrelation is created by introducing the latent variable via data augmentation because of the dependence between the latent variable and the regression parameters. You might want to control the thinning rate of the simulation. For example, THIN=10 keeps every 10th sample in the simulation and discards the rest.

```
proc bchoice data=Commuter outpost=Commupostsamp thin=10 nmc=100000 seed=123;
   class Mode(ref='1') Subject;
   model Choice = Cost Time Mode / choiceset=(Subject) type=probit;
run;
```

Output 27.3.2 shows the summary statistics for the part-worth ($\boldsymbol{\beta}$) of each of the attributes (Cost, Time, Mode 2, Mode 3, and Mode 4) and the covariance of the error difference vector ($\tilde{\boldsymbol{\Sigma}}$), which is displayed by parameters labeled "Sigma 1 1," "Sigma 2 1," and so on.

Output 27.3.2 Posterior Summary Statistics

The BCHOICE Procedure

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Cost	10000	-0.4643	0.0753	-0.6206	-0.3280
Time	10000	-0.0495	0.00569	-0.0610	-0.0386
Mode 2	10000	-3.4074	0.7458	-4.8666	-2.0469
Mode 3	10000	-2.0063	0.2904	-2.5885	-1.4614
Mode 4	10000	-1.6257	0.2123	-2.0489	-1.2241
Sigma 1 1	10000	1.0000	0	1.0000	1.0000
Sigma 2 1	10000	1.1620	0.5604	0.0427	2.2958
Sigma 2 2	10000	4.4616	2.2789	1.2285	8.7152
Sigma 3 1	10000	0.5645	0.2149	0.1426	0.9928
Sigma 3 2	10000	1.5985	0.9345	-0.0286	3.6894
Sigma 3 3	10000	1.3834	0.4737	0.5985	2.3179

It is well known that an identification problem exists in probit models, because location and scale transformations do not change the choices that are made. The solution to the location shift is differencing with respect to the last alternative in each choice set. (See the section "Probit" on page 1069.) After that, a scale shift problem remains, because the parameters $(c\boldsymbol{\beta}, c^2\tilde{\boldsymbol{\Sigma}})$ for any constant $c > 0$ are equivalent to $(\boldsymbol{\beta}, \tilde{\boldsymbol{\Sigma}})$. A solution to the scaling problem is to normalize the parameters with respect to one of the diagonal elements of the covariance of the error difference vector, $\tilde{\boldsymbol{\Sigma}}$. PROC BCHOICE reports $(\boldsymbol{\beta}/\sqrt{\sigma_{11}}, \tilde{\boldsymbol{\Sigma}}/\sigma_{11})$ at each draw, where σ_{11} is the first diagonal entry of $\tilde{\boldsymbol{\Sigma}}$. This explains why "Sigma 1 1" is always 1 in Output 27.3.2.

By the IIA property in a logit model, it is assumed that all alternatives are independent and have the same variance. Therefore, the normalized covariance matrix after differencing with respect to one of the alternatives is of the form

$$\tilde{\boldsymbol{\Sigma}} = \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{pmatrix}$$

Obviously, this matrix is quite different from the estimated normalized covariance matrix for this data set. Fitting a standard logit model would be inappropriate.

Example 27.4: A Random-Effects-Only Logit Model

This example revisits the trash can study that is described earlier in this chapter in the getting-started section "A Logit Model Example with Random Effects" on page 1037.

If you want to create a random-effects-only model using the random walk Metropolis sampling as suggested in Rossi, Allenby, and McCulloch (2005), you can add the ALG=RWM option to the PROC BCHOICE statement to use the random walk Metropolis sampling algorithm instead of the default Gamerman Metropolis sampling, add the REMEAN option to the RANDOM statement to request estimation of the nonzero mean of the random effects, and not specify any fixed effects in the MODEL statement as follows:

```
proc bchoice data=Trashcan outpost=Postsamp seed=123 nmc=50000 thin=2
             alg=rwm nthreads=4;
   class ID Task;
   model Choice = / choiceset=(ID Task);
   random Touchless Steel AutoBag Price80 / sub=ID remean
          monitor=(1 to 5) type=un;
run;
```

Summary statistics for the mean of the random coefficients ($\bar{\boldsymbol{\gamma}}$), the covariance of the random coefficients ($\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$), and the random coefficients ($\boldsymbol{\gamma}_i$) for the first five individuals are shown in Output 27.4.1.

Output 27.4.1 Posterior Summary Statistics

The BCHOICE Procedure

Posterior Summaries and Intervals

Parameter	Subject	N	Mean	Standard Deviation	95% HPD Interval	
REMean Touchless		25000	1.7281	0.2711	1.1983	2.2651
REMean Steel		25000	1.0663	0.2661	0.5577	1.5998
REMean AutoBag		25000	2.2014	0.3520	1.5394	2.9100
REMean Price80		25000	-4.7018	0.6545	-6.0553	-3.4883
RECov Touchless, Touchless		25000	3.1178	1.1462	1.3552	5.3414
RECov Steel, Touchless		25000	-0.4517	0.9922	-2.1999	1.2156
RECov Steel, Steel		25000	2.6809	1.0342	1.0335	4.6468
RECov AutoBag, Touchless		25000	-0.7233	0.9521	-2.6063	1.0850
RECov AutoBag, Steel		25000	0.0301	0.8003	-1.4271	1.5548
RECov AutoBag, AutoBag		25000	3.6804	1.5597	1.1236	6.7726
RECov Price80, Touchless		25000	-1.3032	1.2372	-3.8305	0.8279
RECov Price80, Steel		25000	-1.6533	1.2926	-4.4595	0.5892
RECov Price80, AutoBag		25000	-2.3293	1.7191	-5.7326	0.6730
RECov Price80, Price80		25000	7.8969	3.3306	2.1101	14.3416
Touchless	ID 1	25000	2.2223	1.0613	0.1295	4.2930
Steel	ID 1	25000	0.7433	1.0823	-1.3053	2.9046
AutoBag	ID 1	25000	3.7970	1.4406	1.1535	6.7240
Price80	ID 1	25000	-5.2465	2.3491	-10.0027	-0.9522
Touchless	ID 2	25000	-0.0612	0.8563	-1.6726	1.6508
Steel	ID 2	25000	-0.3962	0.8279	-2.1014	1.1849
AutoBag	ID 2	25000	2.7067	1.3353	0.1149	5.3223
Price80	ID 2	25000	0.0271	1.3024	-2.5173	2.6137
Touchless	ID 3	25000	1.1029	0.9627	-0.7106	3.1099
Steel	ID 3	25000	1.6602	0.9599	-0.1585	3.5981
AutoBag	ID 3	25000	0.2961	1.1676	-2.0028	2.5169
Price80	ID 3	25000	-0.9568	1.5615	-4.0940	2.0521
Touchless	ID 4	25000	1.1556	0.9722	-0.7835	3.0279
Steel	ID 4	25000	1.3922	1.0206	-0.5989	3.4310
AutoBag	ID 4	25000	3.2143	1.4620	0.4988	6.1835
Price80	ID 4	25000	-6.6490	2.5043	-11.7925	-2.0763
Touchless	ID 5	25000	0.4804	1.1831	-1.7839	2.8515
Steel	ID 5	25000	2.7119	1.2432	0.4436	5.2051
AutoBag	ID 5	25000	3.4736	1.6239	0.5679	6.8042
Price80	ID 5	25000	-5.1156	2.5674	-10.0957	-0.0400

The average part-worths ($\bar{\boldsymbol{\gamma}}$) for touchless opening, steel material, automatic trash bag replacement, and price, which are shown as "REMean Touchless," "REMean Steel," "REMean AutoBag," and "REMean Price80," are very similar to the estimates of the fixed effects in the previous model as shown in Figure 27.9, indicating the equivalence of these two setups. The covariance estimates of the random coefficients ($\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$) are also very similar.

The next set of parameters show the estimates for the random effects for the first five respondents (see Output 27.4.1). However, these estimates are no longer the deviation from the overall means but are their own effects. The part-worth for touchless opening for the first respondent (ID 1) is 2.2, which is close to the part-worth that is estimated in the model as shown in Figure 27.9.

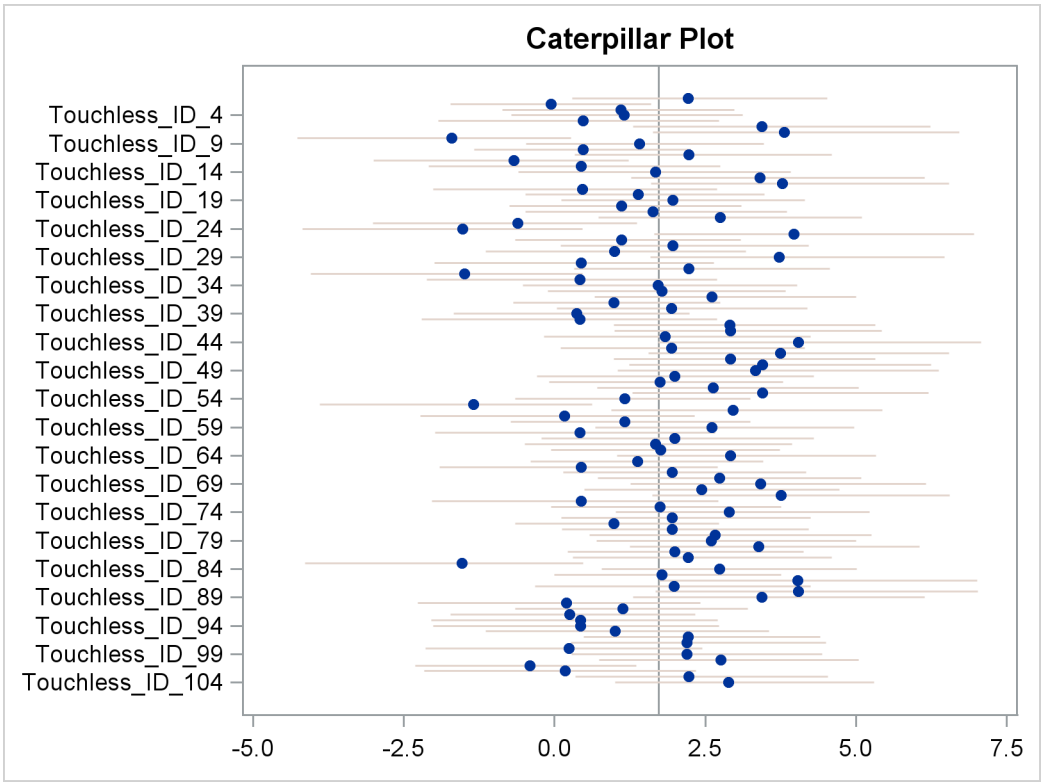
A caterpillar plot is a side-by-side bar plot of the 95% intervals for multiple variables. A caterpillar plot is usually used to visualize and compare random-effects parameters, which can exist in large numbers in certain models. You can use the %CATER autocall macro to create a caterpillar plot. The %CATER macro requires you to specify an input data set and a list of variables that you want to plot.

In this example, the output data set Postsamp contains posterior draws for all the random-effects parameters. The following statements generate a caterpillar plot for Touchless:

```
%CATER(data=Postsamp, var=Touchless_ID_:)
```

Output 27.4.2 is a caterpillar plot of the random effects of touchless opening for the 104 participants in this study. It displays the heterogeneity in preferences for touchless opening among the sample. As shown, the individuals have diverse preferences for the feature of touchless opening for a trash can. Heterogeneity of preferences is an important aspect of the model, and ignoring its presence can lead to incorrect inferences.

Output 27.4.2 Caterpillar Plot of the Random-Effects Parameters



Example 27.5: Heterogeneity Affected by Individual Characteristics

Rossi, Allenby, and McCulloch (2005) studied a scanner panel data about purchases of margarine. The data were first analyzed in Allenby and Rossi (1991) and are about purchases of ten brands of margarine. This example considers a subset of data about six margarine brands: Parkay stick, Blue Bonnet stick, Fleischmann's stick, a house brand stick, a generic stick, and Shedd's Spread tub. There are 313 households, which made a total of 3,405 purchases. Information about a few demographic characteristics of these households (income and family size) is expected to have effects on the central location of the distribution of heterogeneity.

The data set, which is called Sashelp.Margarin, comes from the SASHELP library.

```
proc print data=Sashelp.Margarin (obs=24);
   by HouseID Set;
   id HouseID Set;
run;
```

The data for the first four choice sets are shown in Figure 27.5.1.

Output 27.5.1 Data for the First Four Choice Sets

HouseID	Set	Choice	Brand	LogPrice	LogInc	FamSize
2100016	**1**	1	PPk	-0.41552	3.48124	2
		0	PBB	-0.40048	3.48124	2
		0	PFl	0.08618	3.48124	2
		0	PHse	-0.56212	3.48124	2
		0	PGen	-1.02165	3.48124	2
		0	PSS	-0.16252	3.48124	2

HouseID	Set	Choice	Brand	LogPrice	LogInc	FamSize
2100016	**2**	1	PPk	-0.46204	3.48124	2
		0	PBB	-0.40048	3.48124	2
		0	PFl	-0.01005	3.48124	2
		0	PHse	-0.56212	3.48124	2
		0	PGen	-1.02165	3.48124	2
		0	PSS	-0.16252	3.48124	2

HouseID	Set	Choice	Brand	LogPrice	LogInc	FamSize
2100016	**3**	1	PPk	-1.23787	3.48124	2
		0	PBB	-0.69315	3.48124	2
		0	PFl	-0.01005	3.48124	2
		0	PHse	-0.56212	3.48124	2
		0	PGen	-1.02165	3.48124	2
		0	PSS	-0.23572	3.48124	2

HouseID	Set	Choice	Brand	LogPrice	LogInc	FamSize
2100016	**4**	1	PPk	-0.47804	3.48124	2
		0	PBB	-0.49430	3.48124	2
		0	PFl	-0.01005	3.48124	2
		0	PHse	-0.56212	3.48124	2
		0	PGen	-1.02165	3.48124	2
		0	PSS	-0.16252	3.48124	2

The variable HouseID represents the household ID, and each household made at least five purchases, which are defined by Set. The variable Choice represents the choice made among the six margarine brands for each purchase or choice set. The variable Brand has the value PPK for Parkay stick, PBB for Blue Bonnet stick, PFL for Fleischmann's stick, PHse for the house brand stick, PGen for the generic stick, and PSS for Shedd's Spread tub. The variable LogPrice is the logarithm of the product price. The variables LogInc and variable FamSize provide information about household income and family size, respectively.

The following statements fit a random-effects-only logit model using Gamerman Metropolis sampling

```
proc bchoice data=Sashelp.Margarin seed=123 nmc=40000 thin=2
   nthreads=4 plots=none;
   class Brand(ref='PPk') HouseID Set;
   model Choice = / choiceset=(HouseID Set);
   random  Brand LogPrice / subject=HouseID remean=(LogInc FamSize)
           type=un monitor=(1);
run;
```

The REMEAN=(LOGINC FAMSIZE) option in the RANDOM statement requests estimation of the nonzero mean of the random effects, which is a function of household income and family size. No fixed effects are specified in the MODEL statement.

The NMC= option in the PROC BCHOICE statement requests to run the MCMC chain for 40,000 iterations, and the THIN=2 option keeps one of every two samples for analysis. If ODS Graphics is enabled and you do not specify the PLOTS option, then PROC BCHOICE produces, for each parameter, a panel that contains the trace plot, the autocorrelation function plot, and the kernel density plot by default. In models that have a large number of parameters, it will take a long time to display all the plots. So PLOTS=NONE is specified to stop producing the plots. If you want to see the plots for some parameters, you can save the posterior samples to a SAS data set using the OUTPOST= option and then generate the plots using the %TADPLOT macro as explained in the section "Regenerating Diagnostics Plots" on page 1084.

Summary statistics for the mean matrix of the random coefficients ($\boldsymbol{\Gamma}$), the covariance of the random coefficients ($\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$), and the random coefficients ($\boldsymbol{\gamma}_i$) for the first household are shown in Output 27.5.2.

Output 27.5.2 Posterior Summary Statistics

The BCHOICE Procedure

Posterior Summaries and Intervals

Parameter	Subject	N	Mean	Standard Deviation	95% HPD Interval	
REMean Brand PBB		20000	-1.2079	0.6384	-2.4293	0.0686
REMean Brand PFl		20000	-3.2484	1.9276	-7.0351	0.5201
REMean Brand PGen		20000	-5.1130	1.2332	-7.6390	-2.8030
REMean Brand PHse		20000	-3.2595	0.9194	-5.0725	-1.4761
REMean Brand PSS		20000	0.0915	1.2127	-2.3015	2.4981
REMean LogPrice		20000	-3.4148	0.8359	-5.0397	-1.7620
REMean Brand PBB LogInc		20000	0.0529	0.2114	-0.3485	0.4811
REMean Brand PFl LogInc		20000	0.7596	0.6208	-0.4726	1.9749
REMean Brand PGen LogInc		20000	-0.5079	0.4019	-1.2977	0.2698
REMean Brand PHse LogInc		20000	0.0315	0.3029	-0.5949	0.5931
REMean Brand PSS LogInc		20000	-0.6315	0.4131	-1.4645	0.1555
REMean LogPrice LogInc		20000	-0.2837	0.2817	-0.8434	0.2631
REMean Brand PBB FamSize		20000	-0.0274	0.0959	-0.2180	0.1572
REMean Brand PFl FamSize		20000	-0.7357	0.3059	-1.3283	-0.1267
REMean Brand PGen FamSize		20000	0.5775	0.1824	0.2269	0.9428
REMean Brand PHse FamSize		20000	0.2365	0.1357	-0.0291	0.4997
REMean Brand PSS FamSize		20000	0.0528	0.1974	-0.3347	0.4425
REMean LogPrice FamSize		20000	0.1010	0.1273	-0.1542	0.3424
RECov Brand PBB, Brand PBB		20000	2.2081	0.3730	1.5261	2.9615
RECov Brand PFl, Brand PBB		20000	1.8598	0.9106	0.1374	3.6776
RECov Brand PFl, Brand PFl		20000	12.0894	3.9050	5.5544	20.0864
RECov Brand PGen, Brand PBB		20000	1.9842	0.5697	0.8563	3.0829
RECov Brand PGen, Brand PFl		20000	1.3300	1.9065	-2.4622	5.1401
RECov Brand PGen, Brand PGen		20000	8.3897	1.4835	5.5924	11.3433
RECov Brand PHse, Brand PBB		20000	1.5148	0.4402	0.6799	2.3928
RECov Brand PHse, Brand PFl		20000	2.1554	1.3869	-0.5797	4.9030
RECov Brand PHse, Brand PGen		20000	5.7576	0.9570	3.8799	7.6015
RECov Brand PHse, Brand PHse		20000	5.4834	0.8441	3.9355	7.1970
RECov Brand PSS, Brand PBB		20000	1.1860	0.6287	-0.0412	2.4223
RECov Brand PSS, Brand PFl		20000	0.6096	1.9471	-3.0709	4.6460
RECov Brand PSS, Brand PGen		20000	4.8189	1.1738	2.5960	7.1020
RECov Brand PSS, Brand PHse		20000	3.4484	0.8805	1.7068	5.1649
RECov Brand PSS, Brand PSS		20000	8.7098	1.8047	5.4222	12.2276
RECov LogPrice, Brand PBB		20000	-0.2260	0.3462	-0.8764	0.4813
RECov LogPrice, Brand PFl		20000	2.1909	0.9010	0.5220	4.0796
RECov LogPrice, Brand PGen		20000	-0.9989	0.6371	-2.2175	0.2793
RECov LogPrice, Brand PHse		20000	-0.4254	0.5043	-1.3751	0.6150
RECov LogPrice, Brand PSS		20000	0.1734	0.6757	-1.1897	1.4765
RECov LogPrice, LogPrice		20000	2.1279	0.5373	1.1697	3.2261
Brand PBB	HouseID 2100016	20000	-2.3546	1.0548	-4.4528	-0.3948
Brand PFl	HouseID 2100016	20000	-3.9792	2.7840	-9.4491	0.9694
Brand PGen	HouseID 2100016	20000	-6.5984	1.6472	-9.8601	-3.4056
Brand PHse	HouseID 2100016	20000	-2.9722	1.2014	-5.5148	-0.8522
Brand PSS	HouseID 2100016	20000	-3.3807	2.1701	-7.4577	0.7600
LogPrice	HouseID 2100016	20000	-4.6129	1.2361	-6.9996	-2.1437

Table 27.11 collects the posterior means and standard deviations of $\bar{\mathbf{\Gamma}}$ that are shown in Output 27.5.2. The first column corresponds to the parameters that are specified in the model, namely the brands and price. The second column shows the average part-worths of each brand (versus the brand, Parkay stick) and the price at LogInc=0 and FamSize=0. The LogInc and FamSize columns list the modifying effects on the preference for each brand and price by household income and family size, respectively. Larger families show more interest in the generic and house brands and tend to stay away from the Fleischmann's brand. For example, consider the part-worth estimates for Fleischmann's. The posterior mean for REMean Brand PFI FamSize (the Fleischmann's row and the Famsize column) is –0.74 with a standard deviation of 0.31, meaning that an additional unit increase in family size is associated with a reduction of 0.74 in the estimated part-worth for Fleischmann's. In general, the demographics of households are only weakly associated with preference for brand and price. These results are in good agreement with those of Rossi, Allenby, and McCulloch (2005).

Table 27.11 Posterior Mean and Standard Deviation of $\mathbf{\Gamma}$

Parameter	Intercept	LogInc	FamSize
Blue Bonnet	REMean Brand PBB –1.21 0.64	REMean Brand PBB LogInc 0.05 0.21	REMean Brand PBB FamSize –0.03 0.10
Fleisch-mann's	REMean Brand PFI –3.25 1.93	REMean Brand PFI LogInc 0.76 0.62	REMean Brand PFI FamSize –0.74 0.31
Generic	REMean Brand PGen –5.11 1.23	REMean Brand PGen LogInc –0.51 0.40	REMean Brand PGen FamSize 0.58 0.18
House	REMean Brand PHse –3.26 0.92	REMean Brand PHse LogInc 0.03 0.30	REMean Brand PHse FamSize 0.24 0.14
Shedd's Spread	REMean Brand PSS 0.09 1.21	REMean Brand PSS LogInc –0.63 0.41	REMean Brand PSS FamSize –0.05 0.20
LogPrice	REMean LogPrice –3.41 0.84	REMean LogPrice LogInc –0.28 0.28	REMean LogPrice FamSize 0.10 0.13

Because the demographic variables are not zero-centered, the Intercept column shows the average part-worths of each brand and price for households with LogInc=0 and FamSize=0, which are not very meaningful. It is better to center demographic variables by their means, so that the posterior means listed in the Intercept column can be interpreted as the part-worths of a household that has an average income and average size.

Nevertheless, you can obtain the utilities of households that have any income levels and sizes. For example, the average part-worth of the Fleischmann's brand for a household with average income (LogInc=3.1) and family size (FamSize=3) would be as follows, because the estimated LogInc coefficient is 0.76 and the estimated FamSize coefficient is –0.74 for Fleischmann's:

$$-3.25 + 0.76 \times 3.1 - 0.74 \times 3 = -3.11$$

You can obtain part-worths for all other brands and compare their popularity among average households.

The posterior means and standard deviations of the covariance matrix of the random coefficients ($\mathbf{\Omega}_{\boldsymbol{\gamma}}$) are displayed by parameters that are labeled "RECov Brand PBB, Brand PBB," "RECov Brand PFI, Brand PBB,"

and so on. Some of the diagonal terms are fairly large, indicating that there is quite a bit of heterogeneity between households in margarine brand preference and price sensitivity. The covariance between the generic and house brands, "RECov Brand PHse, Brand PGen," is fairly large, suggesting that household preferences for these two brands are highly correlated.

The next set of parameters, which are displayed in Output 27.5.2, contain the estimates for the random effects for the first household.

Example 27.6: Quantities of Interest and Willingness To Pay

It is easy to give a model an interpretation in terms of a probability ratio (PR). The PR of choosing alternative a instead of alternative b while holding all other attributes or conditions the same is

$$\mathrm{PR}_{ab} = \frac{P_a}{P_b} = \frac{\exp(\beta_a)}{\exp(\beta_b)} = \exp(\beta_a - \beta_b)$$

where P_a and P_b are the probabilities of choosing alternative a and b, respectively.

We consider the fabric softener data simulated from Kuhfeld (2010). There are five fictitious fabric softener brands involved: Sploosh, Plumbbob, Platter, Moosey, and Another, where Another is the reference brand. Most brands were available at these three prices: 1.49, 1.99, and 2.49, except that the Another brand was only offered at 1.49.

```
data fabric;
   input Subj Set Choice Price Brand $ @@;
   datalines;
1       1       0       1.99    Sploosh
1       1       0       1.99    Plumbbob
1       1       1       1.99    Platter
1       1       0       2.49    Moosey
1       1       0       1.99    Another
1       2       0       2.49    Sploosh
1       2       0       1.49    Plumbbob
1       2       1       1.49    Platter
1       2       0       1.99    Moosey

   ... more lines ...

50      18      0       2.49    Sploosh
50      18      0       2.49    Plumbbob
50      18      1       1.99    Platter
50      18      0       2.49    Moosey
50      18      0       1.99    Another
;
```

The following statements fit a logit model with fixed effects only.

```
proc bchoice data=fabric seed=123 out=fabric_out;
   class Brand(ref='Another' order=data) Subj Set;
   model Choice = Brand Price / choiceset=(Subj Set);
run;
```

The posterior summary statistics are shown in Output 27.6.1. The Another brand is the base category, whose part-worth is normalized to zero. The most to least preferred brands are: Platter, Moosey, Another, Plumbbob, and Sploosh. The average part-worth for having to pay an extra dollar is negative (–4.6).

Output 27.6.1 PROC BCHOICE Posterior Summary Statistics

The BCHOICE Procedure

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Brand Sploosh	5000	-1.3133	0.2141	-1.7429	-0.8967
Brand Plumbbob	5000	-0.4940	0.1818	-0.8256	-0.1232
Brand Platter	5000	2.0984	0.1457	1.8241	2.3891
Brand Moosey	5000	0.6273	0.1598	0.3313	0.9504
Price	5000	-4.6210	0.2177	-5.0567	-4.2141

The PR of choosing Sploosh as opposed to Another (which is the reference category) simplifies to

$$\text{PR}_{(\text{Sploosh vs. Another})} = \frac{\exp(-1.31)}{\exp(0)} = 0.27$$

This ratio indicates that the likelihood of choosing Sploosh is 0.27 times that of choosing Another. Sploosh is less favorable than Another.

The PR of choosing Sploosh instead of Plumbbob is

$$\text{PR}_{(\text{Sploosh vs. Plumbbob})} = \frac{\exp(-1.31)}{\exp(-0.49)} = 0.44$$

This shows that Sploosh is less preferred than Plumbbob at the same price.

You can derive the probabilities for any alternatives and any combinations of attributes, even imaginary ones. Suppose now we want to also take price into consideration. For example, the PR of choosing Sploosh at \$1.49 as opposed to Plumbbob at \$1.99 is

$$\frac{\exp(\beta_{\text{Sploosh}} + 1.49 * \beta_{\text{Price}})}{\exp(\beta_{\text{Plumbbob}} + 1.99 * \beta_{\text{Price}})} = \frac{\exp(-1.31 - 1.49 * 4.62)}{\exp(-0.49 - 1.99 * 4.62)} = 4.44$$

This indicates that the likelihood of choosing Sploosh at \$1.49 is 4.44 times that of choosing Plumbbob at \$1.99. Offering some discount on price makes Sploosh more preferred than Plumbbob.

Although it is easy to obtain the point estimates, sometimes you might want to estimate other quantities, such as the standard deviations or various quantiles. If you are interested in making inference based on any quantities that are transformations of the random variables, you can do it either directly in PROC BCHOICE or by using the DATA step after you run the simulation.

Suppose you want to set the reference category as Plumbbob so that you can directly compare any other brand with it, you can specify REF='PLUMBBOB' in the CLASS statement and rerun the PROC BCHOICE simulation. You can also use the DATA step to calculate the quantities of interest. The following DATA step uses the simulated values of Brand_Sploosh and Brand_Plumbbob from the output data set fabric_out to create a new series of posterior samples for Sploosh_Plumbbob, a new variable that is created to calculate the PR of these two brands:

```
data Transout;
   set fabric_out;
   Sploosh_Plumbbob=exp(Brand_Sploosh - Brand_Plumbbob);
run;
```

Then you can use some autocall macros to analyze the posterior samples of Sploosh_Plumbbob. For example, the %SUMINT macro provides summary statistics and 95% HPD intervals as described in the section "Autocall Macros for Postprocessing" on page 1081.

```
%SUMINT(data=Transout, var=Sploosh_Plumbbob)
```

The results from using the %SUMINT macro are shown in Output 27.6.2. The posterior mean of the PR of choosing Sploosh over Plumbbob is about the same as the one that is computed just above. It also provides the standard deviation and 95% HPD interval, which indicates how spread out the distribution of the PR is.

Output 27.6.2 Sploosh versus Plumbbob

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Sploosh_Plumbbob	5000	0.4497	0.0907	0.2722	0.6327

The ratio of coefficients in most choice models has economic meanings. Part-worths are measure in utility units. Can we convert these to dollar equivalents? The answer is to compute the Willingness To Pay (WTP).

One's WTP for attribute a versus attribute b is the increase in purchase price that keeps the utilities equal. We can approximate WTP by dividing the difference in the attributes' part-worths by the price coefficient, given that the price coefficient is less than zero.

$$\text{WTP(a vs. b)} = \frac{\beta_a - \beta_b}{\beta_{\text{price}}}$$

If WTP(a vs. b) = s and s > 0, we say that attribute b is preferred than attribute a when offered at the same price and one is willing to pay up to the amount s to stay with b; If WTP(a vs. b) = s and s < 0, we say that attribute a is preferred than attribute b when offered at the same price and one is willing to pay up to the amount |s| to stay with a.

$$\text{WTP(Sploosh vs. Plumbbob)} = \frac{-1.31 + 0.49}{-4.62} = 0.18$$

Therefore, most subjects prefer Plumbbob over Sploosh at the same price; and they are willing to pay up to 0.18 dollars more to stay with Plumbbob.

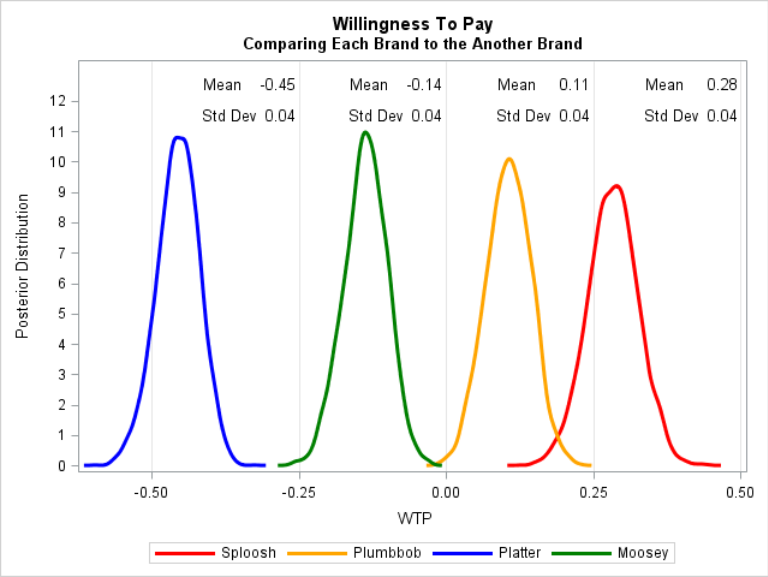
We can even approximate the posterior distributions of the WTPs comparing each brand to Another by using the DATA step:

```
data WTPout;
   set fabric_out;
   Sploosh_WTP = Brand_Sploosh/price;
   Plumbbob_WTP= Brand_Plumbbob/price;
   Platter_WTP = Brand_Platter/price;
```

```
   Moosey_WTP   = Brand_Moosey/price;
run;
```

The posterior distributions of the WTPs is plotted in Figure 27.6.3. The WTP of Sploosh versus Another and the WTP of Plumbbob versus Another are both located to the right of the zero line, meaning that Another is more popular than both Sploosh and Plumbbob; and the WTP of Sploosh versus Another is further away from the zero line compared to the WTP of Plumbbob versus Another, indicating that people are willing to pay more money to stay with Another before switching to Sploosh than switching to Plumbbob. Obtaining the whole distribution of the WTPs provides more information than having just the mean: it displays how well the mean estimates and how spread out the distribution is.

Output 27.6.3 Willingness to Pay Plot



Example 27.7: Predict the Choice Probabilities

This example shows how to obtain the posterior predictive distribution of the choice probability that each alternative is chosen from a choice set. The posterior predictive distribution enables you to get the expected choice probabilities of all the alternatives in the data, or even to predict market share for simulated or hypothetical products or marketplaces that do not directly reflect the choice set in the data.

Suppose you have a data set that contains the attribute variables (the design matrix) for all the alternatives in a choice set. For example, in the candy study earlier in the chapter, in the section "A Simple Logit Model Example" on page 1030, you can use the same eight alternatives: Dark is 1 for dark chocolate and 0 for soft chocolate; Soft is 1 for soft center and 0 for chewy center; Nuts is 1 if the candy contains nuts and 0 if it contains no nuts. The following data set contains the eight alternatives:

```
data DesignMatrix;
   input Dark Soft Nuts;
   datalines;
0 0 0
0 0 1
0 1 0
```

```
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1
;
```

You can use the PREDDIST statement, which obtains samples from the posterior predictive distribution of each of the choice probabilities by using the posterior samples of parameters in the model:

```
proc bchoice data=Chocs outpost=Bsamp nmc=10000 thin=2 seed=124;
   class Dark(ref='0') Soft(ref='0') Nuts(ref='0') Subj;
   model Choice = Dark Soft Nuts / choiceset=(Subj);
   preddist covariates=DesignMatrix nalter=8 outpred=Predout;
run;

%SUMINT(data=Predout, var=Prob_1_:)
```

In the PREDDIST statement, the COVARIATES= option names the data set to contain the explanatory variable values for which the predictions are established. This data set must contain data that have the same variables that the model uses. The NALTER= option specifies the number of alternatives in each choice set in the COVARIATES= data set. All choice sets in the data must have the same number of alternatives. If you omit the COVARIATES= option, the DATA= data set that you specify in the PROC BCHOICE statement is used instead. The OUTPRED= option creates an output data set to contain the samples from the posterior predictive distribution of the choice probabilities. Then you can use SAS autocall macros to analyze the posterior samples. For example, the %SUMINT macro provides summary statistics (means and standard deviations) and the 95% HPD intervals.

You can predict the choice probabilities by using the means of the posterior distributions. The results from using the %SUMINT macro are shown in Output 27.7.1. There is only one choice set for choice probability prediction, in which there are a total of eight alternatives. This explains the parameter names in the first column of the output, where the first number indexes the choice sets and the second number indexes the alternatives in each choice set. The most preferred chocolate candy is the sixth one, Dark/Chewy/Nuts, which takes about half the market.

Output 27.7.1 Choice of Chocolate Candies

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Prob_1_1	5000	0.0554	0.0425	0.00395	0.1434
Prob_1_2	5000	0.1309	0.0801	0.0136	0.2865
Prob_1_3	5000	0.00686	0.00896	0.000165	0.0244
Prob_1_4	5000	0.0158	0.0179	0.000448	0.0535
Prob_1_5	5000	0.2102	0.1033	0.0297	0.3974
Prob_1_6	5000	0.4946	0.1315	0.2407	0.7439
Prob_1_7	5000	0.0262	0.0279	0.000607	0.0830
Prob_1_8	5000	0.0601	0.0526	0.00329	0.1669

Example 27.8: MaxDiff Choice Example

This example uses a MaxDiff type of choice experiment from the Sydney transport project. The study sought to help set priorities for future improvements in public transportation. The data were collected by the Center for the Study of Choice (CenSoC) at the University of Technology Sydney, and analyzed previously by Magidson and Vermunt (2015).

Respondents were asked to choose from among nine short-term improvements (nine alternatives). Table 27.12 lists the nine improvements that were evaluated.

Table 27.12 Alternatives in Sydney Transport Project

Index	Improvement Alternatives
1	More frequent off-peak trains between major centers
2	Improved peak rail capacity
3	More frequent bus service on major routes
4	Extension of light rail services
5	Integrated fares
6	Integrated ticketing
7	Real-time arrival information
8	New cycleways, more bike and scooter parking
9	Use of green power for trains

```
data Sydney;
   input RespID Set Choice Index @@;
   datalines;
5 1 1 2 5 1 0 4 5 1 -1 8 5 2 0 1 5 2 1 4 5 2 -1 5 5 3 0 4 5 3 -1 7 5 3 1
9 5 4 1 3 5 4 0 4 5 4 -1 6 5 5 -1 1 5 5 1 2 5 5 0 3 5 6 1 2 5 6 -1 5 5 6
0 7 5 7 1 2 5 7 -1 6 5 7 0 9 5 8 -1 1 5 8 0 8 5 8 1 9 5 9 0 5 5 9 -1 6 5
9 1 8 5 10 1 3 5 10 -1 7 5 10 0 8 5 11 1 1 5 11 0 6 5 11 -1 7 5 12 1 3 5
12 -1 5 5 12 0 9 6 1 1 2 6 1 0 4 6 1 -1 8 6 2 1 1 6 2 -1 4 6 2 0 5 6 3
-1 4 6 3 1 7 6 3 0 9 6 4 0 3 6 4 -1 4 6 4 1 6 6 5 -1 1 6 5 1 2 6 5 0 3 6
6 1 2 6 6 0 5 6 6 -1 7 6 7 1 2 6 7 0 6 6 7 -1 9 6 8 1 1 6 8 -1 8 6 8 0 9
6 9 0 5 6 9 1 6 6 9 -1 8 6 10 1 3 6 10 0 7 6 10 -1 8 6 11 1 1 6 11 0 6 6
11 -1 7 6 12 1 3 6 12 0 5 6 12 -1 9 7 1 1 2 7 1 0 4 7 1 -1 8 7 2 1 1 7 2

   ... more lines ...

350 1 -1 8 350 2 0 1 350 2 -1 4 350 2 1 5 350 3 0 4 350 3 -1 7 350 3 1 9
350 4 0 3 350 4 -1 4 350 4 1 6 350 5 -1 1 350 5 1 2 350 5 0 3 350 6 1 2
350 6 0 5 350 6 -1 7 350 7 1 2 350 7 0 6 350 7 -1 9 350 8 1 1 350 8 -1 8
350 8 0 9 350 9 0 5 350 9 1 6 350 9 -1 8 350 10 1 3 350 10 -1 7 350 10 0
8 350 11 0 1 350 11 1 6 350 11 -1 7 350 12 1 3 350 12 0 5 350 12 -1 9
;

proc print data=Sydney(obs=18);
run;
```

Specifically, each of the 200 respondents was given 12 choice tasks; each task consisted of a subset of three of these nine short-term improvements (alternatives). For each of these tasks, the respondent was asked to

choose an alternative that should receive the highest priority (indicated by 1) and an alternative that should receive the lowest priority (indicated by –1).

The data for the first several choice sets are shown in Output 27.8.1. The variable RespID identifies each respondent. The variable Set indexes the 12 choice tasks that are given to each respondent. The variable Choice is the response variable that represents the decision: 1 for the highest priority, –1 for the lowest priority, and 0 for those in between. The variable Index has values from 1 to 9, each of which represents one of the nine improvement alternatives. This variable is listed in the CLASS statement as a classification variable, and the last level, 9, is the reference level. The respondent, with RespID=5, answered 12 choice questions, but the table shows only the first six choice tasks. This person picked the second alternative (improved peak rail capacity) as the highest priority, and the eighth alternative (new cycleways, more bike and scooter parking) as the lowest priority.

Output 27.8.1 Data for the First Several MaxDiff Choice Sets

Obs	RespID	Set	Choice	Index
1	5	1	1	2
2	5	1	0	4
3	5	1	-1	8
4	5	2	0	1
5	5	2	1	4
6	5	2	-1	5
7	5	3	0	4
8	5	3	-1	7
9	5	3	1	9
10	5	4	1	3
11	5	4	0	4
12	5	4	-1	6
13	5	5	-1	1
14	5	5	1	2
15	5	5	0	3
16	5	6	1	2
17	5	6	-1	5
18	5	6	0	7

The following statements fit a logit model with random effects for this MaxDiff choice experiment by specifying CHOICETYPE=MAXDIFF:

```
proc bchoice data=Sydney seed=1 nthreads=8 nmc=20000 thin=2 plots=none;
   class Index (ref=last) RespID Set;
   model Choice =  / choiceset=(RespID Set) choicetype=maxdiff;
   random Index / subject=RespID remean type=un;
run;
```

The NMC= option specifies the number of posterior simulation iterations after burn-in, and the THIN=2 option keeps one of every two samples for analysis. PLOTS=NONE is specified to suppress producing the plots to save time. The REMEAN option in the RANDOM statement requests estimation of the mean of the random effects. No fixed effects are specified in the MODEL statement.

The "Model Information" table in Output 27.8.2 shows that the type of choice response is MaxDiff.

Output 27.8.2 Model Information

The BCHOICE Procedure

Model Information	
Data Set	WORK.SYDNEY
Response Variable	Choice
Type of Model	Logit
Type of Choice Response	MaxDiff
Fixed Effects Included	No
Random Effects Included	Yes
Sampling Algorithm	Gamerman Metropolis
Burn-In Size	500
Simulation Size	20000
Thinning	2
Random Number Seed	1
Number of Threads	8

As shown in the "Choice Sets Summary" table in Output 27.8.3, there are a total of 2,400 choice sets (200 respondents, each with 12 choice sets), all of which display the same pattern: each has a total of three alternatives, one of which was chosen as the highest priority (best), another one was chosen as the lowest priority (worst), and the remaining one was not chosen.

Output 27.8.3 Choice Sets Summary

Choice Sets Summary for MaxDiff Data

Pattern	Choice Sets	Total Alternatives	Chosen Best	Chosen Worst	Not Chosen
1	2400	3	1	1	1

Summary statistics for the mean of the random effects $\bar{\boldsymbol{\gamma}}$ (labeled as "REMEAN") and the covariance of the random effects $\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$ (labeled as "RECOV") are shown in Output 27.8.4. On average, the second improvement alternative (Index=2, improved peak rail capacity) has the largest part-worth value of 3.85, and the third improvement alternative (Index=3, more frequent bus service on major routes) has the next-largest part-worth value. The part-worth value of the last alternative (Index=9) is structural 0. The ranking for the nine improvements from the highest priority to the lowest is 2, 3, 1, 5, 6, 9, 7, 4, and 8. It seems that in general people are most concerned about transportation on major routes during busy hours.

Output 27.8.4 PROC BCHOICE Posterior Summary Statistics

Posterior Summaries and Intervals					
Parameter	N	Mean	Standard Deviation	95% HPD Interval	
REMean Index 1	10000	1.4291	0.3042	0.8229	2.0192
REMean Index 2	10000	3.8554	0.3467	3.1852	4.5356
REMean Index 3	10000	3.0110	0.3108	2.4244	3.6337
REMean Index 4	10000	-0.8314	0.3217	-1.4560	-0.1999
REMean Index 5	10000	0.8937	0.3063	0.2722	1.4739
REMean Index 6	10000	0.7749	0.3183	0.1597	1.4073
REMean Index 7	10000	-0.5741	0.3278	-1.2157	0.0646
REMean Index 8	10000	-1.5394	0.3073	-2.1602	-0.9580
RECov Index 1, Index 1	10000	14.8555	2.1808	10.6273	19.1673
RECov Index 2, Index 1	10000	12.8307	2.1035	8.9311	17.1101
RECov Index 2, Index 2	10000	17.2483	2.6550	12.4063	22.6886
RECov Index 3, Index 1	10000	11.4254	1.9311	7.7419	15.2444
RECov Index 3, Index 2	10000	13.2266	2.1780	9.0158	17.4365
RECov Index 3, Index 3	10000	14.2821	2.1736	10.2170	18.6448
RECov Index 4, Index 1	10000	11.4926	1.9608	7.7660	15.3728
RECov Index 4, Index 2	10000	9.9519	2.0063	6.0542	13.8420
RECov Index 4, Index 3	10000	10.5877	1.9189	6.8862	14.3772
RECov Index 4, Index 4	10000	16.6336	2.4546	12.0061	21.5206
RECov Index 5, Index 1	10000	11.3054	1.9592	7.6339	15.2169
RECov Index 5, Index 2	10000	11.6876	2.0882	7.9589	16.0430
RECov Index 5, Index 3	10000	10.8130	1.9255	7.3641	14.8192
RECov Index 5, Index 4	10000	10.6285	1.9907	7.0367	14.7534
RECov Index 5, Index 5	10000	15.2144	2.2827	10.9980	19.7565
RECov Index 6, Index 1	10000	11.3246	1.9902	7.4668	15.1618
RECov Index 6, Index 2	10000	11.7650	2.1211	7.8721	16.0694
RECov Index 6, Index 3	10000	10.5438	1.9467	6.9584	14.4123
RECov Index 6, Index 4	10000	10.9368	2.0576	7.2383	15.1956
RECov Index 6, Index 5	10000	15.3265	2.2869	10.9703	19.7437
RECov Index 6, Index 6	10000	16.8353	2.4481	12.3717	21.7751
RECov Index 7, Index 1	10000	13.0307	2.0579	9.0988	17.1055
RECov Index 7, Index 2	10000	12.4041	2.1295	8.4571	16.6625
RECov Index 7, Index 3	10000	11.0318	1.9352	7.4545	14.9278
RECov Index 7, Index 4	10000	11.3357	2.0386	7.5180	15.4328
RECov Index 7, Index 5	10000	13.3085	2.1345	9.4358	17.5905
RECov Index 7, Index 6	10000	13.2184	2.1730	9.2114	17.5802
RECov Index 7, Index 7	10000	18.0400	2.5687	13.1280	23.0490
RECov Index 8, Index 1	10000	7.2473	1.6590	4.1723	10.5612
RECov Index 8, Index 2	10000	8.1009	1.8005	4.7421	11.6407
RECov Index 8, Index 3	10000	8.2427	1.7017	5.1639	11.6972
RECov Index 8, Index 4	10000	9.2959	1.8450	5.8923	12.9848
RECov Index 8, Index 5	10000	7.9252	1.7191	4.6876	11.2987
RECov Index 8, Index 6	10000	7.8835	1.7567	4.6412	11.3654
RECov Index 8, Index 7	10000	7.5465	1.7841	4.2349	11.1415
RECov Index 8, Index 8	10000	14.3947	2.2484	10.2301	18.7904

The covariance matrix ($\boldsymbol{\Omega}_{\boldsymbol{\gamma}}$) characterizes the extent of heterogeneity among individuals. Large diagonal elements of the covariance matrix indicate substantial heterogeneity in part-worths. Off-diagonal elements indicate patterns in the evaluation of alternatives. Most of the diagonal elements of the matrix are large, indicating that there is a large difference among people in the priorities they would set for future improvements in public transportation.

Example 27.9: Allocation Choice Example

Consider an example of allocation choice data in the MAMR study [2]. In this study, researchers aimed to learn about oncologists' preferences on different scenarios of pharmaceutical therapies and benefits of these therapies in relation to the survival of patients with metastatic breast cancer.

```
data MAMR;
   input ID Set None TrtScheme RespRate Time Status TrtLine Risk Cost Alloc @@;
   datalines;
1     1     1     3     3     1     2     1     6     1     0.2
1     1     1     5     1     1     1     3     4     2     0
1     1     1     7     2     2     2     3     6     2     0
1     1     0     0     0     0     0     0     0     0     0.8
1     2     1     1     1     2     2     1     1     1     0.12
1     2     1     3     3     1     1     2     2     1     0.4
1     2     1     5     1     2     1     3     4     3     0
1     2     0     0     0     0     0     0     0     0     0.48
1     3     1     4     3     1     2     1     4     4     0.08

   ... more lines ...

146   10    0     0     0     0     0     0     0     0     0
146   11    1     1     2     1     1     3     6     1     0.34
146   11    1     2     2     2     1     3     3     1     0.28
146   11    1     1     2     2     2     1     1     1     0.38
146   11    0     0     0     0     0     0     0     0     0
;

proc print data=MAMR(obs=20);
run;
```

The data for the first several choice sets are shown in Output 27.9.1.

[2]The data were provided by Hernan Talledo, Ph.D., associate consultant at Datum International, Lima, Peru, and advanced market research professor at Universidad San Ignacio de Loyola and University of the Pacific in Lima; and Thomas C. Eagle, Ph.D, president of Eagle Analytics of California, Inc.

Output 27.9.1 Data for the First Several Allocation Choice Sets

Obs	ID	Set	None	TrtScheme	RespRate	Time	Status	TrtLine	Risk	Cost	Alloc
1	1	1	1	3	3	1	2	1	6	1	0.20
2	1	1	1	5	1	1	1	3	4	2	0.00
3	1	1	1	7	2	2	2	3	6	2	0.00
4	1	1	0	0	0	0	0	0	0	0	0.80
5	1	2	1	1	1	2	2	1	1	1	0.12
6	1	2	1	3	3	1	1	2	2	1	0.40
7	1	2	1	5	1	2	1	3	4	3	0.00
8	1	2	0	0	0	0	0	0	0	0	0.48
9	1	3	1	4	3	1	2	1	4	4	0.08
10	1	3	1	4	3	2	2	1	5	3	0.20
11	1	3	1	1	1	2	2	1	6	1	0.40
12	1	3	0	0	0	0	0	0	0	0	0.32
13	1	4	1	1	1	2	1	1	6	1	0.60
14	1	4	1	2	2	1	2	2	6	1	0.20
15	1	4	1	3	3	2	1	3	6	1	0.04
16	1	4	0	0	0	0	0	0	0	0	0.16
17	1	5	1	3	2	2	1	1	6	1	0.60
18	1	5	1	4	3	1	2	2	4	2	0.00
19	1	5	1	5	2	1	1	1	4	3	0.20
20	1	5	0	0	0	0	0	0	0	0	0.20

The variable ID represents the IDs for the 133 oncologists in the study. Each oncologist was presented with 11 choice tasks, each consisting of four scenarios of therapies (including the NONE option), and was asked to state the number of times he or she would pick each alternative. These numbers of times were then transformed into percentages, which sum to 100% within each choice task. The variable Alloc is the response variable that gives a percentage of preference for each alternative, and the sum of percentages in each choice set is 100%. Table 27.13 lists all the attributes and their levels.

Table 27.13 Attributes in the MAMR Study

Attribute	Label	Level	Description
TrtScheme	Treatment scheme	0	No treatment
		1	Taxotere + Adriamicina + SG 22.4 months + SLP 9.3 months
		2	Taxotere + Xeloda + SG 14.7 months + SLP 6.2 months
		3	Taxotere + Gemcitabina + SG 18.6 months + SLP 6.1 months
		4	Taxol + Avastin + SG 26.7 months + SLP 11.8 months
		5	Taxotere + Avastin + SG 83% a year + SLP 8.8 months
		6	Taxol weekly 90mg/m2 + SLP 6.2 months
		7	Taxotere 100mg/m2 + SG 15 months + SLP 6.5 months
RespRate	Response rate	1	40%
		2	50%
		3	60%
TrtTime	Treatment time	1	6 months
		2	Until cure
Status	Status	1	0–1 (better)
		2	>=2
TrtLine	Treatment lines	1	First line
		2	Second line
		3	Third line
Risk	Risk factors	1	Cardiovascular disease
		2	Hepatopathy moderate to severe
		3	Renal failure
		4	Thromboembolism history
		5	Arterial Hypertension (HTA)
		6	No risk
Cost	Cost per month	1	1,000
		2	3,000
		3	5,000
		4	7,000

The following statements fit a logit model with random effects for these allocation choice data by specifying CHOICETYPE=ALLOCATION:

```
proc bchoice data=MAMR seed=124 nthreads=8 nmc=20000 nthin=2 plots=none;
   class  TrtScheme(ref=first) RespRate Time Status TrtLine Risk Cost ID Set
          / ref=last;
   model  Alloc = / choiceset=(ID Set) choicetype=allocation;
   random TrtScheme RespRate Time Status TrtLine Risk Cost
          / subject=ID remean;
run;
```

The NMC= option requests to run the MCMC chain for 20,000 iterations, and the THIN=2 option keeps one of every two samples for analysis. PLOTS=NONE is specified, so the plots will not be produced to save time.

The REMEAN option in the RANDOM statement requests estimation of the mean of the random effects.

When CHOICETYPE=ALLOCATION, you can specify a positive integer in the WEIGHT= suboption to indicate how many choice sets the allocated percentages should apply to. For example, if the respondent were asked to imagine the next 20 occasions and to allocate a percentage of those 20 occasions for each alternative, you would specify WEIGHT=20. By default, WEIGHT=10. In this example, the default (WEIGHT=10) is close to the nature of the original data collection.

The "Model Information" table, as shown in Output 27.9.2, shows that the type of choice response is allocation.

Output 27.9.2 Model Information

The BCHOICE Procedure

Model Information	
Data Set	WORK.MAMR
Response Variable	Alloc
Type of Model	Logit
Type of Choice Response	Allocation
Fixed Effects Included	No
Random Effects Included	Yes
Sampling Algorithm	Gamerman Metropolis
Burn-In Size	500
Simulation Size	20000
Thinning	2
Random Number Seed	124
Number of Threads	8

As shown in the "Choice Sets Summary" table in Output 27.9.3, there are a total of 1,463 choice sets, all of which display the same pattern: they have a total of four alternatives, and the sum of percentages is 100%. If the sum of percentages within one choice set is not equal to 100%, the choice set is invalid. PROC BCHOICE deletes all invalid choice sets and produces a warning message.

Output 27.9.3 Choice Sets Summary

Choice Sets Summary for Allocation Data				
Pattern	Choice Sets	Total Alternatives	Percent Sum(%)	Not Chosen(%)
1	1463	4	100	0

As you see in Output 27.9.4, TrtScheme, Status, TrtLine, and Risk have impact on the choices. It seems that treatments (TrtScheme) with shorter periods of SG and SLP follow-up are more preferred than those with longer follow-up times. The oncologists in the study tend to choose the therapies that would have the best performance status (Status); the part-worth for level 1 (better) of Status is positive compared to the referent level 2 of Status, which is set at 0. More important, risk factors (Risk) have a large influence on the choice of various treatments: the estimates of the part-worths for all risk factors are all negative and have relatively large values, reflecting that doctors try to avoid them as much as possible, especially treatments with serious side effects. One surprising result is that therapy cost does not play a role in the decision-making process. But when you realize that the respondents are doctors, not their patients, and that the study deals with metastatic cancer, it might not be hard to figure out why.

Output 27.9.4 PROC BCHOICE Posterior Summary Statistics

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
REMean TrtScheme 1	10000	-0.3178	0.1647	-0.6514	-0.00866
REMean TrtScheme 2	10000	-0.5021	0.1507	-0.8010	-0.2145
REMean TrtScheme 3	10000	-0.3978	0.1476	-0.6803	-0.1057
REMean TrtScheme 4	10000	0.1785	0.1480	-0.1180	0.4624
REMean TrtScheme 5	10000	-0.4356	0.1499	-0.7329	-0.1380
REMean TrtScheme 6	10000	0.1791	0.1746	-0.1612	0.5265
REMean TrtScheme 7	10000	-0.0555	0.1468	-0.3571	0.2126
REMean RespRate 1	10000	-0.0493	0.0781	-0.1982	0.1076
REMean RespRate 2	10000	-0.0348	0.0771	-0.1877	0.1147
REMean Time 1	10000	-0.0371	0.0717	-0.1765	0.1056
REMean Status 1	10000	0.1893	0.0741	0.0456	0.3344
REMean TrtLine 1	10000	0.4689	0.0835	0.3045	0.6285
REMean TrtLine 2	10000	0.2883	0.0806	0.1306	0.4465
REMean Risk 1	10000	-1.2123	0.1782	-1.5556	-0.8573
REMean Risk 2	10000	-0.8808	0.1398	-1.1627	-0.6108
REMean Risk 3	10000	-0.9594	0.1601	-1.2731	-0.6476
REMean Risk 4	10000	-1.4509	0.1794	-1.8124	-1.1090
REMean Risk 5	10000	-1.4393	0.1732	-1.7814	-1.1135
REMean Cost 1	10000	0.1649	0.1300	-0.0943	0.4162
REMean Cost 2	10000	-0.0330	0.1061	-0.2371	0.1751
REMean Cost 3	10000	-0.0936	0.1174	-0.3364	0.1253
REVar TrtScheme 1	10000	1.7718	0.3062	1.2306	2.4020
REVar TrtScheme 2	10000	1.0636	0.2037	0.6920	1.4722
REVar TrtScheme 3	10000	1.2343	0.2166	0.8439	1.6742
REVar TrtScheme 4	10000	1.9933	0.3025	1.4457	2.6027
REVar TrtScheme 5	10000	1.9061	0.3168	1.2955	2.5264
REVar TrtScheme 6	10000	1.3845	0.2915	0.8668	1.9748
REVar TrtScheme 7	10000	0.9070	0.1750	0.5886	1.2585
REVar RespRate 1	10000	0.4506	0.0673	0.3290	0.5842
REVar RespRate 2	10000	0.4533	0.0676	0.3295	0.5859
REVar Time 1	10000	0.4296	0.0625	0.3097	0.5496
REVar Status 1	10000	0.4670	0.0692	0.3404	0.6071
REVar TrtLine 1	10000	0.5918	0.0910	0.4225	0.7684
REVar TrtLine 2	10000	0.4901	0.0748	0.3570	0.6436
REVar Risk 1	10000	1.3516	0.3257	0.7761	2.0061
REVar Risk 2	10000	1.2343	0.2634	0.7722	1.7622
REVar Risk 3	10000	1.1511	0.2694	0.6700	1.6896
REVar Risk 4	10000	2.6897	0.5148	1.7562	3.7233
REVar Risk 5	10000	2.7399	0.4715	1.8703	3.6810
REVar Cost 1	10000	1.0116	0.1675	0.7030	1.3472
REVar Cost 2	10000	0.7321	0.1159	0.5201	0.9731
REVar Cost 3	10000	0.9321	0.1783	0.6065	1.2881

References

Albert, J. H., and Chib, S. (1993). "Bayesian Analysis of Binary and Polychotomous Response Data." *Journal of the American Statistical Association* 88:669–679.

Allenby, G. M. (1997). "An Introduction to Hierarchical Bayesian Modeling." American Marketing Association, Advanced Research Techniques Forum, Tutorial Notes.

Allenby, G. M., and Lenk, P. (1994). "Modeling Household Purchase Behavior with Logistic Normal Regression." *Journal of the American Statistical Association* 89:1218–1231.

Allenby, G. M., and Rossi, P. E. (1991). "Quality Perceptions and Asymmetric Switching between Brands." *Marketing Science* 10:185–205.

Allenby, G. M., and Rossi, P. E. (1999). "Marketing Models of Consumer Heterogeneity." *Journal of Econometrics* 89:57–78.

Bhat, C. R. (1995). "A Heteroscedastic Extreme Value Model of Intercity Travel Mode Choice." *Transportation Research* 29:471–483.

Eilers, P. H. C., and Marx, B. D. (1996). "Flexible Smoothing with *B*-Splines and Penalties." *Statistical Science* 11:89–121. With discussion.

Finn, A., and Louviere, J. J. (1992). "Determining the Appropriate Response to Evidence of Public Concern: The Case of Food Safety." *Journal of Public Policy and Marketing* 11:12–25.

Gamerman, D. (1997). "Sampling from the Posterior Distribution in Generalized Linear Models." *Statistics and Computing* 7:57–68.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. 2nd ed. London: Chapman & Hall.

Herriges, J. A., and Kling, C. L. (1996). "Testing the Consistency of Nested Logit Models with Utility Maximization." *Economics Letters* 50:33–39.

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.

Kling, C. L., and Herriges, J. A. (1995). "An Empirical Investigation of the Consistency of Nested Logit Models with Utility Maximization." *American Journal of Agricultural Economics* 77:875–884.

Kuhfeld, W. F. (2010). *Marketing Research Methods in SAS*. Technical report, SAS Institute Inc., Cary, NC. `http://support.sas.com/resources/papers/tnote/tnote_marketresearch.html`.

Lahiri, K., and Gao, J. (2002). "Bayesian Analysis of Nested Logit Model by Markov Chain Monte Carlo." *Journal of Econometrics* 11:103–133.

Louviere, J. J. (1991). "Best-Worst Scaling: A Model for the Largest Difference Judgements." Working paper, University of Alberta, Edmonton.

Magidson, J., and Vermunt, J. K. (2015). "Analyzing MaxDiff Data with Scale Factors: Applications of Latent GOLD Basic + Choice 5.1." Latent GOLD Tutorial 8A, Statistical Innovations Inc., Belmont, MA. http://www.statisticalinnovations.com/wp-content/uploads/LGChoice_tutorial_8A.pdf.

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

McCulloch, R., and Rossi, P. E. (1994). "An Exact Likelihood Analysis of the Multinomial Probit Model." *Journal of Econometrics* 64:207–240.

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

McFadden, D. (1978). "Modelling the Choice of Residential Location." In *Spatial Interaction Theory and Planning Models*, edited by A. Karlqvist, L. Lundqvist, F. Snickars, and J. Weibull, 75–96. Amsterdam: North-Holland.

McFadden, D. (2001). "Economic Choices." *American Economic Review* 91:351–378.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). "Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms." *Annals of Applied Probability* 7:110–120.

Roberts, G. O., and Rosenthal, J. S. (2001). "Optimal Scaling for Various Metropolis-Hastings Algorithms." *Statistical Science* 16:351–367.

Rossi, P. E. (2012). "Bayesm Package: Bayesian Inference for Marketing/Micro-econometrics." http://www.perossi.org/home/bsm-1.

Rossi, P. E. (2013). Personal communication.

Rossi, P. E., Allenby, G. M., and McCulloch, R. (2005). *Bayesian Statistics and Marketing*. Chichester, UK: John Wiley & Sons.

Rossi, P. E., McCulloch, R., and Allenby, G. M. (1996). "The Value of Purchase History Data in Target Marketing." *Marketing Science* 15:321–340.

Schervish, M. J. (1995). *Theory of Statistics*. New York: Springer-Verlag.

Train, K. E. (2009). *Discrete Choice Methods with Simulation*. 2nd ed. Cambridge: Cambridge University Press.

Train, K. E., McFadden, D. L., and Ben-Akiva, M. (1987). "The Demand for Local Telephone Service: A Fully Discrete Model of Residential Calling Patterns and Service Choice." *Rand Journal of Economics* 18:109–123.

Subject Index

Syntax Index