SAS/ETS® 15.1 User's Guide

The MDC Procedure

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

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

SAS/ETS® 15.1 User's Guide

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

November 2018

Chapter 23
The MDC Procedure

Contents

Overview: MDC Procedure

The MDC (multinomial discrete choice) procedure analyzes models in which the choice set consists of multiple alternatives. This procedure supports conditional logit, mixed logit, heteroscedastic extreme value, nested logit, and multinomial probit models. The MDC procedure uses the maximum likelihood (ML) or simulated maximum likelihood method for model estimation. The term *multinomial logit* is often used in the econometrics literature to refer to the *conditional logit* model of McFadden (1974). Here, the term *conditional logit* refers to McFadden's conditional logit model, and the term *multinomial logit* refers to a model that differs slightly. Early applications of the multinomial logit model in the econometrics literature are provided by Schmidt and Strauss (1975); Theil (1969). The main difference between McFadden's conditional logit model and the multinomial logit model is that the multinomial logit model makes the choice probabilities depend on the characteristics of the individuals only, whereas the conditional logit model considers the effects of choice attributes on choice probabilities as well.

Unordered multiple choices are observed in many settings in different areas of application. For example, choices of housing location, occupation, political party affiliation, type of automobile, and mode of transportation are all unordered multiple choices. Economics and psychology models often explain observed choices by using the *random utility* function. The utility of a specific choice can be interpreted as the relative pleasure or happiness that the decision maker derives from that choice with respect to other alternatives in a finite choice set. It is assumed that the individual chooses the alternative for which the associated utility is highest. However, the utilities are not known to the analyst with certainty and are therefore treated by the analyst as random variables. When the utility function contains a random component, the individual choice behavior becomes a probabilistic process.

The random utility function of individual i for choice j can be decomposed into deterministic and stochastic components

$$U_{ij} = V_{ij} + \epsilon_{ij}$$

where V_{ij} is a deterministic utility function, assumed to be linear in the explanatory variables, and ϵ_{ij} is an unobserved random variable that captures the factors that affect utility that are not included in V_{ij}. Different assumptions on the distribution of the errors, ϵ_{ij}, give rise to different classes of models.

The features of discrete choice models available in the MDC procedure are summarized in Table 23.1.

Table 23.1 Summary of Models Supported by PROC MDC

Model Type	Utility Function	Distribution of ϵ_{ij}
Conditional logit	$U_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \epsilon_{ij}$	IEV, independent and identical
HEV	$U_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \epsilon_{ij}$	HEV, independent and nonidentical
Nested logit	$U_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \epsilon_{ij}$	GEV, correlated and identical
Mixed logit	$U_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \xi_{ij} + \epsilon_{ij}$	IEV, independent and identical
Multinomial probit	$U_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \epsilon_{ij}$	MVN, correlated and nonidentical

IEV stands for type I extreme-value (or Gumbel) distribution with the probability density function and the cumulative distribution function of the random error given by $f(\epsilon_{ij}) = \exp(-\epsilon_{ij})\exp(-\exp(-\epsilon_{ij}))$ and $F(\epsilon_{ij}) = \exp(-\exp(-\epsilon_{ij}))$. HEV stands for heteroscedastic extreme-value distribution with the probability density function and the cumulative distribution function of the random error given by $f(\epsilon_{ij}) = \frac{1}{\theta_j}\exp(\frac{\epsilon_{ij}}{\theta_j})\exp[-\exp(-\frac{\epsilon_{ij}}{\theta_j})]$ and $F(\epsilon_{ij}) = \exp[-\exp(-\frac{\epsilon_{ij}}{\theta_j})]$, where θ_j is a scale parameter for the random component of the jth alternative. GEV stands for generalized extreme-value distribution. MVN represents multivariate normal distribution; and ξ_{ij} is an error component. For more information about ξ_{ij}, see the section "Mixed Logit Model" on page 1382. .

Getting Started: MDC Procedure

Conditional Logit: Estimation and Prediction

The MDC procedure is similar in use to the other regression model procedures in the SAS System. However, the MDC procedure requires identification and choice variables. For example, consider a random utility function

$$U_{ij} = x_{1,ij}\beta_1 + x_{2,ij}\beta_2 + \epsilon_{ij} \quad j = 1, \ldots, 3$$

where the cumulative distribution function of the stochastic component is a Type I extreme value, $F(\epsilon_{ij}) = \exp(-\exp(-\epsilon_{ij}))$. You can estimate this conditional logit model with the following statements:

```
proc mdc;
   model decision = x1 x2 / type=clogit
      choice=(mode 1 2 3);
   id pid;
run;
```

Note that the MDC procedure, unlike other regression procedures, does not include the intercept term automatically. The dependent variable decision takes the value 1 when a specific alternative is chosen; otherwise, it takes the value 0. Each individual is allowed to choose one and only one of the possible alternatives. In other words, the variable decision takes the value 1 one time only for each individual. If each individual has three elements (1, 2, and 3) in the choice set, the NCHOICE=3 option can be specified instead of CHOICE=(mode 1 2 3).

Consider the following trinomial data from Daganzo (1979). The original data (origdata) contain travel time (ttime1–ttime3) and choice (choice) variables. The variables ttime1–ttime3 are the travel times for three different modes of transportation, and choice indicates which one of the three modes is chosen. The choice variable must have integer values.

```
data origdata;
   input ttime1 ttime2 ttime3 choice @@;
datalines;
16.481  16.196  23.89   2  15.123  11.373  14.182  2
19.469  8.822   20.819  2  18.847  15.649  21.28   2
12.578  10.671  18.335  2  11.513  20.582  27.838  1
10.651  15.537  17.418  1  8.359   15.675  21.05   1

   ... more lines ...
```

A new data set (newdata) is created because PROC MDC requires that each individual decision maker has one case for each alternative in his choice set. Note that the ID statement is required for all MDC models. In the following example, there are two public transportation modes, 1 and 2, and one private transportation mode, 3, and all individuals share the same choice set.

The first nine observations of the raw data set are shown in Figure 23.1.

Figure 23.1 Initial Choice Data

Obs	ttime1	ttime2	ttime3	choice
1	16.481	16.196	23.890	2
2	15.123	11.373	14.182	2
3	19.469	8.822	20.819	2
4	18.847	15.649	21.280	2
5	12.578	10.671	18.335	2
6	11.513	20.582	27.838	1
7	10.651	15.537	17.418	1
8	8.359	15.675	21.050	1
9	11.679	12.668	23.104	1

The following statements transform the data according to MDC procedure requirements:

```
data newdata(keep=pid decision mode ttime);
   set origdata;
   array tvec{3} ttime1 - ttime3;
   retain pid 0;
   pid + 1;
   do i = 1 to 3;
      mode = i;
      ttime = tvec{i};
      decision = ( choice = i );
      output;
   end;
run;
```

The first nine observations of the transformed data set are shown in Figure 23.2.

Figure 23.2 Transformed Modal Choice Data

Obs	pid	mode	ttime	decision
1	1	1	16.481	0
2	1	2	16.196	1
3	1	3	23.890	0
4	2	1	15.123	0
5	2	2	11.373	1
6	2	3	14.182	0
7	3	1	19.469	0
8	3	2	8.822	1
9	3	3	20.819	0

The decision variable, decision, must have one nonzero value for each decision maker that corresponds to the actual choice. When the RANK option is specified, the decision variable must contain rank data. For more details, see the section "MODEL Statement" on page 1367. The following SAS statements estimate the conditional logit model by using maximum likelihood:

```
proc mdc data=newdata;
   model decision = ttime /
            type=clogit
            nchoice=3
            optmethod=qn
            covest=hess;
   id pid;
run;
```

The MDC procedure enables different individuals to have different choice sets. When all individuals have the same choice set, the NCHOICE= option can be used instead of the CHOICE= option. However, the NCHOICE= option is not allowed when a nested logit model is estimated. When the NCHOICE=*number* option is specified, the choices are generated as $1, \ldots, number$. For more flexible alternatives (for example, 1, 3, 6, 8), you need to use the CHOICE= option. The choice variable must have integer values.

The OPTMETHOD=QN option specifies the quasi-Newton optimization technique. The covariance matrix of the parameter estimates is obtained from the Hessian matrix because COVEST=HESS is specified. You can also specify COVEST=OP or COVEST=QML. For more information, see the section "MODEL Statement" on page 1367.

The MDC procedure produces a summary of model estimation displayed in Figure 23.3. Since there are multiple observations for each individual, the "Number of Cases" (150)—that is, the total number of choices faced by all individuals—is larger than the number of individuals, "Number of Observations" (50).

Figure 23.3 Estimation Summary Table

The MDC Procedure

Conditional Logit Estimates

Model Fit Summary	
Dependent Variable	decision
Number of Observations	50
Number of Cases	150
Log Likelihood	-33.32132
Log Likelihood Null (LogL(0))	-54.93061
Maximum Absolute Gradient	2.97024E-6
Number of Iterations	6
Optimization Method	Dual Quasi-Newton
AIC	68.64265
Schwarz Criterion	70.55467

Figure 23.4 shows the frequency distribution of the three choice alternatives. In this example, mode 2 is most frequently chosen.

Figure 23.4 Choice Frequency

Discrete Response Profile			
Index	**CHOICE**	**Frequency**	**Percent**
0	1	14	28.00
1	2	29	58.00
2	3	7	14.00

The MDC procedure computes nine goodness-of-fit measures for the discrete choice model. Seven of them are pseudo-R-square measures based on the null hypothesis that all coefficients except for an intercept term are zero (Figure 23.5). McFadden's likelihood ratio index (LRI) is the smallest in value. For more details, see the section "Model Fit and Goodness-of-Fit Statistics" on page 1390.

Figure 23.5 Likelihood Ratio Test and R-Square Measures

Goodness-of-Fit Measures		
Measure	**Value**	**Formula**
Likelihood Ratio (R)	43.219	2 * (LogL - LogL0)
Upper Bound of R (U)	109.86	- 2 * LogL0
Aldrich-Nelson	0.4636	R / (R+N)
Cragg-Uhler 1	0.5787	1 - exp(-R/N)
Cragg-Uhler 2	0.651	(1-exp(-R/N)) / (1-exp(-U/N))
Estrella	0.6666	1 - (1-R/U)^(U/N)
Adjusted Estrella	0.6442	1 - ((LogL-K)/LogL0)^(-2/N*LogL0)
McFadden's LRI	0.3934	R / U
Veall-Zimmermann	0.6746	(R * (U+N)) / (U * (R+N))
N = # of observations, K = # of regressors		

Finally, the parameter estimate is displayed in Figure 23.6.

Figure 23.6 Parameter Estimate of Conditional Logit

The MDC Procedure

Conditional Logit Estimates

Parameter Estimates					
Parameter	**DF**	**Estimate**	**Standard Error**	**t Value**	**Approx Pr > |t|**
ttime	1	-0.3572	0.0776	-4.60	<.0001

The predicted choice probabilities are produced using the OUTPUT statement:

```
output out=probdata pred=p;
```

The parameter estimates can be used to forecast the choice probability of individuals that are not in the input data set. To do so, you need to append to the input data set extra observations whose values of the dependent variable decision are missing, since these extra observations are not supposed to be used in the estimation stage. The identification variable pid must have values that are not used in the existing observations. The output data set, probdata, contains a new variable, p, in addition to input variables in the data set extdata.

The following statements forecast the choice probability of individuals that are not in the input data set:

```
data extra;
   input pid mode decision ttime;
datalines;
51  1  .   5.0
51  2  .  15.0
51  3  .  14.0
;

data extdata;
   set newdata extra;
run;

proc mdc data=extdata;
   model decision = ttime /
            type=clogit
            covest=hess
            nchoice=3;
   id pid;
   output out=probdata pred=p;
run;

proc print data=probdata( where=( pid >= 49 ) );
   var mode decision p ttime;
   id pid;
run;
```

The last nine observations from the forecast data set (probdata) are displayed in Figure 23.7. It is expected that the decision maker will choose mode "1" based on predicted probabilities for all modes.

Figure 23.7 Out-of-Sample Mode Choice Forecast

pid	mode	decision	p	ttime
49	1	0	0.46393	11.852
49	2	1	0.41753	12.147
49	3	0	0.11853	15.672
50	1	0	0.06936	15.557
50	2	1	0.92437	8.307
50	3	0	0.00627	22.286
51	1	.	0.93611	5.000
51	2	.	0.02630	15.000
51	3	.	0.03759	14.000

Nested Logit Modeling

A more general model can be specified using the nested logit model.

Consider, for example, the following random utility function:

$$U_{ij} = x_{ij}\beta + \epsilon_{ij} \quad j = 1, \ldots, 3$$

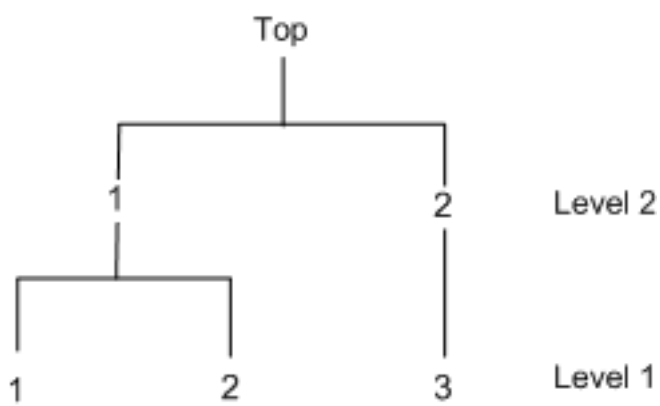
Suppose the set of all alternatives indexed by j is partitioned into K nests, $B_1, \ldots, B_K$. The nested logit model is obtained by assuming that the error term in the utility function has the GEV cumulative distribution function

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

where λ_k is a measure of a degree of independence among the alternatives in nest k. When $\lambda_k = 1$ for all k, the model reduces to the standard logit model.

Since the public transportation modes, 1 and 2, tend to be correlated, these two choices can be grouped together. The decision tree displayed in Figure 23.8 is constructed.

Figure 23.8 Decision Tree for Model Choice



The two-level decision tree is specified in the NEST statement. The NCHOICE= option is not allowed for nested logit estimation. Instead, the CHOICE= option needs to be specified, as in the following statements:

```
/*-- nested logit estimation --*/
proc mdc data=newdata;
   model decision = ttime /
            type=nlogit
            choice=(mode 1 2 3)
            covest=hess;
   id pid;
   utility u(1,) = ttime;
   nest level(1) = (1 2 @ 1, 3 @ 2),
        level(2) = (1 2 @ 1);
run;
```

In Figure 23.9, estimates of the inclusive value parameters, INC_L2G1C1 and INC_L2G1C2, are indicative of a nested model structure. For more information about inclusive values, see the sections "Nested Logit" on page 1386 and "Decision Tree and Nested Logit" on page 1388.

Figure 23.9 Two-Level Nested Logit Estimates

The MDC Procedure

Nested Logit Estimates

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
ttime_L1	1	-0.4040	0.1241	-3.25	0.0011
INC_L2G1C1	1	0.8016	0.4352	1.84	0.0655
INC_L2G1C2	1	0.8087	0.3591	2.25	0.0243

The nested logit model is estimated with the restriction INC_L2G1C1 = INC_L2G1C2 by specifying the SAMESCALE option, as in the following statements:

```
/*-- nlogit with samescale option --*/
proc mdc data=newdata;
   model decision = ttime /
            type=nlogit
            choice=(mode 1 2 3)
            samescale
            covest=hess;
   id pid;
   utility u(1,) = ttime;
   nest level(1) = (1 2 @ 1, 3 @ 2),
        level(2) = (1 2 @ 1);
run;
```

The estimation result is displayed in Figure 23.10.

Figure 23.10 Nested Logit Estimates with One Dissimilarity Parameter

The MDC Procedure

Nested Logit Estimates

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
ttime_L1	1	-0.4025	0.1217	-3.31	0.0009
INC_L2G1	1	0.8209	0.3019	2.72	0.0066

The nested logit model is equivalent to the conditional logit model if INC_L2G1C1 = INC_L2G1C2 = 1. You can verify this relationship by estimating a constrained nested logit model as shown in the following statements. (For more information about imposing linear restrictions on parameter estimates, see the section "RESTRICT Statement" on page 1376.)

```
/*-- constrained nested logit estimation --*/
proc mdc data=newdata;
   model decision = ttime /
            type=nlogit
            choice=(mode 1 2 3)
            covest=hess;
```

```
   id pid;
   utility u(1,) = ttime;
   nest level(1) = (1 2 @ 1, 3 @ 2),
        level(2) = (1 2 @ 1);
   restrict INC_L2G1C1 = 1, INC_L2G1C2 =1;
run;
```

The parameter estimates and the active linear constraints for the constrained nested logit model are displayed in Figure 23.11.

Figure 23.11 Constrained Nested Logit Estimates

The MDC Procedure

Nested Logit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Parameter Label
ttime_L1	1	-0.3572	0.0776	-4.60	<.0001	
INC_L2G1C1	0	1.0000	0			
INC_L2G1C2	0	1.0000	0			
Restrict1	1	-2.1706	8.4098	-0.26	0.7993*	Linear EC [1]
Restrict2	1	3.6573	10.0001	0.37	0.7186*	Linear EC [2]

*** Probability computed using beta distribution.**

Linearly Independent Active Linear Constraints							
1	0	=	-1.0000	+	1.0000	*	INC_L2G1C1
2	0	=	-1.0000	+	1.0000	*	INC_L2G1C2

Multivariate Normal Utility Function

Consider the random utility function

$$U_{ij} = \text{ttime}_{ij}\beta + \epsilon_{ij}, \quad j = 1, 2, 3$$

where

$$\begin{bmatrix} \epsilon_{i1} \\ \epsilon_{i2} \\ \epsilon_{i3} \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} 1 & \rho_{21} & 0 \\ \rho_{21} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

The correlation coefficient (ρ_{21}) between U_{i1} and U_{i2} represents commonly neglected attributes of public transportation modes, 1 and 2. The following SAS statements estimate this trinomial probit model:

```
/*-- homoscedastic mprobit --*/
proc mdc data=newdata;
   model decision = ttime /
            type=mprobit
            nchoice=3
            unitvariance=(1 2 3)
```

```
            covest=hess;
   id pid;
run;
```

The UNITVARIANCE=(1 2 3) option specifies that the random component of utility for each of these choices has unit variance. If the UNITVARIANCE= option is specified, it needs to include at least two choices. The results of this constrained multinomial probit model estimation are displayed in Figure 23.12 and Figure 23.13. The test for ttime = 0 is rejected at the 1% significance level.

Figure 23.12 Constrained Probit Estimation Summary

The MDC Procedure

Multinomial Probit Estimates

Model Fit Summary	
Dependent Variable	decision
Number of Observations	50
Number of Cases	150
Log Likelihood	-33.88604
Log Likelihood Null (LogL(0))	-54.93061
Maximum Absolute Gradient	0.0002380
Number of Iterations	8
Optimization Method	Dual Quasi-Newton
AIC	71.77209
Schwarz Criterion	75.59613
Number of Simulations	100
Starting Point of Halton Sequence	11

Figure 23.13 Multinomial Probit Estimates with Unit Variances

The MDC Procedure

Multinomial Probit Estimates

Parameter Estimates					
Parameter	**DF**	**Estimate**	**Standard Error**	**t Value**	**Approx Pr > |t|**
ttime	1	-0.2307	0.0472	-4.89	<.0001
RHO_21	1	0.4820	0.3135	1.54	0.1242

HEV and Multinomial Probit: Heteroscedastic Utility Function

When the stochastic components of utility are heteroscedastic and independent, you can model the data by using an HEV or a multinomial probit model. The HEV model assumes that the utility of alternative j for each individual i has heteroscedastic random components,

$$U_{ij} = V_{ij} + \epsilon_{ij}$$

where the cumulative distribution function of the Gumbel distributed ϵ_{ij} is

$$F(\epsilon_{ij}) = \exp(-\exp(-\epsilon_{ij}/\theta_j))$$

Note that the variance of ϵ_{ij} is $\frac{1}{6}\pi^2\theta_j^2$. Therefore, the error variance is proportional to the square of the scale parameter θ_j. For model identification, at least one of the scale parameters must be normalized to 1. The following SAS statements estimate an HEV model under a unit scale restriction for mode "1" ($\theta_1 = 1$):

```
/*-- hev with gauss-laguerre method --*/
proc mdc data=newdata;
   model decision = ttime /
            type=hev
            nchoice=3
            hev=(unitscale=1, integrate=laguerre)
            covest=hess;
   id pid;
run;
```

The results of computation are presented in Figure 23.14 and Figure 23.15.

Figure 23.14 HEV Estimation Summary

The MDC Procedure

Heteroscedastic Extreme Value Model Estimates

Model Fit Summary	
Dependent Variable	decision
Number of Observations	50
Number of Cases	150
Log Likelihood	-33.41383
Maximum Absolute Gradient	0.0000218
Number of Iterations	11
Optimization Method	Dual Quasi-Newton
AIC	72.82765
Schwarz Criterion	78.56372

Figure 23.15 HEV Parameter Estimates

The MDC Procedure

Heteroscedastic Extreme Value Model Estimates

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
ttime	1	-0.4407	0.1798	-2.45	0.0143
SCALE2	1	0.7765	0.4348	1.79	0.0741
SCALE3	1	0.5753	0.2752	2.09	0.0366

The parameters SCALE2 and SCALE3 in the output correspond to the estimates of the scale parameters θ_2 and θ_3, respectively.

Note that the estimate of the HEV model is not always stable because computation of the log-likelihood function requires numerical integration. Bhat (1995) proposed the Gauss-Laguerre method. In general, the log-likelihood function value of HEV should be larger than that of conditional logit because HEV models include the conditional logit as a special case. However, in this example the reverse is true (–33.414 for the HEV model, which is less than –33.321 for the conditional logit model). (See Figure 23.14 and Figure 23.3.) This indicates that the Gauss-Laguerre approximation to the true probability is too coarse. You can see how well the Gauss-Laguerre method works by specifying a unit scale restriction for all modes, as in the following statements, since the HEV model with the unit variance for all modes reduces to the conditional logit model:

```
/*-- hev with gauss-laguerre and unit scale --*/
proc mdc data=newdata;
   model decision = ttime /
            type=hev
            nchoice=3
            hev=(unitscale=1 2 3, integrate=laguerre)
            covest=hess;
   id pid;
run;
```

Figure 23.16 shows that the ttime coefficient is not close to that of the conditional logit model.

Figure 23.16 HEV Estimates with All Unit Scale Parameters

The MDC Procedure

Heteroscedastic Extreme Value Model Estimates

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
ttime	1	-0.2926	0.0438	-6.68	<.0001

There is another option of specifying the integration method. The INTEGRATE=HARDY option uses the adaptive Romberg-type integration method. The adaptive integration produces much more accurate probability and log-likelihood function values, but often it is not practical to use this method of analyzing the HEV model because it requires excessive CPU time. The following SAS statements produce the HEV estimates by using the adaptive Romberg-type integration method:

```
/*-- hev with adaptive integration --*/
proc mdc data=newdata;
   model decision = ttime /
             type=hev
             nchoice=3
             hev=(unitscale=1, integrate=hardy)
             covest=hess;
   id pid;
run;
```

The results are displayed in Figure 23.17 and Figure 23.18.

Figure 23.17 HEV Estimation Summary Using Alternative Integration Method

The MDC Procedure

Heteroscedastic Extreme Value Model Estimates

Model Fit Summary	
Dependent Variable	decision
Number of Observations	50
Number of Cases	150
Log Likelihood	-33.02598
Maximum Absolute Gradient	0.0001202
Number of Iterations	8
Optimization Method	Dual Quasi-Newton
AIC	72.05197
Schwarz Criterion	77.78803

Figure 23.18 HEV Estimates Using Alternative Integration Method

The MDC Procedure

Heteroscedastic Extreme Value Model Estimates

Parameter Estimates					
Parameter	**DF**	**Estimate**	**Standard Error**	**t Value**	**Approx Pr > |t|**
ttime	1	-0.4580	0.1861	-2.46	0.0139
SCALE2	1	0.7757	0.4283	1.81	0.0701
SCALE3	1	0.6908	0.3384	2.04	0.0412

With the INTEGRATE=HARDY option, the log-likelihood function value of the HEV model, –33.026, is greater than that of the conditional logit model, –33.321. (See Figure 23.17 and Figure 23.3.)

When you impose unit scale restrictions on all choices, as in the following statements, the HEV model gives the same estimates as the conditional logit model. (See Figure 23.19 and Figure 23.6.)

```
/*-- hev with adaptive integration and unit scale --*/
proc mdc data=newdata;
   model decision = ttime /
            type=hev
            nchoice=3
            hev=(unitscale=1 2 3, integrate=hardy)
            covest=hess;
   id pid;
run;
```

Figure 23.19 Alternative HEV Estimates with Unit Scale Restrictions

The MDC Procedure

Heteroscedastic Extreme Value Model Estimates

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
ttime	1	-0.3572	0.0776	-4.60	<.0001

For comparison, the following statements estimate a heteroscedastic multinomial probit model by imposing a zero restriction on the correlation parameter, $\rho_{31} = 0$. The MDC procedure requires normalization of at least two of the error variances in the multinomial probit model. Also, for identification, the correlation parameters associated with a unit normalized variance are restricted to be zero. When the UNITVARIANCE= option is specified, the zero restriction on correlation coefficients applies to the last choice of the list. In the following statements, the variances of the first and second choices are normalized. The UNITVARIANCE=(1 2) option imposes additional restrictions that $\rho_{32} = \rho_{21} = 0$. The default for the UNITVARIANCE= option is the last two choices (which would have been equivalent to UNITVARIANCE=(2 3) for this example). The result is presented in Figure 23.20.

The utility function can be defined as

$$U_{ij} = V_{ij} + \epsilon_{ij}$$

where

$$\boldsymbol{\epsilon}_i \sim N\left(\mathbf{0}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sigma_3^2 \end{bmatrix}\right)$$

```
/*-- mprobit estimation --*/
proc mdc data=newdata;
   model decision = ttime /
            type=mprobit
            nchoice=3
            unitvariance=(1 2)
            covest=hess;
   id pid;
   restrict RHO_31 = 0;
run;
```

Figure 23.20 Heteroscedastic Multinomial Probit Estimates

The MDC Procedure

Multinomial Probit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Parameter Label
ttime	1	-0.3206	0.0920	-3.49	0.0005	
STD_3	1	1.6913	0.6906	2.45	0.0143	
RHO_31	0	0	0			
Restrict1	1	1.1854	1.5490	0.77	0.4499*	Linear EC [1]

*** Probability computed using beta distribution.**

Note that in the output the estimates of standard errors and correlations are denoted by STD_i and RHO_ij, respectively. In this particular case the first two variances (STD_1 and STD_2) are normalized to one, and corresponding correlations (RHO_21 and RHO_32) are set to zero, so they are not listed among parameter estimates.

Parameter Heterogeneity: Mixed Logit

One way of modeling unobserved heterogeneity across individuals in their sensitivity to observed exogenous variables is to use the mixed logit model with a random parameters or random coefficients specification. The probability of choosing alternative j is written as

$$P_i(j) = \frac{\exp(\mathbf{x}'_{ij}\boldsymbol{\beta})}{\sum_{k=1}^{J}\exp(\mathbf{x}'_{ik}\boldsymbol{\beta})}$$

where $\boldsymbol{\beta}$ is a vector of coefficients that varies across individuals and $\mathbf{x}_{ij}$ is a vector of exogenous attributes.

For example, you can specify the distribution of the parameter β to be the normal distribution.

The mixed logit model uses a Monte Carlo simulation method to estimate the probabilities of choice. There are two simulation methods available. If the RANDNUM=PSEUDO option is specified in the MODEL statement, pseudo-random numbers are generated; if the RANDNUM=HALTON option is specified, Halton quasi-random sequences are used. The default value is RANDNUM=HALTON.

You can estimate the model with normally distributed random coefficients of ttime with the following SAS statements:

```
/*-- mixed logit estimation --*/
proc mdc data=newdata type=mixedlogit;
   model decision = ttime /
            nchoice=3
            mixed=(normalparm=ttime);
   id pid;
run;
```

Let β^m and β^s be mean and scale parameters, respectively, for the random coefficient, β. The relevant utility function is

$$U_{ij} = \text{ttime}_{ij}\beta + \epsilon_{ij}$$

where $\beta = \beta^m + \beta^s\eta$ (β^m and β^s are fixed mean and scale parameters, respectively). The stochastic component, η, is assumed to be standard normal since the NORMALPARM= option is given. Alternatively, the UNIFORMPARM= or LOGNORMALPARM= option can be specified. The LOGNORMALPARM= option is useful when nonnegative parameters are being estimated. The NORMALPARM=, UNIFORMPARM=, and LOGNORMALPARM= variables must be included in the right-hand side of the MODEL statement. For more information, see the section "Mixed Logit Model" on page 1382. To estimate a mixed logit model by using the transportation mode choice data, the MDC procedure requires the MIXED= option for random components. Results of the mixed logit estimation are displayed in Figure 23.21.

Figure 23.21 Mixed Logit Model Parameter Estimates

The MDC Procedure

Mixed Multinomial Logit Estimates

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
ttime_M	1	-0.5342	0.2184	-2.45	0.0144
ttime_S	1	0.2843	0.1911	1.49	0.1368

Note that the parameter ttime_M corresponds to the constant mean parameter β^m and the parameter ttime_S corresponds to the constant scale parameter β^s of the random coefficient β.

Syntax: MDC Procedure

The MDC procedure is controlled by the following statements:

PROC MDC *options* ;
  MDCDATA *options* ;
  BOUNDS *bound1* < , *bound2* . . . > ;
  BY *variables* ;
  CLASS *variables* ;
  ID *variable* ;
  MODEL *dependent-variable* = *regressors* / *options* ;
  NEST LEVEL(*level-number*) = ((*choices*)@(*choice*), . . . , (*choices*)@(*choice*)) ;
  NLOPTIONS *options* ;
  OUTPUT *options* ;
  RESTRICT *restriction1* < , *restriction2* . . . > ;
  TEST *options* ;
  UTILITY **U() =** *variables*, . . . , **U() =** *variables* ;

Functional Summary

Table 23.2 summarizes the statements and options used with the MDC procedure.

Table 23.2 Functional Summary

Description	Statement	Option
Data Set Options		
Formats the data for use by PROC MDC	MDCDATA	
Specifies the input data set	MDC	DATA=
Specifies the output data set for CLASS STATEMENT	CLASS	OUT =
Writes parameter estimates to an output data set	MDC	OUTEST=
Includes covariances in the OUTEST= data set	MDC	COVOUT
Writes linear predictors and predicted probabilities to an output data set	OUTPUT	OUT=
Declaring the Role of Variables		
Specifies the ID variable	ID	
Specifies BY-group processing variables	BY	
Printing Control Options		
Requests all printing options	MODEL	ALL
Displays correlation matrix of the estimates	MODEL	CORRB
Displays covariance matrix of the estimates	MODEL	COVB
Displays detailed information about optimization iterations	MODEL	ITPRINT
Suppresses all displayed output	MODEL	NOPRINT

Table 23.2 *continued*

Description	Statement	Option
Model Estimation Options		
Specifies the choice variables	MODEL	CHOICE=()
Specifies the convergence criterion	MODEL	CONVERGE=
Specifies the type of covariance matrix	MODEL	COVEST=
Specifies the starting point of the Halton sequence	MODEL	HALTONSTART=
Specifies options specific to the HEV model	MODEL	HEV=()
Sets the initial values of parameters used by the iterative optimization algorithm	MODEL	INITIAL=()
Specifies the maximum number of iterations	MODEL	MAXITER=
Specifies the options specific to mixed logit	MODEL	MIXED=()
Specifies the number of choices for each person	MODEL	NCHOICE=
Specifies the number of simulations	MODEL	NSIMUL=
Specifies the optimization technique	MODEL	OPTMETHOD=
Specifies the type of random number generators	MODEL	RANDNUM=
Specifies that initial values are generated using random numbers	MODEL	RANDINIT
Specifies the rank dependent variable	MODEL	RANK
Specifies optimization restart options	MODEL	RESTART=()
Specifies a restriction on inclusive parameters	MODEL	SAMESCALE
Specifies a seed for pseudo-random number generation	MODEL	SEED=
Specifies a stated preference data restriction on inclusive parameters	MODEL	SPSCALE
Specifies the type of the model	MODEL	TYPE=
Specifies normalization restrictions on multinomial probit error variances	MODEL	UNITVARIANCE=()
Controlling the Optimization Process		
Specifies upper and lower bounds for the parameter estimates	BOUNDS	
Specifies linear restrictions on the parameter estimates	RESTRICT	
Specifies nonlinear optimization options	NLOPTIONS	
Nested Logit Related Options		
Specifies the tree structure	NEST	LEVEL()=
Specifies the type of utility function	UTILITY	U()=
Output Control Options		
Outputs predicted probabilities	OUTPUT	P=
outputs estimated linear predictor	OUTPUT	XBETA=

Table 23.2 *continued*

Description	Statement	Option
Test Request Options		
Requests Wald, Lagrange multiplier, and likelihood ratio tests	TEST	ALL
Requests the Wald test	TEST	WALD
Requests the Lagrange multiplier test	TEST	LM
Requests the likelihood ratio test	TEST	LR

PROC MDC Statement

PROC MDC *options* **;**

The following *options* can be used in the PROC MDC statement:

DATA=*SAS-data-set*
specifies the input SAS data set. If the DATA= option is not specified, PROC MDC uses the most recently created SAS data set.

OUTEST=*SAS-data-set*
names the SAS data set that the parameter estimates are written to. For information about the contents of this data set, see the section "OUTEST= Data Set" on page 1392.

COVOUT
writes the covariance matrix for the parameter estimates to the OUTEST= data set. This option is valid only if the OUTEST= option is specified.

In addition, any of the following MODEL statement options can be specified in the PROC MDC statement, which is equivalent to specifying the option for the MODEL statement: ALL, CONVERGE=, CORRB, COVB, COVEST=, HALTONSTART=, ITPRINT, MAXITER=, NOPRINT, NSIMUL=, OPTMETHOD=, RANDINIT, RANK, RESTART=, SAMESCALE, SEED=, SPSCALE, TYPE=, and UNITVARIANCE=.

BOUNDS Statement

BOUNDS *bound1* < , *bound2* . . . > **;**

The BOUNDS statement imposes simple boundary constraints on the parameter estimates. BOUNDS statement constraints refer to the parameters estimated by the MDC procedure. You can specify any number of BOUNDS statements.

Each *bound* is composed of parameters, constants, and inequality operators:

item operator item < *operator item* < *operator item* . . . > > **;**

Each *item* is a constant, parameter, or list of parameters. Parameters associated with a regressor variable are referred to by the name of the corresponding regressor variable. Each *operator* is <, >, <=, or >=.

You can use both the BOUNDS statement and the RESTRICT statement to impose boundary constraints; however, the BOUNDS statement provides a simpler syntax for specifying these kinds of constraints. (See also the section "RESTRICT Statement" on page 1376.)

Lagrange multipliers are reported for all the active boundary constraints. In the displayed output, the Lagrange multiplier estimates are identified with the names Restrict1, Restrict2, and so on. The probability of the Lagrange multipliers is computed using a beta distribution (LaMotte 1994). Nonactive (nonbinding) bounds have no effect on the estimation results and are not noted in the output.

The following BOUNDS statement constrains the estimates of the coefficient of ttime to be negative and the coefficients of x1 through x10 to be between zero and one. This example illustrates the use of parameter lists to specify boundary constraints.

```
bounds ttime < 0,
       0 < x1-x10 < 1;
```

BY Statement

BY *variables* **;**

A BY statement can be used with PROC MDC to obtain separate analyses on observations in groups defined by the BY variables.

CLASS Statement

CLASS *variables* **;**

The CLASS statement names the classification variables to be used in the analysis. Classification variables can be either character or numeric.

ID Statement

ID *variable* **;**

The ID statement must be used with PROC MDC to specify the identification variable that controls multiple choice-specific cases. The MDC procedure requires only one ID statement even with multiple MODEL statements.

MDCDATA Statement

MDCDATA *options* < / **OUT=***SAS-data-set* > **;**

The MDCDATA statement prepares data for use by PROC MDC when the choice-specific information is stored in multiple variables (for example, see Figure 23.1 in the section "Conditional Logit: Estimation and Prediction" on page 1347).

VARLIST (*name1 = (var1 var2 . . .) name2 = (var1 var2 . . .) . . .***)**
creates *name* variables from a multiple-variable list of choice alternatives in parentheses. The choice-specific dummy variables are created for the first set of multiple variables. At least one set of multiple variables must be specified. The order of (*var1 var2* . . .) in the VARLIST option determines the numbering of the alternative; that is, *var1* corresponds to alternative 1, *var2* corresponds to alternative 2, and so on.

SELECT=(*variable***)**
specifies a variable that contains choices for each individual. The SELECT= *variable* needs to be a character-type variable, with values that match variable names in the first VARLIST option: *name1=(var1 var2 . . .)*.

ID=(*name***)**
creates a variable that identifies each individual.

ALT=(*name***)**
identifies selection alternatives for each individual.

DECVAR=(*name***)**
creates a 0/1 variable that indicates the choice made for each individual.

OUT=*SAS-data-set*
specifies a SAS data set to which modified data are output.

MODEL Statement

MODEL *dependent-variable = regressors* < / *options* > **;**

The MODEL statement specifies the dependent variable and independent regressor variables for the regression model. When the nested logit model is estimated, regressors in the UTILITY statement are used for estimation.

The following options can be used in the MODEL statement after a slash (/).

CHOICE=(*variables* **)**

CHOICE=(*variable numbers* **)**
specifies the variables that contain possible choices for each individual. Choice variables must have integer values. Multiple choice variables are allowed only for nested logit models and must be specified in order from the highest level to the lowest level. For example, CHOICE=(upmode, mode) indicates that the nested logit model has two levels. The choices at the upper level are described by the upmode variable, and the choices at the lower level are described by the mode variable. If all possible alternatives are written with the variable name, the MDC procedure checks all values of the choice variable. CHOICE=(X 1 2 3) implies that the value of X should be 1, 2, or 3. On the other hand, the CHOICE=(X) considers all distinctive nonmissing values of X as elements of the choice set.

CONVERGE=*number*
specifies the convergence criterion. The CONVERGE= option is the same as the ABSGCONV= option in the NLOPTIONS statement. The ABSGCONV= option in the NLOPTIONS statement overrides the CONVERGE= option. The default value is 1E–5.

HALTONSTART=*number*
specifies the starting point of the Halton sequence. The specified number must be a positive integer. The default is HALTONSTART=11.

HEV=(*option-list* **)**
specifies options that are used to estimate the HEV model. The HEV model with a unit scale for the alternative 1 is estimated using the following SAS statement:

```
model y = x1 x2 x3 / hev=(unitscale=1);
```

The following options can be used in the HEV= option. These options are listed within parentheses and separated by commas.

INTORDER=*number*
specifies the number of summation terms for Gaussian quadrature integration. The default is INTORDER=40. The maximum order is limited to 45. This option applies only to the INTEGRATION=LAGUERRE method.

UNITSCALE=*number-list*
specifies restrictions on scale parameters of stochastic utility components.

INTEGRATE=LAGUERRE | HARDY
specifies the integration method. The INTEGRATE=HARDY option specifies an adaptive integration method, while the INTEGRATE=LAGUERRE option specifies the Gauss-Laguerre approximation method. The default is INTEGRATE=LAGUERRE.

MIXED=(*option-list* **)**
specifies options that are used for mixed logit estimation. The mixed logit model with normally distributed random parameters is specified as follows:

```
model y = x1 x2 x3 / mixed=(normalparm=x1);
```

The following options can be used in the MIXED= option. The options are listed within parentheses and separated by commas.

LOGNORMALPARM=*variables*
specifies the variables whose random coefficients are lognormally distributed. LOGNORMALPARM= *variables* must be included on the right-hand side of the MODEL statement.

NORMALEC=*variables*
specifies the error component variables whose coefficients have a normal distribution $N(0, \sigma^2)$.

NORMALPARM=*variables*
specifies the variables whose random coefficients are normally distributed. NORMALPARM= *variables* must be included on the right-hand side of the MODEL statement.

UNIFORMEC=*variables*
specifies the error component variables whose coefficients have a uniform distribution $U(-\sqrt{3}\sigma, \sqrt{3}\sigma)$.

UNIFORMPARM=*variables*
specifies the variables whose random coefficients are uniformly distributed. UNIFORMPARM= *variables* must be included on the right-hand side of the MODEL statement.

NCHOICE=*number*
specifies the number of choices for multinomial choice models when all individuals have the same choice set. When individuals have different number of choices, the NCHOICE= option is not allowed, and the CHOICE= option should be used. The NCHOICE= and CHOICE= options must not be used simultaneously, and the NCHOICE= option cannot be used for nested logit models.

NSIMUL=*number*
specifies the number of simulations when the mixed logit or multinomial probit model is estimated. The default is NSIMUL=100. In general, you need a smaller number of simulations with RANDNUM=HALTON than with RANDNUM=PSEUDO.

RANDNUM=*value*
specifies the type of the random number generator used for simulation. RANDNUM=HALTON is the default. The following option values are allowed:

PSEUDO specifies pseudo-random number generation.

HALTON specifies Halton sequence generation.

RANDINIT
RANDINIT=*number*
specifies that initial parameter values be perturbed by uniform pseudo-random numbers for numerical optimization of the objective function. The default is $U(-1, 1)$. When the RANDINIT=r option is specified, $U(-r, r)$ pseudo-random numbers are generated. The value r should be positive. With a RANDINIT or RANDINIT= option, there are pure random searches for a given number of trials (1,000 for conditional or nested logit, and 500 for other models) to get a maximum (or minimum) value of the objective function. For example, when there is a parameter estimate with an initial value of 1, the RANDINIT option adds a generated random number u to the initial value and computes an objective function value by using $1 + u$. This option is helpful in finding the initial value automatically if there is no guidance in setting the initial estimate.

RANK
specifies that the dependent variable contain ranks. The numbers must be positive integers starting from 1. When the dependent variable has value 1, the corresponding alternative is chosen. This option is provided only as a convenience to the user; the extra information contained in the ranks is not currently used for estimation purposes.

RESTART=(*option-list* **)**
specifies options that are used for reiteration of the optimization problem. When the ADDRANDOM option is specified, the initial value of reiteration is computed using random grid searches around the initial solution, as follows:

```
model y = x1 x2 / type=clogit
   restart=(addvalue=(.01 .01));
```

The preceding SAS statement reestimates a conditional logit model by adding ADDVALUE= values. If the ADDVALUE= option contains missing values, the RESTART= option uses the corresponding estimate from the initial stage. If no ADDVALUE= value is specified for an estimate, a default value equal to (|estimate| * 1e-3) is added to the corresponding estimate from the initial stage. If both the ADDVALUE= and ADDRANDOM(=) options are specified, ADDVALUE= is ignored.

The following options can be used in the RESTART= option. The options are listed within parentheses.

ADDMAXIT=*number*
specifies the maximum number of iterations for the second stage of the estimation. The default is ADDMAXIT=100.

ADDRANDOM | ADDRANDOM=*value*
specifies random added values to the estimates from the initial stage. With the ADDRANDOM option, $U(-1, 1)$ random numbers are created and added to the estimates obtained in the initial stage. When the ADDRANDOM=r option is specified, $U(-r, r)$ random numbers are generated. The restart initial value is determined based on the given number of random searches (1,000 for conditional or nested logit, and 500 for other models).

ADDVALUE=(*value-list* **)**
specifies values added to the estimates from the initial stage. A missing value in the list is considered as a zero value for the corresponding estimate. When the ADDVALUE= option is not specified, default values equal to (|estimate| * 1e-3) are added.

SAMESCALE
specifies that the parameters of the inclusive values be the same within a group at each level when the nested logit is estimated.

SEED=*number*
specifies an initial seed for pseudo-random number generation. The SEED= value must be less than $2^{31} - 1$. If the SEED= value is negative or zero, the time of day from the computer's clock is used to obtain the initial seed. The default is SEED=0.

SPSCALE
specifies that the parameters of the inclusive values be the same for any choice with only one nested choice within a group, for each level in a nested logit model. This option is useful in analyzing stated preference data.

TYPE=*value*
specifies the type of model to be analyzed. The following model types are supported:

CONDITIONLOGIT | CLOGIT | CL	specifies a conditional logit model.
HEV	specifies a heteroscedastic extreme-value model.
MIXEDLOGIT | MXL	specifies a mixed logit model.
MULTINOMPROBIT | MPROBIT | MP	specifies a multinomial probit model.
NESTEDLOGIT | NLOGIT | NL	specifies a nested logit model.

UNITVARIANCE=(*number-list* **)**
specifies normalization restrictions on error variances of multinomial probit for the choices whose numbers are given in the list. If the UNITVARIANCE= option is specified, it must include at least two choices. Also, for identification, additional zero restrictions are placed on the correlation coefficients for the last choice in the list.

COVEST=*value*
specifies the type of covariance matrix. The following types are supported:

OP	specifies the covariance from the outer product matrix.
HESSIAN	specifies the covariance from the Hessian matrix.
QML	specifies the covariance from the outer product and Hessian matrices.

When COVEST=OP is specified, the outer product matrix is used to compute the covariance matrix of the parameter estimates. The COVEST=HESSIAN option produces the covariance matrix by using the inverse Hessian matrix. The quasi-maximum likelihood estimates are computed with COVEST=QML. The default is COVEST=HESSIAN when the Newton-Raphson method is used. COVEST=OP is the default when the OPTMETHOD=QN option is specified.

Printing Options

ALL
requests all printing options.

COVB
displays the estimated covariances of the parameter estimates.

CORRB
displays the estimated correlation matrix of the parameter estimates.

ITPRINT
displays the initial parameter estimates, convergence criteria, and constraints of the optimization. At each iteration, the objective function value, the maximum absolute gradient element, the step size, and the slope of search direction are printed. The objective function is the full negative log-likelihood function for the maximum likelihood method. When the ITPRINT option is specified and the NLOPTIONS statement is specified, all printing options in the NLOPTIONS statement are ignored.

NOPRINT
suppresses all displayed output.

Estimation Control Options

You can also specify detailed optimization options in the NLOPTIONS statement. The OPTMETHOD= option overrides the TECHNIQUE= option in the NLOPTIONS statement. The NLOPTIONS statement is ignored if the OPTMETHOD= option is specified.

INITIAL=(*initial-values* **)**

START=(*initial-values* **)**

specifies initial values for some or all of the parameter estimates. The values specified are assigned to model parameters in the same order in which the parameter estimates are displayed in the MDC procedure output.

When you use the INITIAL= option, the initial values in the INITIAL= option must satisfy the restrictions specified for the parameter estimates. If they do not, the initial values you specify are adjusted to satisfy the restrictions.

MAXITER=*number*

sets the maximum number of iterations allowed. The MAXITER= option overrides the MAXITER= option in the NLOPTIONS statement. The default is MAXITER=100.

OPTMETHOD=*value*

specifies the optimization technique when the estimation method uses nonlinear optimization. The following techniques are supported:

QN specifies the quasi-Newton method.

NR specifies the Newton-Raphson method.

TR specifies the trust region method.

The OPTMETHOD=NR option is the same as the TECHNIQUE=NEWRAP option in the NLOPTIONS statement. For the conditional and nested logit models, the default is OPTMETHOD=NR. For other models, the default is OPTMETHOD=QN.

NEST Statement

NEST LEVEL *(level-number)= (choices@choice, . . .)* **;**

The NEST statement is used when one choice variable contains all possible alternatives and the TYPE=NLOGIT option is specified. The decision tree is constructed based on the NEST statement. When the choice set is specified using multiple CHOICE= variables in the MODEL statement, the NEST statement is ignored.

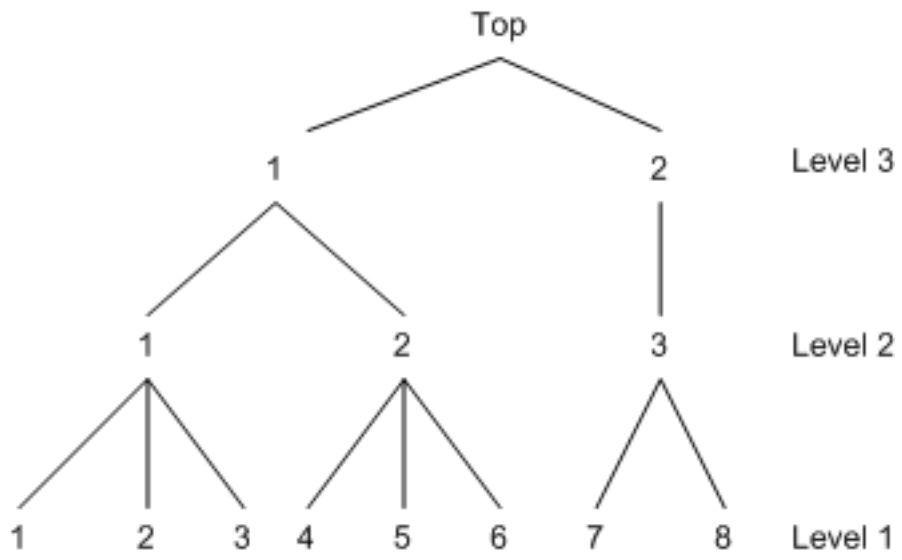
Consider the following eight choices that are nested in a three-level tree structure:

Level 1	Level 2	Level 3	top
1	1	1	1
2	1	1	1
3	1	1	1
4	2	1	1
5	2	1	1
6	2	1	1
7	3	2	1
8	3	2	1

You can use the following NEST statement to specify the tree structure displayed in Figure 23.22:

```
nest level(1) = (1 2 3 @ 1, 4 5 6 @ 2, 7 8 @ 3),
     level(2) = (1 2 @ 1, 3 @ 2),
     level(3) = (1 2 @ 1);
```

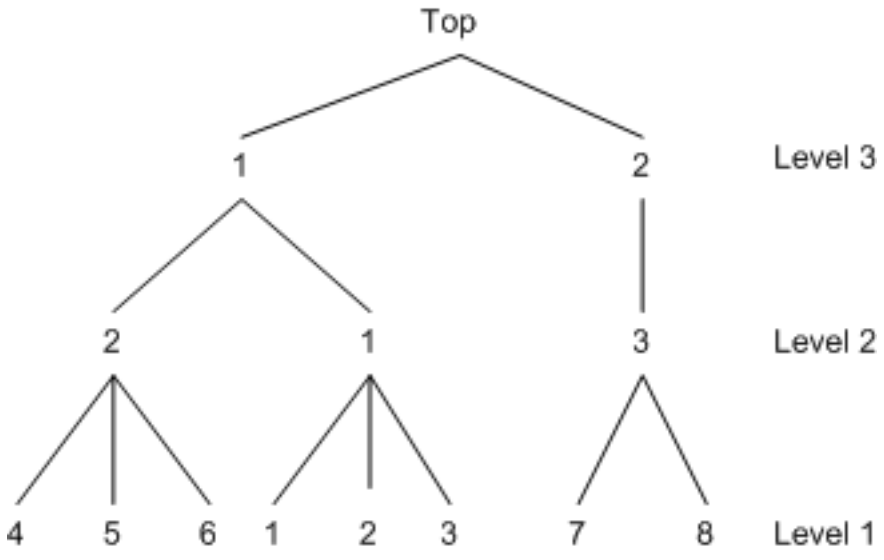
Figure 23.22 A Three-Level Tree



Note that the decision tree is constructed based on the sequence of first-level choice set specification. Therefore, specifying another order at Level 1 builds a different tree. The following NEST statement builds the tree displayed in Figure 23.23:

```
nest level(1) = (4 5 6 @ 2, 1 2 3 @ 1, 7 8 @ 3),
     level(2) = (1 2 @ 1, 3 @ 2),
     level(3) = (1 2 @ 1);
```

Figure 23.23 An Alternative Three-Level Tree



However, the NEST statement with a different sequence of choice specification at higher levels builds the same tree as displayed in Figure 23.22 if the sequence at the first level is the same:

```
nest level(1) = (1 2 3 @ 1, 4 5 6 @ 2, 7 8 @ 3),
     level(2) = (3 @ 2, 1 2 @ 1),
     level(3) = (1 2 @ 1);
```

The following specifications are equivalent:

```
nest level(2) = (3 @ 2, 1 2 @ 1)

nest level(2) = (3 @ 2, 1 @ 1, 2 @ 1)

nest level(2) = (1 @ 1, 2 @ 1, 3 @ 2)
```

Since the MDC procedure contains multiple cases for each individual, it is important to keep the data sequence in the proper order. Consider the four-choice multinomial model with one explanatory variable cost:

pid	choice	y	cost
1	1	1	10
1	2	0	25
1	3	0	20
1	4	0	30
2	1	0	15
2	2	0	22
2	3	1	16
2	4	0	25

The order of data needs to correspond to the value of choice. Therefore, the following data set is equivalent to the preceding data:

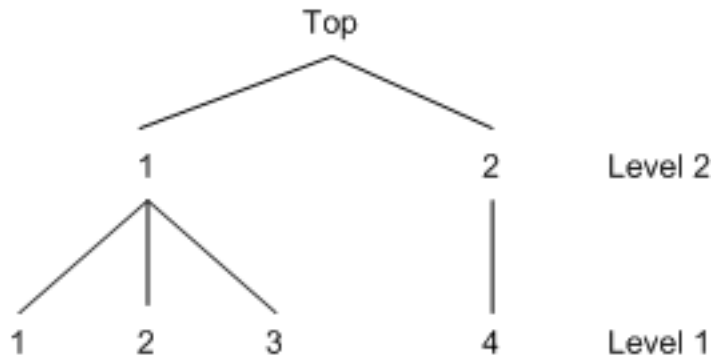
pid	choice	y	cost
1	2	0	25
1	3	0	20
1	1	1	10
1	4	0	30
2	3	1	16
2	4	0	25
2	1	0	15
2	2	0	22

The two-level nested model is estimated with a NEST statement, as follows:

```
proc mdc data=one type=nlogit;
   model y = cost / choice=(choice);
   id pid;
   utility u(1,) = cost;
   nest level(1) = (1 2 3 @ 1, 4 @ 2),
        level(2) = (1 2 @ 1);
run;
```

The tree is constructed as in Figure 23.24.

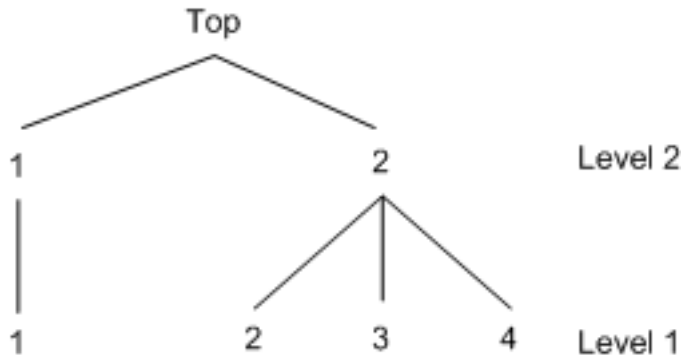
Figure 23.24 A Two-Level Tree



Another model is estimated if you specify the decision tree as in Figure 23.25. The different nested tree structure is specified in the following SAS statements:

```
proc mdc data=one type=nlogit;
   model y = cost / choice=(choice);
   id pid;
   utility u(1,) = cost;
   nest level(1) = (1 @ 1, 2 3 4 @ 2),
        level(2) = (1 2 @ 1);
run;
```

Figure 23.25 An Alternate Two-Level Tree



NLOPTIONS Statement

NLOPTIONS *options* **;**

PROC MDC uses the nonlinear optimization (NLO) subsystem to perform nonlinear optimization tasks. The NLOPTIONS statement specifies nonlinear optimization options. The NLOPTIONS statement must follow the MODEL statement. For a list of all the options of the NLOPTIONS statement, see Chapter 6, "Nonlinear Optimization Methods."

OUTPUT Statement

OUTPUT *options* **;**

The OUTPUT statement creates a new SAS data set that contains all the variables in the input data set and, optionally, the estimated linear predictors (XBETA) and predicted probabilities (P). The input data set must be sorted by the choice variables within each ID.

OUT=*SAS-data-set*
specifies the name of the output data set.

PRED=*variable name*

P=*variable name*
requests the predicted probabilities by naming the variable that contains the predicted probabilities in the output data set.

XBETA=*variable name*
names the variable that contains the linear predictor ($x'\boldsymbol{\beta}$) values. However, the XBETA= option is not supported in the nested logit model.

RESTRICT Statement

RESTRICT *restriction1* < *, restriction2 . . .* > **;**

The RESTRICT statement imposes linear restrictions on the parameter estimates. You can specify any number of RESTRICT statements.

Each *restriction* is written as an expression, followed by an equality operator (=) or an inequality operator (<, >, <=, >=), followed by a second expression:

expression operator expression **;**

The *operator* can be =, <, >, <= , or >=.

Restriction expressions can be composed of parameters; multiplication ($*$), summation ($+$), and subtraction ($-$) operators; and constants. Parameters named in restriction expressions must be among the parameters estimated by the model. Parameters associated with a regressor variable are referred to by the name of the corresponding regressor variable. The restriction expressions must be a linear function of the parameters.

Lagrange multipliers are reported for all the active linear constraints. In the displayed output, the Lagrange multiplier estimates are identified with the names Restrict1, Restrict2, and so on. The probability of the Lagrange multipliers is computed using a beta distribution (LaMotte 1994).

The following are examples of using the RESTRICT statement:

```
proc mdc data=one;
model y = x1-x10 /
          type=clogit
          choice=(mode 1 2 3);
id pid;
restrict x1*2 <= x2 + x3, ;
```

```
run;

proc mdc data=newdata;
model decision = ttime /
         type=mprobit
         nchoice=3
         unitvariance=(1 2)
         covest=hess;
id pid;
restrict RHO_31 = 0, STD_3<=1;
run;
```

TEST Statement

<'label':> **TEST** *<'string':> equation <,equation. . . > </ options>* **;**

The TEST statement performs Wald, Lagrange multiplier, and likelihood ratio tests of linear hypotheses about the regression parameters in the preceding MODEL statement. Each equation specifies a linear hypothesis to be tested. All hypotheses in one TEST statement are tested jointly. Variable names in the equations must correspond to regressors in the preceding MODEL statement, and each name represents the coefficient of the corresponding regressor. The keyword INTERCEPT refers to the coefficient of the intercept.

The following *options* can be specified after the slash (/):

ALL
requests Wald, Lagrange multiplier, and likelihood ratio tests.

WALD
requests the Wald test.

LM
requests the Lagrange multiplier test.

LR
requests the likelihood ratio test.

The following statements illustrate the use of the TEST statement:

```
proc mdc;
   model decision = x1 x2 / type=clogit
         choice=(mode 1 2 3);
   id pid;
   test x1 = 0, 0.5 * x1 + 2 * x2 = 0;
run;
```

The test investigates the joint hypothesis that

$$\beta_1 = 0$$

and

$$0.5\beta_1 + 2\beta_2 = 0$$

Only linear equality restrictions and tests are permitted in PROC MDC. Tests expressions can be composed only of algebraic operations that use the addition symbol (+), subtraction symbol (–), and multiplication symbol (*).

The TEST statement accepts labels that are reproduced in the printed output. The TEST statement can be labeled in two ways. A TEST statement can be preceded by a label followed by a colon. Alternatively, the keyword TEST can be followed by a quoted string followed by a colon. If both are present, PROC MDC uses the label that precedes the first colon. If no label is present, PROC MDC automatically labels the tests.

UTILITY Statement

UTILITY U *(level < , choices >)= variables* **;**

The UTILITY statement specifies a utility function that can be used in estimating a nested logit model. The U()= option can have two arguments. The first argument contains level information, and the second argument is related to choice information. The second argument can be omitted for the first level when all the choices at the first level share the same variables and the same parameters. However, for any level above the first, the second argument must be provided. The UTILITY statement specifies a utility function while the NEST statement constructs the decision tree.

Consider a two-level nested logit model that has one explanatory variable at level 1. This model can be specified as follows:

```
proc mdc data=one type=nlogit;
   model y = cost / choice=(choice);
   id pid;
   utility u(1,2 3 4) = cost;
   nest level(1) = (1 @ 1, 2 3 4 @ 2),
        level(2) = (1 2 @ 1);
run;
```

You also can specify the following statement because all the variables at the first level share the same explanatory variable, cost:

```
utility u(1,) = cost;
```

The variable, cost, must be listed in the MODEL statement. When the additional explanatory variable, dummy, is included at level 2, another U()= option needs to be specified. Note that the U()= option must specify choices within any level above the first. Thus, it is specified as U(2, 1 2) in the following statements:

```
proc mdc data=one type=nlogit;
   model y = cost dummy / choice=(choice);
   id pid;
   utility u(1,) = cost,
           u(2,1 2) = dummy;
   nest level(1) = (1 @ 1, 2 3 4 @ 2),
        level(2) = (1 2 @ 1);
run;
```

Details: MDC Procedure

Multinomial Discrete Choice Modeling

When the dependent variable takes multiple discrete values, you can use multinomial discrete choice modeling to analyze the data. This section considers models for unordered multinomial data.

Let the random utility function be defined by

$$U_{ij} = V_{ij} + \epsilon_{ij}$$

where the subscript i is an index for the individual, the subscript j is an index for the alternative, V_{ij} is a nonstochastic utility function, and ϵ_{ij} is a random component (error) that captures unobserved characteristics of alternatives or individuals or both. In multinomial discrete choice models, the utility function is assumed to be linear, so that $V_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta}$.

In the conditional logit model, each ϵ_{ij} for all $j \in C_i$ is distributed independently and identically (iid) with the Type I extreme-value distribution, $\exp(-\exp(-\epsilon_{ij}))$, also known as the Gumbel distribution.

The iid assumption on the random components of the utilities of the different alternatives can be relaxed to overcome the well-known and restrictive *independence from irrelevant alternatives* (IIA) property of the conditional logit model. This allows for more flexible substitution patterns among alternatives than the one imposed by the conditional logit model. (See the section "Independence from Irrelevant Alternatives (IIA)" on page 1381.)

The nested logit model is derived by allowing the random components to be identical but nonindependent. Instead of independent Type I extreme-value errors, the errors are assumed to have a generalized extreme-value distribution. This model generalizes the conditional logit model to allow for particular patterns of correlation in unobserved utility (McFadden 1978).

Another generalization of the conditional logit model, the heteroscedastic extreme-value (HEV) model, is obtained by allowing independent but nonidentical errors distributed with a Type I extreme-value distribution (Bhat 1995). It permits different variances on the random components of utility across the alternatives.

Mixed logit models are also generalizations of the conditional logit model that can represent very general patterns of substitution among alternatives. For more information, see the section "Mixed Logit Model" on page 1382.

The multinomial probit (MNP) model is derived when the errors, $(\epsilon_{i1}, \epsilon_{i2}, \ldots, \epsilon_{iJ})$, have a multivariate normal (MVN) distribution. Thus, this model accommodates a very general error structure.

The multinomial probit model requires burdensome computation compared to a family of multinomial choice models derived from the Gumbel distributed utility function, since it involves multi-dimensional integration (with dimension $J - 1$) in the estimation process. In addition, the multinomial probit model requires more parameters than other multinomial choice models. As a result, conditional and nested logit models are used more frequently, even though they are derived from a utility function whose random component is more restrictively defined than the multinomial probit model.

The event of a choice being made, $\{y_i = j\}$, can be expressed using a random utility function

$$U_{ij} \geq \max_{k \in C_i, k \neq j} U_{ik}$$

where C_i is the choice set of individual i. Individual i chooses alternative j if and only if it provides a level of utility that is greater than or equal to that of any other alternative in his choice set. Then, the probability that individual i chooses alternative j (from among the n_i choices in his choice set C_i) is

$$P_i(j) = P_{ij} = P[\mathbf{x}'_{ij}\boldsymbol{\beta} + \epsilon_{ij} \geq \mathbf{max}_{k \in C_i}(\mathbf{x}'_{ik}\boldsymbol{\beta} + \epsilon_{ik})]$$

Multinomial Logit and Conditional Logit

When explanatory variables contain only individual characteristics, the multinomial logit model is defined as

$$P(y_i = j) = P_{ij} = \frac{\exp(\mathbf{x}'_i\boldsymbol{\beta}_j)}{\sum_{k=0}^{J}\exp(\mathbf{x}'_i\boldsymbol{\beta}_k)} \quad \text{for}\, j = 0, \ldots, J$$

where y_i is a random variable that indicates the choice made, x_i is a vector of characteristics specific to the ith individual, and $\boldsymbol{\beta}_j$ is a vector of coefficients specific to the jth alternative. Thus, this model involves choice-specific coefficients and only individual specific regressors. For model identification, it is often assumed that $\boldsymbol{\beta}_0 = 0$. The multinomial logit model reduces to the binary logit model if $J = 1$.

The ratio of the choice probabilities for alternatives j and l (the *odds ratio* of alternatives j and l) is

$$\frac{P_{ij}}{P_{il}} = \frac{\exp(\mathbf{x}'_i\boldsymbol{\beta}_j)/\sum_{k=0}^{J}\exp(\mathbf{x}'_i\boldsymbol{\beta}_k)}{\exp(\mathbf{x}'_i\boldsymbol{\beta}_l)/\sum_{k=0}^{J}\exp(\mathbf{x}'_i\boldsymbol{\beta}_k)} = \exp[\mathbf{x}'_i(\boldsymbol{\beta}_j - \boldsymbol{\beta}_l)]$$

Note that the odds ratio of alternatives j and l does not depend on any alternatives other than j and l. For more information, see the section "Independence from Irrelevant Alternatives (IIA)" on page 1381.

The log-likelihood function of the multinomial logit model is

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{j=0}^{J} d_{ij} \ln P(y_i = j)$$

where

$$d_{ij} = \begin{cases} 1 & \text{if individual } i \text{ chooses alternative } j \\ 0 & \text{otherwise} \end{cases}$$

This type of multinomial choice modeling has a couple of weaknesses: it has too many parameters (the number of individual characteristics times J), and it is difficult to interpret. The multinomial logit model can be used to predict the choice probabilities, among a given set of $J + 1$ alternatives, of an individual with known vector of characteristics $\mathbf{x}_i$.

The parameters of the multinomial logit model can be estimated with the TYPE=CLOGIT option in the MODEL statement; however, this requires modification of the conditional logit model to allow individual specific effects.

The conditional logit model, sometimes called the multinomial logit model, is similarly defined when choice-specific data are available. Using properties of Type I extreme-value (Gumbel) distribution, the probability that individual i chooses alternative j from among the choices in his choice set C_i is

$$P(y_i = j) = P_{ij} = P[\mathbf{x}'_{ij}\boldsymbol{\beta} + \epsilon_{ij} \geq \max_{k \in C_i, k \neq j}(\mathbf{x}'_{ik}\boldsymbol{\beta} + \epsilon_{ik})] = \frac{\exp(\mathbf{x}'_{ij}\boldsymbol{\beta})}{\sum_{k \in C_i}\exp(\mathbf{x}'_{ik}\boldsymbol{\beta})}$$

where $\mathbf{x}_{ij}$ is a vector of attributes specific to the *j*th alternative as perceived by the *i*th individual. It is assumed that there are n_i choices in each individual's choice set, C_i.

The log-likelihood function of the conditional logit model is

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{j \in C_i} d_{ij} \ln P(y_i = j)$$

The conditional logit model can be used to predict the probability that an individual will choose a previously unavailable alternative, given knowledge of $\boldsymbol{\beta}$ and the vector $\mathbf{x}_{ij}$ of choice-specific characteristics.

Independence from Irrelevant Alternatives (IIA)

The problematic aspect of the conditional logit (and the multinomial logit) model lies in the property of independence from irrelevant alternatives (IIA). The IIA property can be derived from the probability ratio of any two choices. For the conditional logit model,

$$\frac{P_{ij}}{P_{il}} = \frac{\exp(\mathbf{x}'_{ij}\boldsymbol{\beta}) / \sum_{k \in C_i} \exp(\mathbf{x}'_{ik}\boldsymbol{\beta})}{\exp(\mathbf{x}'_{il}\boldsymbol{\beta}) / \sum_{k \in C_i} \exp(\mathbf{x}'_{ik}\boldsymbol{\beta})} = \exp[(\mathbf{x}_{ij} - \mathbf{x}_{il})'\boldsymbol{\beta}]$$

It is evident that the ratio of the probabilities for alternatives *j* and *l* does not depend on any alternatives other than *j* and *l*. This was also shown to be the case for the multinomial logit model. Thus, for the conditional and multinomial logit models, the ratio of probabilities of any two alternatives is necessarily the same regardless of what other alternatives are in the choice set or what the characteristics of the other alternatives are. This is referred to as the IIA property.

The IIA property is useful from the point of view of estimation and forecasting. For example, it allows the prediction of demand for currently unavailable alternatives. If the IIA property is appropriate for the choice situation being considered, then estimation can be based on the set of currently available alternatives, and then the estimated model can be used to calculate the probability that an individual would choose a new alternative not considered in the estimation procedure. However, the IIA property is restrictive from the point of view of choice behavior. Models that display the IIA property predict that a change in the attributes of one alternative changes the probabilities of the other alternatives proportionately such that the ratios of probabilities remain constant. Thus, cross elasticities due to a change in the attributes of an alternative *j* are equal for all alternatives $k \neq j$. This particular substitution pattern might be too restrictive in some choice settings.

The IIA property of the conditional logit model follows from the assumption that the random components of utility are identically and independently distributed. The other models in PROC MDC (namely, nested logit, HEV, mixed logit, and multinomial probit) relax the IIA property in different ways.

For an example of Hausman's specification test of IIA assumption, see "Example 23.6: Hausman's Specification Test" on page 1413.

Heteroscedastic Extreme-Value Model

The heteroscedastic extreme-value (HEV) model (Bhat 1995) allows the random components of the utility function to be nonidentical. Specifically, the HEV model assumes independent but nonidentical error terms distributed with the Type I extreme-value distribution. The HEV model allows the variances of the random

components of utility to differ across alternatives. Bhat (1995) argues that the HEV model does not have the IIA property. The HEV model contains the conditional logit model as a special case. The probability that an individual i will choose alternative j from the set C_i of available alternatives is

$$P_i(j) = \int_{-\infty}^{\infty} \prod_{k \in C_i, k \neq j} \Gamma\left[\frac{\mathbf{x}'_{ij}\boldsymbol{\beta} - \mathbf{x}'_{ik}\boldsymbol{\beta} + \theta_j w}{\theta_k}\right] \gamma(w) dw$$

where the choice set C_i has n_i elements and

$$\Gamma(x) = \exp(-\exp(-x))$$

$$\gamma(x) = \exp(-x)\Gamma(x)$$

are the cumulative distribution function and probability density function of the Type I extreme-value distribution. The variance of the error term for the jth alternative is $\frac{1}{6}\pi^2\theta_j^2$. If the scale parameters, θ_j, of the random components of utility of all alternatives are equal, then this choice probability is the same as that of the conditional logit model. The log-likelihood function of the HEV model can be written as

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{j \in C_i} d_{ij} \ln[P_i(j)]$$

where

$$d_{ij} = \begin{cases} 1 & \text{if individual } i \text{ chooses alternative } j \\ 0 & \text{otherwise} \end{cases}$$

Since the log-likelihood function contains an improper integral function, it is computationally difficult to get a stable estimate. With the transformation $u = \exp(-w)$, the probability can be written

$$\begin{aligned} P_i(j) &= \int_0^{\infty} \Pi_{k \in C_i, k \neq j} \Gamma\left[\frac{\mathbf{x}'_{ij}\boldsymbol{\beta} - \mathbf{x}'_{ik}\boldsymbol{\beta} - \theta_j \ln(u)}{\theta_k}\right] \exp(-u) du \\ &= \int_0^{\infty} G_{ij}(u) \exp(-u) du \end{aligned}$$

Using the Gauss-Laguerre weight function, $W(x) = \exp(-u)$, the integration of the log-likelihood function can be replaced with a summation as follows:

$$\int_0^{\infty} G_{ij}(u) \exp(-u) du = \sum_{k=1}^{K} w_k G_{ij}(x_k)$$

Weights (w_k) and abscissas (x_k) are tabulated by Abramowitz and Stegun (1970).

Mixed Logit Model

In mixed logit models, an individual's utility from any alternative can be decomposed into a deterministic component, $\mathbf{x}'_{ij}\boldsymbol{\beta}$, which is a linear combination of observed variables, and a stochastic component, $\xi_{ij} + \epsilon_{ij}$,

$$U_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \xi_{ij} + \epsilon_{ij}$$

where $\mathbf{x}_{ij}$ is a vector of observed variables that relate to individual i and alternative j, $\boldsymbol{\beta}$ is a vector of parameters, ξ_{ij} is an error component that can be correlated among alternatives and heteroscedastic for each individual, and ϵ_{ij} is a random term with zero mean that is independently and identically distributed over alternatives and individuals. The conditional logit model is derived if you assume ϵ_{ij} has an iid Gumbel distribution and $V(\xi_{ij}) = 0$.

The mixed logit model assumes a general distribution for ξ_{ij} and an iid Gumbel distribution for ϵ_{ij}. Denote the density function of the error component ξ_{ij} as $f(\xi_{ij}|\boldsymbol{\gamma})$, where $\boldsymbol{\gamma}$ is a parameter vector of the distribution of ξ_{ij}. The choice probability of alternative j for individual i is written as

$$P_i(j) = \int Q_i(j|\xi_{ij}) f(\xi_{ij}|\boldsymbol{\gamma}) d\xi_{ij}$$

where the conditional choice probability for a given value of ξ_{ij} is the logit

$$Q_i(j|\xi_{ij}) = \frac{\exp(\mathbf{x}'_{ij}\boldsymbol{\beta} + \xi_{ij})}{\sum_{k \in C_i} \exp(\mathbf{x}'_{ik}\boldsymbol{\beta} + \xi_{ik})}$$

Since ξ_{ij} is not given, the unconditional choice probability, $P_i(j)$, is the integral of the conditional choice probability, $Q_i(j|\xi_{ij})$, over the distribution of ξ_{ij}. This model is called "mixed logit" since the choice probability is a mixture of logits with $f(\xi_{ij}|\boldsymbol{\gamma})$ as the mixing distribution.

In general, the mixed logit model does not have an exact likelihood function because the probability $P_i(j)$ does not always have a closed form solution. Therefore, a simulation method is used for computing the approximate probability,

$$\tilde{P}_i(j) = 1/S \sum_{s=1}^{S} \tilde{Q}_i(j|\xi_{ij}^s)$$

where S is the number of simulation replications and $\tilde{P}_i(j)$ is a simulated probability. The simulated log-likelihood function is computed as

$$\tilde{\mathcal{L}} = \sum_{i=1}^{N} \sum_{j=1}^{n_i} d_{ij} \ln(\tilde{P}_i(j))$$

where

$$d_{ij} = \begin{cases} 1 & \text{if individual } i \text{ chooses alternative } j \\ 0 & \text{otherwise} \end{cases}$$

For simulation purposes, assume that the error component has a specific structure,

$$\xi_{ij} = \mathbf{z}'_{ij}\boldsymbol{\mu} + \mathbf{w}'_{ij}\boldsymbol{\beta}^*$$

where $\mathbf{z}_{ij}$ is a vector of observed data and $\boldsymbol{\mu}$ is a random vector with zero mean and density function $\psi(\boldsymbol{\mu}|\boldsymbol{\gamma})$. The observed data vector ($\mathbf{z}_{ij}$) of the error component can contain some or all elements of $\mathbf{x}_{ij}$. The component $\mathbf{z}'_{ij}\boldsymbol{\mu}$ induces heteroscedasticity and correlation across unobserved utility components of the alternatives. This allows flexible substitution patterns among the alternatives. The kth element of vector $\boldsymbol{\mu}$ is distributed as

$$\mu_k \sim (0, \sigma_k^2)$$

Therefore, μ_k can be specified as

$$\mu_k = \sigma_k \epsilon_\mu$$

where

$$\epsilon_\mu \sim N(0, 1)$$

or

$$\epsilon_\mu \sim U(-\sqrt{3}, \sqrt{3})$$

In addition, $\boldsymbol{\beta}^*$ is a vector of random parameters (random coefficients). Random coefficients allow heterogeneity across individuals in their sensitivity to observed exogenous variables. The observed data vector, $\mathbf{w}_{ij}$, is a subset of $\mathbf{x}_{ij}$. The following three types of distributions for the random coefficients are supported, where the mth element of $\boldsymbol{\beta}^*$ is denoted as β^*_m:

- Normally distributed coefficient with the mean b_m and spread s_m being estimated.

  $$\beta^*_m = b_m + s_m \epsilon_\beta \quad \text{and} \quad \epsilon_\beta \sim N(0, 1)$$

- Uniformly distributed coefficient with the mean b_m and spread s_m being estimated. A uniform distribution with mean b and spread s is $U(b - s, b + s)$.

  $$\beta^*_m = b_m + s_m \epsilon_\beta \quad \text{and} \quad \epsilon_\beta \sim U(-1, 1)$$

- Lognormally distributed coefficient. The coefficient is calculated as

  $$\beta^*_m = \exp(b_m + s_m \epsilon_\beta) \quad \text{and} \quad \epsilon_\beta \sim N(0, 1)$$

  where b_m and s_m are parameters that are estimated.

The estimate of spread for normally, uniformly, and lognormally distributed coefficients can be negative. The absolute value of the estimated spread can be interpreted as an estimate of standard deviation for normally distributed coefficients.

A detailed description of mixed logit models can be found, for example, in Brownstone and Train (1999).

Multinomial Probit

The multinomial probit model allows the random components of the utility of the different alternatives to be nonindependent and nonidentical. Thus, it does not have the IIA property. The increase in the flexibility of the error structure comes at the expense of introducing several additional parameters in the covariance matrix of the errors.

Consider the random utility function

$$U_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \epsilon_{ij}$$

where the joint distribution of $(\epsilon_{i1}, \epsilon_{i2}, \ldots, \epsilon_{iJ})$ is multivariate normal:

$$\begin{bmatrix} \epsilon_{i1} \\ \epsilon_{i2} \\ \vdots \\ \epsilon_{iJ} \end{bmatrix} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$$

$$\boldsymbol{\Sigma} = \left[\sigma_{jk}\right]_{j,k=1,\ldots,J}$$

The dimension of the error covariance matrix is determined by the number of alternatives J. Given $(\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{iJ})$, the jth alternative is chosen if and only if $U_{ij} \geq U_{ik}$ for all $k \neq j$. Thus, the probability that the jth alternative is chosen is

$$P(y_i = j) = P_{ij} = P[\epsilon_{i1} - \epsilon_{ij} < (\mathbf{x}_{ij} - \mathbf{x}_{i1})'\boldsymbol{\beta}, \ldots, \epsilon_{iJ} - \epsilon_{ij} < (\mathbf{x}_{ij} - \mathbf{x}_{iJ})'\boldsymbol{\beta}]$$

where y_i is a random variable that indicates the choice made. This is a cumulative probability from a $(J-1)$-variate normal distribution. Since evaluation of this probability involves multidimensional integration, it is practical to use a simulation method to estimate the model. Many studies have shown that the simulators proposed by the following authors (henceforth referred to as GHK) perform well: Geweke (1989); Hajivassiliou (1993); Keane (1994). For example, Hajivassiliou, McFadden, and Ruud (1996) compare 13 simulators using 11 different simulation methods and conclude that the GHK simulation method is the most reliable. To compute the probability of the multivariate normal distribution, the recursive simulation method is used. For more information about GHK simulators, see Hajivassiliou (1993).

The log-likelihood function for the multinomial probit model can be written as

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{j=1}^{J} d_{ij} \ln P(y_i = j)$$

where

$$d_{ij} = \begin{cases} 1 & \text{if individual } i \text{ chooses alternative } j \\ 0 & \text{otherwise} \end{cases}$$

For identification of the multinomial probit model, two of the diagonal elements of Σ are normalized to 1, and it is assumed that for one of the choices whose error variance is normalized to 1 (say, k), it is also true that $\sigma_{jk} = \sigma_{kj} = 0$ for $j = 1, \ldots, J$ and $j \neq k$. Thus, a model with J alternatives has at most $J(J-1)/2 - 1$ covariance parameters after normalization.

Let D and R be defined as

$$D = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_J \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1J} \\ \rho_{21} & 1 & \cdots & \rho_{2J} \\ \vdots & \vdots & \vdots & \vdots \\ \rho_{J1} & \rho_{J2} & \cdots & 1 \end{bmatrix}$$

where $\sigma_j^2 = \sigma_{jj}$ and $\rho_{jk} = \frac{\sigma_{jk}}{\sigma_j \sigma_k}$. Then, for identification, $\sigma_{J-1} = \sigma_J = 1$ and $\rho_{kJ} = \rho_{Jk} = 0$, for all $k \neq J$ can be imposed, and the error covariance matrix is $\mathbf{\Sigma} = DRD$.

In the standard MDC output, the parameter estimates STD_j and RHO_jk correspond to σ_j and ρ_{jk}.

In principle, the multinomial probit model is fully identified with the preceding normalizations. However, in practice, convergence in applications of the model with more than three alternatives often requires additional restrictions on the elements of $\mathbf{\Sigma}$.

It must also be noted that the unrestricted structure of the error covariance matrix makes it impossible to forecast demand for a new alternative without knowledge of the new $(J + 1)$ by $(J + 1)$ error covariance matrix.

Nested Logit

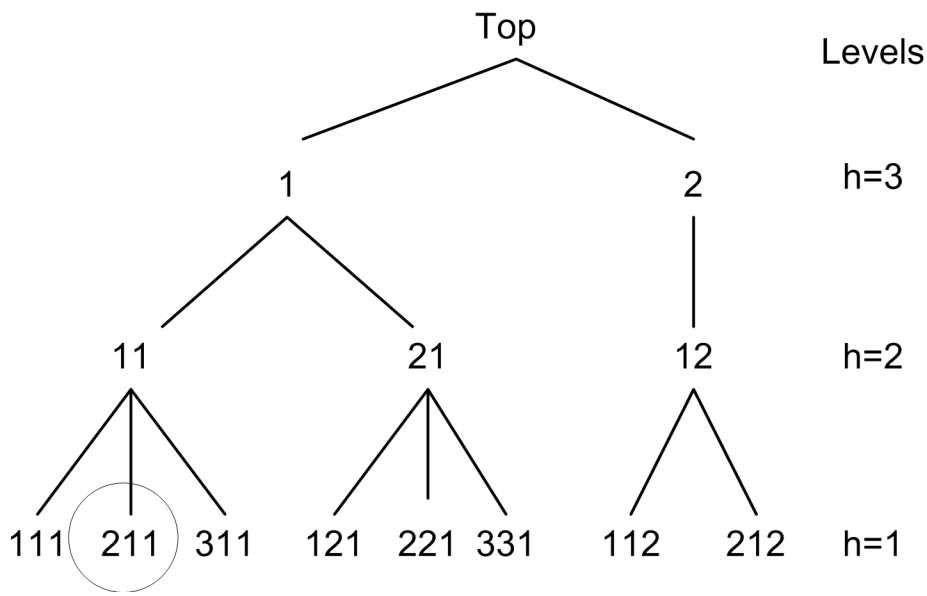
The nested logit model (McFadden 1978, 1981) allows partial relaxation of the assumption of independence of the stochastic components of utility of alternatives. In some choice situations, the IIA property holds for some pairs of alternatives but not all. In these situations, the nested logit model can be used if the set of alternatives faced by an individual can be partitioned into subsets such that the IIA property holds within subsets but not across subsets.

In the nested logit model, the joint distribution of the errors is generalized extreme value (GEV). This is a generalization of the Type I extreme-value distribution that gives rise to the conditional logit model. Note that all ϵ_{ij} within each subset are correlated with each other. For more information, see McFadden (1978, 1981).

Nested logit models can be described analytically following the notation of McFadden (1981). Assume that there are L levels, with 1 representing the lowest and L representing the highest level of the tree. The index of a node at level h in the tree is a pair (j_h, π_h), where $\pi_h = (j_{h+1}, \ldots, j_L)$ is the index of the adjacent node at level $h + 1$. Thus, the primitive alternatives, at level 1 in the tree, are indexed by vectors $(j_1, \ldots, j_L)$, and the alternative nodes at level L are indexed by integers j_L. The choice set C_{π_h} is the set of primitive alternatives (at level 1) that belong to branches below the node π_h. The notation C_{π_h} is also used to denote a set of indices j_h such that (j_h, π_h) is a node immediately below π_h. Note that C_{π_0} is a set with a single element, while C_{π_L} represents a choice set that contains all possible alternatives. As an example, consider the circled node at level 1 in Figure 23.26. Since it stems from node 11, $\pi_h = 11$, and since it is the second node stemming from 11, $j_h = 2$, its index is $\pi_{h-1} = \pi_0 = (j_h, \pi_h) = 211$. Similarly, $C_{11} = \{111, 211, 311\}$ contains all the possible choices below 11.

Although this notation is useful for writing closed-form solutions for probabilities, the MDC procedure allows a more flexible definition of indices. For more information about how to describe decision trees within the MDC procedure, see the section "NEST Statement" on page 1372.

Figure 23.26 Node Indices for a Three-Level Tree



Let $\mathbf{x}^{(h)}_{i;j_h\pi_h}$ denote the vector of observed variables for individual i common to the alternatives below node $j_h\pi_h$. The probability of choice at level h has a closed-form solution and is written

$$P_i(j_h|\pi_h) = \frac{\exp\left[\mathbf{x}^{(h)\prime}_{i;j_h\pi_h}\boldsymbol{\beta}^{(h)} + \sum_{k\in C_{i;j_h\pi_h}} I_{k,j_h\pi_h}\theta_{k,j_h\pi_h}\right]}{\sum_{j\in C_{i;\pi_h}} \exp\left[\mathbf{x}^{(h)\prime}_{i;j\pi_h}\boldsymbol{\beta}^{(h)} + \sum_{k\in C_{i;j\pi_h}} I_{k,j\pi_h}\theta_{k,j\pi_h}\right]}, h = 2, \ldots, L$$

where I_{π_h} is the *inclusive value* (at level $h+1$) of the branch below node π_h and is defined recursively as follows:

$$I_{\pi_h} = \ln\left\{\sum_{j\in C_{i;\pi_h}} \exp\left[\mathbf{x}^{(h)\prime}_{i;j\pi_h}\boldsymbol{\beta}^{(h)} + \sum_{k\in C_{i;j\pi_h}} I_{k,j\pi_h}\theta_{k,j\pi_h}\right]\right\}$$

$$0 \le \theta_{k,\pi_1} \le \cdots \le \theta_{k,\pi_{L-1}}$$

The inclusive value I_{π_h} denotes the average utility that the individual can expect from the branch below π_h. The *dissimilarity parameters* or *inclusive value parameters* ($\theta_{k,j\pi_h}$) are the coefficients of the inclusive values and have values between 0 and 1 if nested logit is the correct model specification. When they all take value 1, the nested logit model is equivalent to the conditional logit model.

At decision level 1, there is no inclusive value; that is, $I_{\pi_0} = 0$. Therefore, the conditional probability is

$$P_i(j_1|\pi_1) = \frac{\exp\left[\mathbf{x}^{(1)\prime}_{i;j_1\pi_1}\boldsymbol{\beta}^{(1)}\right]}{\sum_{j\in C_{i;\pi_1}} \exp\left[\mathbf{x}^{(1)\prime}_{i;j\pi_1}\boldsymbol{\beta}^{(1)}\right]}$$

The log-likelihood function at level h can then be written

$$\mathcal{L}^{(h)} = \sum_{i=1}^{N} \sum_{\pi_{h'}\in C_{i,\pi_{h+1}}} \sum_{j\in C_{i,\pi_{h'}}} y_{i,j\pi_{h'}} \ln P(C_{i,j\pi_{h'}}|C_{i,\pi_{h'}})$$

where $y_{i,j\pi_{h'}}$ is an indicator variable that has the value of 1 for the selected choice. The full log-likelihood function of the nested logit model is obtained by adding the conditional log-likelihood functions at each level:

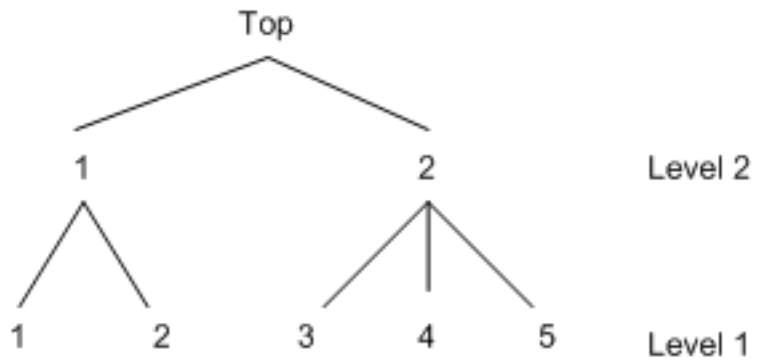
$$\mathcal{L} = \sum_{h=1}^{L} \mathcal{L}^{(h)}$$

Note that the log-likelihood functions are computed from conditional probabilities when $h < L$. The nested logit model is estimated using the full information maximum likelihood method.

Decision Tree and Nested Logit

You can view choices as a decision tree and model the decision tree by using the nested logit model. You need to use either the NEST statement or the CHOICE= option of the MODEL statement to specify the nested tree structure. Additionally, you need to identify which explanatory variables are used at each level of the decision tree. These explanatory variables are arguments for what is called a *utility function*. The utility function is specified using UTILITY statements. For example, consider a two-level decision tree. The tree structure is displayed in Figure 23.27.

Figure 23.27 Two-Level Decision Tree



A nested logit model with two levels can be specified using the following SAS statements:

```
proc mdc data=one type=nlogit;
   model decision = x1 x2 x3 x4 x5 /
         choice=(upmode 1 2, mode 1 2 3 4 5);
   id pid;
   utility u(1, 3 4 5 @ 2) = x1 x2,
           u(1, 1 2 @ 1) = x3 x4,
           u(2, 1 2) = x5;
run;
```

The DATA=one data set should be arranged as follows:

obs	pid	upmode	mode	x1	x2	x3	x4	x5	decision
1	1	1	1	#	#	#	#	#	1
2	1	1	2	#	#	#	#	#	0
3	1	2	3	#	#	#	#	#	0
4	1	2	4	#	#	#	#	#	0
5	1	2	5	#	#	#	#	#	0
6	2	1	1	#	#	#	#	#	0
7	2	1	2	#	#	#	#	#	0
8	2	2	3	#	#	#	#	#	0
9	2	2	4	#	#	#	#	#	0
10	2	2	5	#	#	#	#	#	1

All model variables, x1 through x5, are specified in the UTILITY statement. It is required that entries denoted as # have values for model estimation and prediction. The values of the level 2 utility variable x5 should be the same for all the primitive (level 1) alternatives below node 1 at level 2 and, similarly, for all the primitive alternatives below node 2 at level 2. In other words, x5 should have the same value for primitive alternatives 1 and 2 and, similarly, it should have the same value for primitive alternatives 3, 4, and 5. More generally, the values of any level 2 or higher utility function variables should be constant across primitive alternatives under each node for which the utility function applies. Since PROC MDC expects this to be the case, it uses the values of x5 only for the primitive alternatives 1 and 3, ignoring the values for the primitive alternatives 2, 4, and 5. Thus, PROC MDC uses the values of the utility function variable only for the primitive alternatives that come first under each node for which the utility function applies. This behavior applies to any utility function variables that are specified above the first level. The choice variable for level 2 (upmode) should be placed before the first-level choice variable (mode) when the CHOICE= option is specified. Alternatively, the NEST statement can be used to specify the decision tree. The following SAS statements fit the same nested logit model:

```
proc mdc data=a type=nlogit;
   model decision = x1 x2 x3 x4 x5 /
            choice=(mode 1 2 3 4 5);
   id pid;
   utility u(1, 3 4 5 @ 2) = x1 x2,
           u(1, 1 2 @ 1) = x3 x4,
           u(2, 1 2) = x5;
   nest level(1) = (1 2 @ 1, 3 4 5 @ 2),
        level(2) = (1 2 @ 1);
run;
```

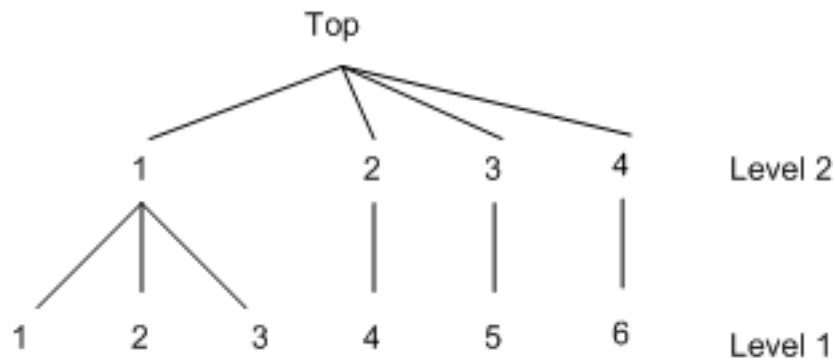
The U(1, 3 4 5 @ 2)= option specifies three choices, 3, 4, and 5, at level 1 of the decision tree. They are connected to the upper branch 2. The specified variables (x1 and x2) are used to model this utility function. The bottom level of the decision tree is level 1. All variables in the UTILITY statement must be included in the MODEL statement. When all choices at the first level share the same variables, you can omit the second argument of the U()= option for that level. However, U(1,) = x1 x2 is not equivalent to the following statements:

```
u(1, 3 4 5 @ 2) = x1 x2;
u(1, 1 2 @ 1) = x1 x2;
```

The CHOICE= variables need to be specified from the top to the bottom level. To forecast demand for new products, stated preference data are widely used. Stated preference data are attractive for market researchers because attribute variations can be controlled. Hensher (1993) explores the advantage of combining revealed

preference (market data) and stated preference data. The scale factor (V_{rp}/V_{sp}) can be estimated using the nested logit model with the decision tree structure displayed in Figure 23.28.

Figure 23.28 Decision Tree for Revealed and Stated Preference Data



Example SAS statements are as follows:

```
proc mdc data=a type=nlogit;
   model decision = x1 x2 x3 /
            spscale
            choice=(mode 1 2 3 4 5 6);
   id pid;
   utility u(1,) = x1 x2 x3;
   nest level(1) = (1 2 3 @ 1, 4 @ 2, 5 @ 3, 6 @ 4),
        level(2) = (1 2 3 4 @ 1);
run;
```

The SPSCALE option specifies that parameters of inclusive values for nodes 2, 3, and 4 at level 2 be the same. When you specify the SAMESCALE option, the MDC procedure imposes the same coefficient of inclusive values for choices 1–4.

Model Fit and Goodness-of-Fit Statistics

McFadden (1974) suggests a likelihood ratio index that is analogous to the R-square in the linear regression model,

$$R_M^2 = 1 - \frac{\ln L}{\ln L_0}$$

where L is the maximum of the log-likelihood function and L_0 is the maximum of the log-likelihood function when all coefficients, except for an intercept term, are zero. McFadden's likelihood ratio index is bounded by 0 and 1.

Estrella (1998) proposes the following requirements for a goodness-of-fit measure to be desirable in discrete choice modeling:

- The measure must take values in [0, 1], where 0 represents no fit and 1 corresponds to perfect fit.
- The measure should be directly related to the valid test statistic for the significance of all slope coefficients.

- The derivative of the measure with respect to the test statistic should comply with corresponding derivatives in a linear regression.

Estrella's measure is written as

$$R^2_{E1} = 1 - \left(\frac{\ln L}{\ln L_0}\right)^{-(2/N)\ln L_0}$$

Estrella suggests an alternative measure,

$$R^2_{E2} = 1 - [(\ln L - K)/\ln L_0]^{-(2/N)\ln L_0}$$

where $\ln L_0$ is computed with null parameter values, N is the number of observations used, and K represents the number of estimated parameters.

Other goodness-of-fit measures are summarized as follows:

$$\begin{aligned}
R^2_{CU1} &= 1 - \left(\frac{L_0}{L}\right)^{\frac{2}{N}} && \text{(Cragg-Uhler 1)}\\
R^2_{CU2} &= \frac{1-(L_0/L)^{\frac{2}{N}}}{1-L_0^{\frac{2}{N}}} && \text{(Cragg-Uhler 2)}\\
R^2_{A} &= \frac{2(\ln L - \ln L_0)}{2(\ln L - \ln L_0)+N} && \text{(Aldrich-Nelson)}\\
R^2_{VZ} &= R^2_A \frac{2\ln L_0 - N}{2\ln L_0} && \text{(Veall-Zimmermann)}
\end{aligned}$$

The AIC and SBC are computed as follows,

$$\mathrm{AIC} = -2\ \ln(L) + 2\,k$$

$$\mathrm{SBC} = -2\ \ln(L) + \ln(n)\,k$$

where $\ln(L)$ is the log-likelihood value for the model, k is the number of parameters estimated, and n is the number of observations (that is, the number of respondents).

Tests on Parameters

In general, the hypothesis to be tested can be written as

$$H_0 : \mathbf{h}(\theta) = 0$$

where $\mathbf{h}(\theta)$ is an r-by-1 vector-valued function of the parameters θ given by the r expressions specified in the TEST statement.

Let $\hat{V}$ be the estimate of the covariance matrix of $\hat{\theta}$. Let $\hat{\theta}$ be the unconstrained estimate of θ and $\tilde{\theta}$ be the constrained estimate of θ such that $h(\tilde{\theta}) = 0$. Let

$$A(\theta) = \partial h(\theta)/\partial\theta\ |_{\hat{\theta}}$$

Using this notation, the test statistics for the three kinds of tests are computed as follows:

- The Wald test statistic is defined as

  $$W = h'(\hat{\theta}) \left(A(\hat{\theta}) \hat{V} A'(\hat{\theta}) \right)^{-1} h(\hat{\theta})$$

  The Wald test is not invariant to reparameterization of the model (Gregory and Veall 1985; Gallant 1987, p. 219). For more information about the theoretical properties of the Wald test, see Phillips and Park (1988).

- The Lagrange multiplier test statistic is

  $$LM = \lambda' A(\tilde{\theta}) \tilde{V} A'(\tilde{\theta}) \lambda$$

  where λ is the vector of Lagrange multipliers from the computation of the restricted estimate $\tilde{\theta}$.

- The likelihood ratio test statistic is

  $$LR = 2 \left(L(\hat{\theta}) - L(\tilde{\theta}) \right)$$

  where $\tilde{\theta}$ represents the constrained estimate of θ and L is the concentrated log-likelihood value.

For each kind of test, under the null hypothesis the test statistic is asymptotically distributed as a χ^2 random variable with r degrees of freedom, where r is the number of expressions in the TEST statement. The p-values reported for the tests are computed from the $\chi^2(r)$ distribution and are only asymptotically valid.

Monte Carlo simulations suggest that the asymptotic distribution of the Wald test is a poorer approximation to its small sample distribution than that of the other two tests. However, the Wald test has the lowest computational cost, since it does not require computation of the constrained estimate $\tilde{\theta}$.

The following statements are an example of using the TEST statement to perform a likelihood ratio test:

```
proc mdc;
   model decision = x1 x2 / type=clogit
         choice=(mode 1 2 3);
   id pid;
   test 0.5 * x1 + 2 * x2 = 0 / lr;
run;
```

OUTEST= Data Set

The OUTEST= data set contains all the parameters that are estimated in a MODEL statement. The OUTEST= option can be used when the PROC MDC call contains one MODEL statement. There are additional restrictions. For the HEV and multinomial probit models, you need to specify exactly all possible elements of the choice set, since additional parameters (for example, SCALE1 or STD1) are generated automatically in the MDC procedure. Therefore, the following SAS statements are not valid when the OUTEST= option is specified:

```
proc mdc data=a outest=e;
   model y = x / type=hev choice=(alter);
run;
```

You need to specify all possible choices in the CHOICE= option since the OUTEST= option is specified as follows:

```
proc mdc data=a outest=e;
   model y = x / type=hev choice=(alter 1 2 3);
run;
```

When the NCHOICE= option is specified, no additional information about possible choices is required. Therefore, the following SAS statements are correct:

```
proc mdc data=a outest=e;
   model y = x / type=mprobit nchoice=3;
run;
```

The nested logit model does not produce the OUTEST= data set unless the NEST statement is specified.

Each parameter contains the estimate for the corresponding parameter in the corresponding model. In addition, the OUTEST= data set contains the following variables:

DEPVAR	the name of the dependent variable
METHOD	the estimation method
MODEL	the label of the MODEL statement if one is specified, or blank otherwise
STATUS	a character variable that indicates whether the optimization process reached convergence or failed to converge: 0 indicates that the convergence was reached, 1 indicates that the maximum number of iterations allowed was exceeded, 2 indicates a failure to improve the function value, and 3 indicates a failure to converge because the objective function or its derivatives could not be evaluated or improved, or linear constraints were dependent, or the algorithm failed to return to feasible region, or the number of iterations was greater than prespecified.
NAME	the name of the row of the covariance matrix for the parameter estimate, if the COVOUT option is specified, or blank otherwise
LIKLHD	the log-likelihood value
STDERR	standard error of the parameter estimate, if the COVOUT option is specified
TYPE	PARMS for observations that contain parameter estimates, or COV for observations that contain covariance matrix elements

The OUTEST= data set contains one observation for the MODEL statement giving the parameter estimates for that model. If the COVOUT option is specified, the OUTEST= data set includes additional observations for the MODEL statement giving the rows of the covariance matrix of parameter estimates. For covariance observations, the value of the _TYPE_ variable is COV, and the _NAME_ variable identifies the parameter associated with that row of the covariance matrix.

ODS Table Names

PROC MDC assigns a name to each table it creates. You can use these names to denote the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the Table 23.3.

Table 23.3 ODS Tables Produced in PROC MDC

ODS Table Name	Description	Option
ODS Tables Created by the MODEL Statement		
ResponseProfile	Response profile	Default
ClassLevels	Class levels	Default
FitSummary	Summary of nonlinear estimation	Default
GoodnessOfFit	Pseudo-R-square measures	Default
ConvergenceStatus	Convergence status	Default
ParameterEstimates	Parameter estimates	Default
CovB	Covariance of parameter estimates	COVB
CorrB	Correlation of parameter estimates	CORRB
LinCon	Linear constraints	ITPRINT
InputOptions	Input options	ITPRINT
ProblemDescription	Problem description	ITPRINT
IterStart	Optimization start	ITPRINT
IterHist	Iteration history	ITPRINT
IterStop	Optimization results	ITPRINT
ConvergenceStatus	Convergence status	ITPRINT
ParameterEstimatesResults	Resulting parameters	ITPRINT
LinConSol	Linear constraints evaluated at solution	ITPRINT
ODS Tables Created by the TEST Statement		
TestResults	Test results	Default

Examples: MDC Procedure

Example 23.1: Binary Data Modeling

The MDC procedure supports various multinomial choice models. However, you can also use PROC MDC to estimate binary choice models such as binary logit and probit because these models are special cases of multinomial models.

Spector and Mazzeo (1980) studied the effectiveness of a new teaching method on students' performance in an economics course. They reported grade point average (gpa), previous knowledge of the material (tuce), a dummy variable for the new teaching method (psi), and the final course grade (grade). A value of 1 is recorded for grade if a student earned the letter grade "A," and 0 otherwise.

The binary logit can be estimated using the conditional logit model. In order to use the MDC procedure, the data are converted as follows so that each possible choice corresponds to one observation:

```
data smdata;
   input gpa tuce psi grade;
datalines;
2.66      20      0        0
2.89      22      0        0
3.28      24      0        0
2.92      12      0        0

   ... more lines ...


data smdata1;
   set smdata;
   retain id 0;
   id + 1;

   /*-- first choice --*/
   choice1 = 1;
   choice2 = 0;
   decision = (grade = 0);
   gpa_2 = 0;
   tuce_2 = 0;
   psi_2 = 0;
   output;

   /*-- second choice --*/
   choice1 = 0;
   choice2 = 1;
   decision = (grade = 1);
   gpa_2 = gpa;
   tuce_2 = tuce;
   psi_2 = psi;
   output;
run;
```

The first 10 observations are displayed in Output 23.1.1. The variables related to grade=0 are omitted since these are not used for binary choice model estimation.

Output 23.1.1 Converted Binary Data

id	decision	choice2	gpa_2	tuce_2	psi_2
1	1	0	0.00	0	0
1	0	1	2.66	20	0
2	1	0	0.00	0	0
2	0	1	2.89	22	0
3	1	0	0.00	0	0
3	0	1	3.28	24	0
4	1	0	0.00	0	0
4	0	1	2.92	12	0
5	0	0	0.00	0	0
5	1	1	4.00	21	0

Consider the choice probability of the conditional logit model for binary choice:

$$P_i(j) = \frac{\exp(\mathbf{x}_{ij}'\boldsymbol{\beta})}{\sum_{k=1}^{2}\exp(\mathbf{x}_{ik}'\boldsymbol{\beta})}, \quad j = 1, 2$$

The choice probability of the binary logit model is computed based on normalization. The preceding conditional logit model can be converted as

$$P_i(1) = \frac{1}{1 + \exp((\mathbf{x}_{i2} - \mathbf{x}_{i1})'\boldsymbol{\beta})}$$

$$P_i(2) = \frac{\exp((\mathbf{x}_{i2} - \mathbf{x}_{i1})'\boldsymbol{\beta})}{1 + \exp((\mathbf{x}_{i2} - \mathbf{x}_{i1})'\boldsymbol{\beta})}$$

Therefore, you can interpret the binary choice data as the difference between the first and second choice characteristics. In the following statements, it is assumed that $\mathbf{x}_{i1} = \mathbf{0}$. The binary logit model is estimated and displayed in Output 23.1.2.

```
/*-- Conditional Logit --*/
proc mdc data=smdata1;
   model decision = choice2 gpa_2 tuce_2 psi_2 /
            type=clogit
            nchoice=2
            covest=hess;
   id id;
run;
```

Output 23.1.2 Binary Logit Estimates

The MDC Procedure

Conditional Logit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
choice2	1	-13.0213	4.9313	-2.64	0.0083
gpa_2	1	2.8261	1.2629	2.24	0.0252
tuce_2	1	0.0952	0.1416	0.67	0.5014
psi_2	1	2.3787	1.0646	2.23	0.0255

Consider the choice probability of the multinomial probit model:

$$P_i(j) = P[\epsilon_{i1} - \epsilon_{ij} < (\mathbf{x}_{ij} - \mathbf{x}_{i1})'\boldsymbol{\beta}, \ldots, \epsilon_{iJ} - \epsilon_{ij} < (\mathbf{x}_{ij} - \mathbf{x}_{iJ})'\boldsymbol{\beta}]$$

The probabilities of choice of the two alternatives can be written as

$$P_i(1) = P[\epsilon_{i2} - \epsilon_{i1} < (\mathbf{x}_{i1} - \mathbf{x}_{i2})'\boldsymbol{\beta}]$$

$$P_i(2) = P[\epsilon_{i1} - \epsilon_{i2} < (\mathbf{x}_{i2} - \mathbf{x}_{i1})'\boldsymbol{\beta}]$$

where $\begin{bmatrix} \epsilon_{i1} \\ \epsilon_{i2} \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}\right)$. Assume that $\mathbf{x}_{i1} = 0$ and $\sigma_{12} = 0$. The binary probit model is estimated and displayed in Output 23.1.3. You do not get the same estimates as that of the usual binary probit

model. The probabilities of choice in the binary probit model are

$$P_i(2) = P[\epsilon_i < \mathbf{x}_i'\boldsymbol{\beta}]$$

$$P_i(1) = 1 - P[\epsilon_i < \mathbf{x}_i'\boldsymbol{\beta}]$$

where $\epsilon_i \sim N(0,1)$. However, the multinomial probit model has the error variance $\mathrm{Var}(\epsilon_{i2} - \epsilon_{i1}) = \sigma_1^2 + \sigma_2^2$ if ϵ_{i1} and ϵ_{i2} are independent ($\sigma_{12} = 0$). In the following statements, unit variance restrictions are imposed on choices 1 and 2 ($\sigma_1^2 = \sigma_2^2 = 1$). Therefore, the usual binary probit estimates (and standard errors) can be obtained by multiplying the multinomial probit estimates (and standard errors) in Output 23.1.3 by $1/\sqrt{2}$.

```
/*-- Multinomial Probit --*/
proc mdc data=smdata1;
   model decision = choice2 gpa_2 tuce_2 psi_2 /
            type=mprobit
            nchoice=2
            covest=hess
            unitvariance=(1 2);
   id id;
run;
```

Output 23.1.3 Binary Probit Estimates

The MDC Procedure

Multinomial Probit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
choice2	1	-10.5392	3.5956	-2.93	0.0034
gpa_2	1	2.2992	0.9813	2.34	0.0191
tuce_2	1	0.0732	0.1186	0.62	0.5375
psi_2	1	2.0171	0.8415	2.40	0.0165

Example 23.2: Conditional Logit and Data Conversion

In this example, data are prepared for use by the MDCDATA statement. Sometimes, choice-specific information is stored in multiple variables. Since the MDC procedure requires multiple observations for each decision maker, you need to arrange the data so that there is an observation for each subject-alternative (individual-choice) combination. Simple binary choice data are obtained from Ben-Akiva and Lerman (1985). The following statements create the SAS data set:

```
data travel;
   length mode $ 8;
   input auto transit mode $;
datalines;
52.9   4.4 Transit
4.1   28.5 Transit
4.1   86.9 Auto
56.2  31.6 Transit
51.8  20.2 Transit
```

```
0.2   91.2 Auto
27.6  79.7 Auto
89.9  2.2  Transit
41.5  24.5 Transit
95.0  43.5 Transit
99.1  8.4  Transit

   ... more lines ...
```

The travel time is stored in two variables, auto and transit. In addition, the chosen alternatives are stored in a character variable, mode. The choice variable, mode, is converted to a numeric variable, decision, since the MDC procedure supports only numeric variables. The following statements convert the original data set, travel, and estimate the binary logit model. The first 10 observations of a relevant subset of the new data set and the parameter estimates are displayed in Output 23.2.1 and Output 23.2.2, respectively.

```
data new;
   set travel;
   retain id 0;
   id+1;
   /*-- create auto variable --*/
   decision = (upcase(mode) = 'AUTO');
   ttime = auto;
   autodum = 1;
   trandum = 0;
   output;
   /*-- create transit variable --*/
   decision = (upcase(mode) = 'TRANSIT');
   ttime = transit;
   autodum = 0;
   trandum = 1;
   output;
run;

proc print data=new(obs=10);
   var decision autodum trandum ttime;
   id id;
run;
```

Output 23.2.1 Converted Data

id	decision	autodum	trandum	ttime
1	0	1	0	52.9
1	1	0	1	4.4
2	0	1	0	4.1
2	1	0	1	28.5
3	1	1	0	4.1
3	0	0	1	86.9
4	0	1	0	56.2
4	1	0	1	31.6
5	0	1	0	51.8
5	1	0	1	20.2

The following statements perform the binary logit estimation:

```
proc mdc data=new;
   model decision = autodum ttime /
            type=clogit
            nchoice=2;
   id id;
run;
```

Output 23.2.2 Binary Logit Estimation of Modal Choice Data

The MDC Procedure

Conditional Logit Estimates

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
autodum	1	-0.2376	0.7505	-0.32	0.7516
ttime	1	-0.0531	0.0206	-2.57	0.0101

In order to handle more general cases, you can use the MDCDATA statement. Choice-specific dummy variables are generated and multiple observations for each individual are created. The following example converts the original data set travel by using the MDCDATA statement and performs conditional logit analysis. Interleaved data are output into the new data set new3. This data set has twice as many observations as the original travel data set.

```
proc mdc data=travel;
   mdcdata varlist( x1 = (auto transit) )
           select=mode
           id=id
           alt=alternative
           decvar=Decision / out=new3;
   model decision = auto x1 /
            nchoice=2
            type=clogit;
   id id;
run;
```

The first nine observations of the modified data set are shown in Output 23.2.3. The result of the preceding program is listed in Output 23.2.4.

Output 23.2.3 Transformed Model Choice Data

Obs	MODE	AUTO	TRANSIT	X1	ID	ALTERNATIVE	DECISION
1	TRANSIT	1	0	52.9	1	1	0
2	TRANSIT	0	1	4.4	1	2	1
3	TRANSIT	1	0	4.1	2	1	0
4	TRANSIT	0	1	28.5	2	2	1
5	AUTO	1	0	4.1	3	1	1
6	AUTO	0	1	86.9	3	2	0
7	TRANSIT	1	0	56.2	4	1	0
8	TRANSIT	0	1	31.6	4	2	1
9	TRANSIT	1	0	51.8	5	1	0

Output 23.2.4 Results Using MDCDATA Statement

The MDC Procedure

Conditional Logit Estimates

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
AUTO	1	-0.2376	0.7505	-0.32	0.7516
X1	1	-0.0531	0.0206	-2.57	0.0101

Example 23.3: Correlated Choice Modeling

Often, it is not realistic to assume that the random components of utility for all choices are independent. This example shows the solution to the problem of correlated random components by using multinomial probit and nested logit.

To analyze correlated data, trinomial choice data (1,000 observations) are created using a pseudo-random number generator by using the following statements. The random utility function is

$$U_{ij} = V_{ij} + \epsilon_{ij}, \quad j = 1, 2, 3$$

where

$$\epsilon_{ij} \sim N\left(0, \begin{bmatrix} 2 & .6 & 0 \\ .6 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

```
/*-- generate simulated series --*/
%let ndim = 3;
%let nobs = 1000;

data trichoice;
   array error{&ndim} e1-e3;
   array vtemp{&ndim} _temporary_;
   array lm{6} _temporary_ (1.4142136 0.4242641 1 0 0 1);
   retain nseed 345678;

   do id = 1 to &nobs;
      index = 0;
      /* generate independent normal variate */
      do i = 1 to &ndim;
         /* index of diagonal element */
         vtemp{i} = rannor(nseed);
      end;
      /* get multivariate normal variate */
      index = 0;
      do i = 1 to &ndim;
         error{i} = 0;
         do j = 1 to i;
            error{i} = error{i} + lm{index+j}*vtemp{j};
         end;
         index = index + i;
```

```
      end;
      x1 = 1.0 + 2.0 * ranuni(nseed);
      x2 = 1.2 + 2.0 * ranuni(nseed);
      x3 = 1.5 + 1.2 * ranuni(nseed);
      util1 = 2.0 * x1 + e1;
      util2 = 2.0 * x2 + e2;
      util3 = 2.0 * x3 + e3;
      do i = 1 to &ndim;
         vtemp{i} = 0;
      end;
      if ( util1 > util2 & util1 > util3 ) then
         vtemp{1} = 1;
      else if ( util2 > util1 & util2 > util3 ) then
         vtemp{2} = 1;
      else if ( util3 > util1 & util3 > util2 ) then
         vtemp{3} = 1;
      else continue;
      /*-- first choice --*/
      x = x1;
      mode = 1;
      decision = vtemp{1};
      output;
      /*-- second choice --*/
      x = x2;
      mode = 2;
      decision = vtemp{2};
      output;
      /*-- third choice --*/
      x = x3;
      mode = 3;
      decision = vtemp{3};
      output;
   end;
run;
```

First, the multinomial probit model is estimated (see the following statements). Results show that the standard deviation, correlation, and slope estimates are close to the parameter values. Note that $\rho_{12} = \frac{\sigma_{12}}{\sqrt{(\sigma_1^2)(\sigma_2^2)}} = \frac{0.6}{\sqrt{(2)(1)}} = 0.42$, $\sigma_1 = \sqrt{2} = 1.41$, $\sigma_2 = \sqrt{1} = 1$, and the parameter value for the variable x is 2.0. (See Output 23.3.1.)

```
/*-- Trinomial Probit --*/
proc mdc data=trichoice randnum=halton nsimul=100;
   model decision = x /
            type=mprobit
            choice=(mode 1 2 3)
            covest=op
            optmethod=qn;
   id id;
run;
```

Output 23.3.1 Trinomial Probit Model Estimation

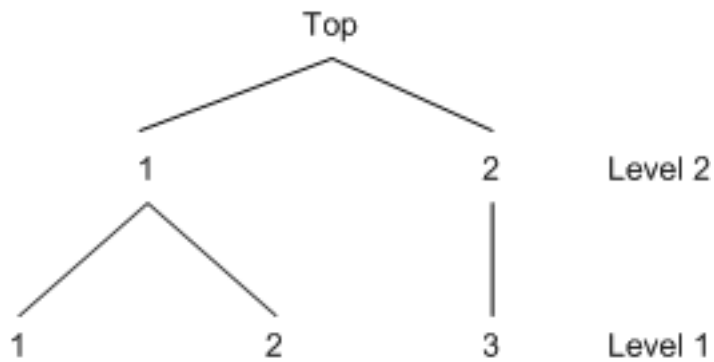
The MDC Procedure

Multinomial Probit Estimates

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
x	1	1.7685	0.1191	14.85	<.0001
STD_1	1	1.2514	0.1494	8.38	<.0001
RHO_21	1	0.3971	0.1087	3.65	0.0003

Figure 23.29 shows a two-level decision tree.

Figure 23.29 Nested Tree Structure



The following statements estimate the nested model shown in Figure 23.29:

```
/*-- Two-Level Nested Logit --*/
proc mdc data=trichoice;
   model decision = x /
            type=nlogit
            choice=(mode 1 2 3)
            covest=op
            optmethod=qn;
   id id;
   utility u(1,) = x;
   nest level(1) = (1 2 @ 1, 3 @ 2),
        level(2) = (1 2 @ 1);
run;
```

The estimated result (see Output 23.3.2) shows that the data support the nested tree model since the estimates of the inclusive value parameters are significant and are less than 1.

Output 23.3.2 Two-Level Nested Logit

The MDC Procedure

Nested Logit Estimates

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
x_L1	1	2.5907	0.1958	13.23	<.0001
INC_L2G1C1	1	0.8103	0.0859	9.43	<.0001
INC_L2G1C2	1	0.8189	0.0955	8.57	<.0001

Example 23.4: Testing for Homoscedasticity of the Utility Function

The conditional logit model imposes equal variances on random components of utility of all alternatives. This assumption can often be too restrictive and the calculated results misleading. This example shows several approaches to testing the homoscedasticity assumption.

The section "Getting Started: MDC Procedure" on page 1347 analyzes an HEV model by using Daganzo's trinomial choice data and displays the HEV parameter estimates in Figure 23.15. The inverted scale estimates for mode "2" and mode "3" suggest that the conditional logit model (which imposes equal variances on random components of utility of all alternatives) might be misleading. The HEV estimation summary from that analysis is repeated in Output 23.4.1.

Output 23.4.1 HEV Estimation Summary ($\theta_1 = 1$)

Model Fit Summary	
Dependent Variable	decision
Number of Observations	50
Number of Cases	150
Log Likelihood	-33.41383
Maximum Absolute Gradient	0.0000218
Number of Iterations	11
Optimization Method	Dual Quasi-Newton
AIC	72.82765
Schwarz Criterion	78.56372

You can estimate the HEV model with unit scale restrictions on all three alternatives ($\theta_1 = \theta_2 = \theta_3 = 1$) with the following statements.

```
/*-- HEV Estimation --*/
proc mdc data=newdata;
   model decision = ttime /
            type=hev
            nchoice=3
            hev=(unitscale=1 2 3, integrate=laguerre)
            covest=hess;
   id pid;
run;
```

Output 23.4.2 displays the estimation summary.

Output 23.4.2 HEV Estimation Summary ($\theta_1 = \theta_2 = \theta_3 = 1$)

The MDC Procedure

Heteroscedastic Extreme Value Model Estimates

Model Fit Summary	
Dependent Variable	decision
Number of Observations	50
Number of Cases	150
Log Likelihood	-34.12756
Maximum Absolute Gradient	6.7951E-9
Number of Iterations	5
Optimization Method	Dual Quasi-Newton
AIC	70.25512
Schwarz Criterion	72.16714

The test for scale equivalence (SCALE2=SCALE3=1) is performed using a likelihood ratio test statistic. The following SAS statements compute the test statistic (1.4276) and its p-value (0.4898) from the log-likelihood values in Output 23.4.1 and Output 23.4.2:

```
data _null_;
   /*-- test for H0: scale2 = scale3 = 1 --*/
   /*  ln L(max) = -34.1276                 */
   /*  ln L(0)   = -33.4138                 */
   stat = -2 * ( - 34.1276 + 33.4138 );
   df = 2;
   p_value = 1 - probchi(stat, df);
   put stat= p_value=;
run;
```

The test statistic fails to reject the null hypothesis of equal scale parameters, which implies that the random utility function is homoscedastic.

A multinomial probit model also allows heteroscedasticity of the random components of utility for different alternatives. Consider the utility function

$$U_{ij} = V_{ij} + \epsilon_{ij}$$

where

$$\boldsymbol{\epsilon}_i \sim N\left(\mathbf{0}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sigma_3^2 \end{bmatrix}\right)$$

This multinomial probit model is estimated by using the following statements:

```
/*-- Heteroscedastic Multinomial Probit --*/
proc mdc data=newdata;
   model decision = ttime /
            type=mprobit
            nchoice=3
            unitvariance=(1 2)
            covest=hess;
   id pid;
   restrict RHO_31 = 0;
run;
```

The estimation summary is displayed in Output 23.4.3.

Output 23.4.3 Heteroscedastic Multinomial Probit Estimation Summary

The MDC Procedure

Multinomial Probit Estimates

Model Fit Summary	
Dependent Variable	decision
Number of Observations	50
Number of Cases	150
Log Likelihood	-33.88604
Log Likelihood Null (LogL(0))	-54.93061
Maximum Absolute Gradient	5.60276E-6
Number of Iterations	8
Optimization Method	Dual Quasi-Newton
AIC	71.77209
Schwarz Criterion	75.59613
Number of Simulations	100
Starting Point of Halton Sequence	11

Next, the multinomial probit model with unit variances ($\sigma_1 = \sigma_2 = \sigma_3 = 1$) is estimated in the following statements:

```
/*-- Homoscedastic Multinomial Probit --*/
proc mdc data=newdata;
   model decision = ttime /
             type=mprobit
             nchoice=3
             unitvariance=(1 2 3)
             covest=hess;
   id pid;
   restrict RHO_21 = 0;
run;
```

The estimation summary is displayed in Output 23.4.4.

Output 23.4.4 Homoscedastic Multinomial Probit Estimation Summary

The MDC Procedure

Multinomial Probit Estimates

Model Fit Summary	
Dependent Variable	decision
Number of Observations	50
Number of Cases	150
Log Likelihood	-34.54252
Log Likelihood Null (LogL(0))	-54.93061
Maximum Absolute Gradient	1.37303E-7
Number of Iterations	5
Optimization Method	Dual Quasi-Newton
AIC	71.08505
Schwarz Criterion	72.99707
Number of Simulations	100
Starting Point of Halton Sequence	11

The test for homoscedasticity ($\sigma_3 = 1$) under $\sigma_1 = \sigma_2 = 1$ shows that the error variance is not heteroscedastic since the test statistic (1.313) is less than $\chi^2_{0.05,1} = 3.84$. The marginal probability or p-value computed in the following statements from the PROBCHI function is 0.2519:

```
data _null_;
   /*-- test for H0: sigma3 = 1 --*/
   /*   ln L(max) = -33.8860      */
   /*   ln L(0)   = -34.5425      */
   stat = -2 * ( -34.5425 + 33.8860 );
   df = 1;
   p_value = 1 - probchi(stat, df);
   put stat= p_value=;
run;
```

Example 23.5: Choice of Time for Work Trips: Nested Logit Analysis

This example uses sample data of 527 automobile commuters in the San Francisco Bay Area to demonstrate the use of the nested logit model.[1]

Brownstone and Small (1989) analyzed a two-level nested logit model that is displayed in Figure 23.30. The probability of choosing j at level 2 is written as

$$P_i(j) = \frac{\exp(\tau_j I_j)}{\sum_{j'=1}^{3} \exp(\tau_{j'} I_{j'})}$$

[1]These data were provided by Professor Kenneth Small. They were collected for the urban travel demand forecasting project, which was carried out by McFadden, Talvitie, and Associates (1977). The project was supported by the National Science Foundation, Research Applied to National Needs Program, through grants GI-43740 and APR74-20392 and by the Alfred P. Sloan Foundation through grant 74-21-8.

where $I_{j'}$ is an inclusive value and is computed as

$$I_{j'} = \ln\left[\sum_{k' \in C_{j'}} \exp(\mathbf{x}'_{ik'}\boldsymbol{\beta})\right]$$

The probability of choosing an alternative k is denoted as

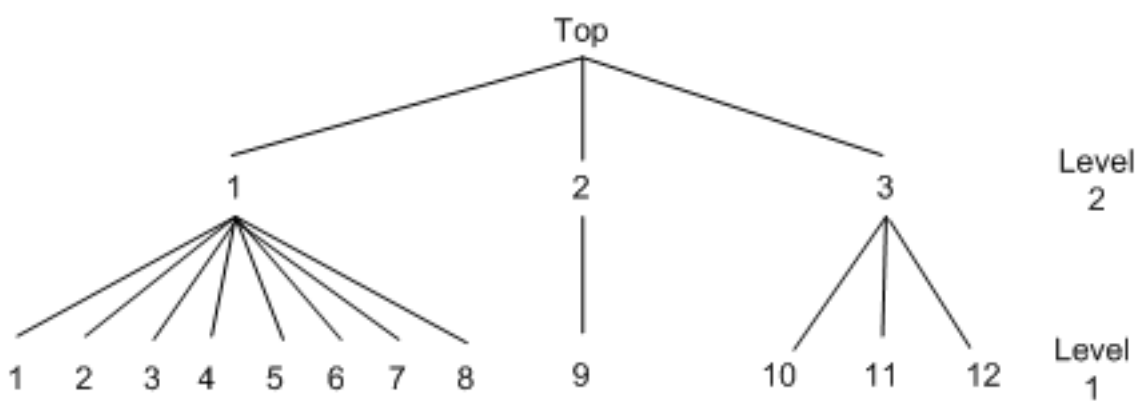
$$P_i(k|j) = \frac{\exp(\mathbf{x}'_{ik}\boldsymbol{\beta})}{\sum_{k' \in C_j} \exp(\mathbf{x}'_{ik'}\boldsymbol{\beta})}$$

The full information maximum likelihood (FIML) method maximizes the following log-likelihood function,

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{J} d_{ij}\,[\ln(P_i(k|j)) + \ln(P_i(j))]$$

where $d_{ij} = 1$ if a decision maker i chooses j, and 0 otherwise.

Figure 23.30 Decision Tree for Two-Level Nested Logit



Sample data of 527 automobile commuters in the San Francisco Bay Area have been analyzed by Small (1982); Brownstone and Small (1989). The regular time of arrival is recorded as between 42.5 minutes early and 17.5 minutes late, and indexed by 12 alternatives, using five-minute interval groups. For more information about these data, see Small (1982). The following statements estimate the two-level nested logit model:

```
/*-- Two-level Nested Logit --*/
proc mdc data=small maxit=200 outest=a;
   model decision = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l /
            type=nlogit
            choice=(alt);
   id id;
   utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l;
   nest level(1) = (1 2 3 4 5 6 7 8 @ 1, 9 @ 2, 10 11 12 @ 3),
        level(2) = (1 2 3 @ 1);
run;
```

The following statements add the upalt variable, which describes the choice at the upper level of the nested tree to the data set:

```
data small;
   set small;
   upalt=1;
   if alt=9 then upalt=2;
   if alt>9 then upalt=3;
run;
```

The following statements show and alternative specification, which uses the CHOICE= option with two nested levels that are represented by upalt and alt:

```
proc mdc data=upalt maxit=200;
   model decision = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l /
            type=nlogit
            choice=(upalt,alt);
   id id;
   utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l;
run;
```

The estimation summary, discrete response profile, and the FIML estimates are displayed in Output 23.5.1 through Output 23.5.3.

Output 23.5.1 Nested Logit Estimation Summary

The MDC Procedure

Nested Logit Estimates

Model Fit Summary	
Dependent Variable	decision
Number of Observations	527
Number of Cases	6324
Log Likelihood	-990.81912
Log Likelihood Null (LogL(0))	-1310
Maximum Absolute Gradient	4.93868E-6
Number of Iterations	18
Optimization Method	Newton-Raphson
AIC	2006
Schwarz Criterion	2057

Output 23.5.2 Discrete Choice Characteristics

Discrete Response Profile			
Index	**alt**	**Frequency**	**Percent**
0	1	6	1.14
1	2	10	1.90
2	3	61	11.57
3	4	15	2.85
4	5	27	5.12
5	6	80	15.18
6	7	55	10.44
7	8	64	12.14
8	9	187	35.48
9	10	13	2.47
10	11	8	1.52
11	12	1	0.19

Output 23.5.3 Nested Logit Estimates

The MDC Procedure

Nested Logit Estimates

Parameter Estimates					
Parameter	**DF**	**Estimate**	**Standard Error**	**t Value**	**Approx Pr > |t|**
r15_L1	1	1.1034	0.1221	9.04	<.0001
r10_L1	1	0.3931	0.1194	3.29	0.0010
ttime_L1	1	-0.0465	0.0235	-1.98	0.0474
ttime_cp_L1	1	-0.0498	0.0305	-1.63	0.1028
sde_L1	1	-0.6618	0.0833	-7.95	<.0001
sde_cp_L1	1	0.0519	0.1278	0.41	0.6850
sdl_L1	1	-2.1006	0.5062	-4.15	<.0001
sdlx_L1	1	-3.5240	1.5346	-2.30	0.0217
d2l_L1	1	-1.0941	0.3273	-3.34	0.0008
INC_L2G1C1	1	0.6762	0.2754	2.46	0.0141
INC_L2G1C2	1	1.0906	0.3090	3.53	0.0004
INC_L2G1C3	1	0.7622	0.1649	4.62	<.0001

Now policy makers are particularly interested in predicting shares of each alternative to be chosen by population. One application of such predictions are market shares. Going even further, it is extremely useful to predict choice probabilities out of sample; that is, under alternative policies.

Suppose that in this particular transportation example you are interested in projecting the effect of a new program that indirectly shifts individual preferences with respect to late arrival to work. This means that you manage to decrease the coefficient for the "late dummy" D2L, which is a penalty for violating some margin of arriving on time. Suppose that you alter it from an estimated –1.0941 to almost twice that level, –2.0941.

But first, in order to have a benchmark share, you predict probabilities to choose each particular option and output them to the new data set with the following additional statement:

```
   /*-- Create new data set with predicted probabilities --*/
   output out=predicted1 p=probs;
```

Having these in sample predictions, you sort the data by alternative and aggregate across each of them as shown in the following statements:

```
/*-- Sort the data by alternative --*/
proc sort data=predicted1;
   by alt;
run;

/*-- Calculate average probabilities of each alternative --*/
proc means data=predicted1 nonobs mean;
   var probs;
   class alt;
run;
```

Output 23.5.4 shows the summary table that is produced by the preceding statements.

Output 23.5.4 Average Probabilities of Choosing Each Particular Alternative

The MEANS Procedure

Analysis Variable : probs	
alt	Mean
1	0.0178197
2	0.0161712
3	0.0972584
4	0.0294659
5	0.0594076
6	0.1653871
7	0.1118181
8	0.1043445
9	0.3564940
10	0.0272324
11	0.0096334
12	0.0049677

Now you change the preference parameter for variable D2L. In order to fix all the parameters, you use the MAXIT=0 option to prevent optimization and the START= option in MODEL statement to specify initial parameters.

```
/*-- Two-level Nested Logit --*/
proc mdc data=small maxit=0 outest=a;
   model decision = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l /
            type=nlogit
            choice=(alt)
            start=( 1.1034 0.3931 -0.0465 -0.0498
                   -0.6618 0.0519 -2.1006 -3.5240
                   -2.0941 0.6762  1.0906  0.7622);
```

```
   id id;
   utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l;
   nest level(1) = (1 2 3 4 5 6 7 8 @ 1, 9 @ 2, 10 11 12 @ 3),
        level(2) = (1 2 3 @ 1);
   output out=predicted2 p=probs;
run;
```

You apply the same SORT and MEANS procedures as applied earlier to obtain the following summary table in Output 23.5.5.

Output 23.5.5 Average Probabilities of Choosing Each Particular Alternative after Changing the Preference Parameter

The MEANS Procedure

Analysis Variable : probs	
alt	Mean
1	0.0207766
2	0.0188966
3	0.1138816
4	0.0345654
5	0.0697830
6	0.1944572
7	0.1315588
8	0.1228049
9	0.2560674
10	0.0236178
11	0.0090781
12	0.0045128

Comparing the two tables shown in Output 23.5.4 and Output 23.5.5, you clearly see the effect of increased dislike of late arrival. People shifted their choices towards earlier times (alternatives 1–8) from the on-time option (alternative 9).

Brownstone and Small (1989) also estimate the two-level nested logit model with equal scale parameter constraints, $\tau_1 = \tau_2 = \tau_3$. Replication of their model estimation is shown in the following statements:

```
/*-- Nested Logit with Equal Dissimilarity Parameters --*/
proc mdc data=small maxit=200 outest=a;
   model decision = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l /
            samescale
            type=nlogit
            choice=(alt);
   id id;
   utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l;
   nest level(1) = (1 2 3 4 5 6 7 8 @ 1, 9 @ 2, 10 11 12 @ 3),
        level(2) = (1 2 3 @ 1);
run;
```

The parameter estimates and standard errors are almost identical to those in Brownstone and Small (1989, p. 69). Output 23.5.6 and Output 23.5.7 display the results.

Output 23.5.6 Nested Logit Estimation Summary with Equal Dissimilarity Parameters

The MDC Procedure

Nested Logit Estimates

Model Fit Summary	
Dependent Variable	decision
Number of Observations	527
Number of Cases	6324
Log Likelihood	-994.39402
Log Likelihood Null (LogL(0))	-1310
Maximum Absolute Gradient	2.97172E-6
Number of Iterations	16
Optimization Method	Newton-Raphson
AIC	2009
Schwarz Criterion	2051

Output 23.5.7 Nested Logit Estimates with Equal Dissimilarity Parameters

The MDC Procedure

Nested Logit Estimates

Parameter Estimates					
Parameter	**DF**	**Estimate**	**Standard Error**	**t Value**	**Approx Pr > |t|**
r15_L1	1	1.1345	0.1092	10.39	<.0001
r10_L1	1	0.4194	0.1081	3.88	0.0001
ttime_L1	1	-0.1626	0.0609	-2.67	0.0076
ttime_cp_L1	1	0.1285	0.0853	1.51	0.1319
sde_L1	1	-0.7548	0.0669	-11.28	<.0001
sde_cp_L1	1	0.2292	0.0981	2.34	0.0195
sdl_L1	1	-2.0719	0.4860	-4.26	<.0001
sdlx_L1	1	-2.8216	1.2560	-2.25	0.0247
d2l_L1	1	-1.3164	0.3474	-3.79	0.0002
INC_L2G1	1	0.8059	0.1705	4.73	<.0001

However, the test statistic for $H_0 : \tau_1 = \tau_2 = \tau_3$ rejects the null hypothesis at the 5% significance level since $-2 * (\ln L(0) - \ln L) = 7.15 > \chi^2_{0.05,2} = 5.99$. The p-value is computed in the following statements and is equal to 0.0280:

```
data _null_;
   /*-- test for H0: tau1 = tau2 = tau3 --*/
   /*   ln L(max) = -990.8191             */
   /*   ln L(0)   = -994.3940             */
   stat = -2 * ( -994.3940 + 990.8191 );
   df = 2;
   p_value = 1 - probchi(stat, df);
   put stat= p_value=;
run;
```

Example 23.6: Hausman's Specification Test

As discussed under multinomial and conditional logits, the odds ratios in the multinomial or conditional logits are independent of the other alternatives. (See the section "Multinomial Logit and Conditional Logit" on page 1380.) This property of the logit models is often viewed as rather restrictive and provides substitution patterns that do not represent the actual relationship among choice alternatives.

This independence assumption, called independence of irrelevant alternatives (IIA), can be tested with Hausman's specification test. According to Hausman and McFadden (1984), if a subset of choice alternatives is irrelevant, it can be omitted from the sample without changing the remaining parameters systematically.

Under the null hypothesis (IIA holds), omitting the irrelevant alternatives leads to consistent and efficient parameter estimates β_R, while parameter estimates β_U from the unrestricted model are consistent but inefficient. Under the alternative, only the parameter estimates β_U obtained from the unrestricted model are consistent.

This example demonstrates the use of Hausman's specification test to analyze the IIA assumption and decide on an appropriate model that provides less restrictive substitution patterns (nested logit or multinomial probit). A sample data set of 527 automobile commuters in the San Francisco Bay Area is used (Small 1982).[2] The regular time of arrival is recorded as between 42.5 minutes early and 17.5 minutes late, and is indexed by 12 alternatives, using five-minute interval groups. For more information about these data, see Small (1982).

The data can be divided into three groups: commuters who arrive early (alternatives 1–8), commuters who arrive on time (alternative 9), and commuters who arrive late (alternatives 10–12). Suppose that you want to test whether the IIA assumption holds for commuters who arrived on time (alternative 9).

Hausman's specification test is distributed as χ^2 with k degrees of freedom (equal to the number of independent variables) and can be written as

$$\chi^2 = (\hat{\beta_U} - \hat{\beta_R})'[\hat{V_U} - \hat{V_R}]^{-1}(\hat{\beta_U} - \hat{\beta_R})$$

where $\hat{\beta_R}$ and $\hat{V_R}$ represent parameter estimates and the variance-covariance matrix, respectively, from the model where the ninth alternative was omitted, and $\hat{\beta_U}$ and $\hat{V_U}$ represent parameter estimates and the variance-covariance matrix, respectively, from the full model. The following macro can be used to perform the IIA test for the ninth alternative:

```
/*-------------------------------------------------------------
 * name: %IIA
 * note: This macro test the IIA hypothesis using the Hausman's
 *       specification test. Inputs into the macro are as follows:
 *       indata:    input data set
 *       varlist:   list of RHS variables
 *       nchoice:   number of choices for each individual
 *       choice:    list of choices
 *       nvar:      number of independent variables
 *       nIIA:      number of choice alternatives used to test IIA
 *       IIA:       choice alternatives used to test IIA
```

[2]These data were provided by Professor Kenneth Small. They were collected for the urban travel demand forecasting project, which was carried out by McFadden, Talvitie, and Associates (1977). The project was supported by the National Science Foundation, Research Applied to National Needs Program, through grants GI-43740 and APR74-20392 and by the Alfred P. Sloan Foundation through grant 74-21-8.

```
 *       id:        ID variable
 *       decision:  0-1 LHS variable representing nchoice choices
 * purpose: Hausman's specification test
 *-----------------------------------------------------------------*/

%macro IIA(indata=, varlist=, nchoice=, choice= , nvar= , IIA= ,
               nIIA=, id= , decision=);

   %let n=%eval(&nchoice-&nIIA);

   proc mdc data=&indata outest=cov covout ;
      model &decision = &varlist /
               type=clogit
               nchoice=&nchoice;
      id &id;
   run;

   data two;
      set &indata;
      if &choice in &IIA and &decision=1 then output;
   run;

   data two;
      set two;
      keep &id ind;
      ind=1;
   run;

   data merged;
      merge &indata two;
      by &id;
      if ind=1 or &choice in &IIA then delete;
   run;

   proc mdc data=merged outest=cov2 covout ;
      model &decision = &varlist /
               type=clogit
               nchoice=&n;
      id &id;
   run;

   proc IML;
      use cov var{_TYPE_ &varlist};
         read first into BetaU;
         read all into CovVarU where(_TYPE_='COV');
      close cov;

      use cov2 var{_TYPE_ &varlist};
         read first into BetaR;
         read all into CovVarR where(_TYPE_='COV');
      close cov;

      tmp = BetaU-BetaR;
      ChiSq=tmp*ginv(CovVarR-CovVarU)*tmp`;
```

```
      if ChiSq<0 then ChiSq=0;
      Prob=1-Probchi(ChiSq, &nvar);
      Print "Hausman Test for IIA for Variable &IIA";
      Print ChiSq Prob;
   run; quit;

%mend IIA;
```

The following statement invokes the %IIA macro to test IIA for commuters who arrive on time:

```
%IIA( indata=small,
      varlist=r15 r10 ttime ttime_cp sde sde_cp sdl sdlx d2l,
      nchoice=12,
      choice=alt,
      nvar=9,
      nIIA=1,
      IIA=(9),
      id=id,
      decision=decision );
```

The obtained χ^2 of 7.9 and the p-value of 0.54 indicate that IIA holds for commuters who arrive on time (alternative 9). If the IIA assumption did not hold, the following model (nested logit), which reserves a subcategory for alternative 9, might be more appropriate. (See Output 23.30.)

```
proc mdc data=small maxit=200 outest=a;
   model decision = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l /
            type=nlogit
            choice=(alt);
   id id;
   utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l;
   nest level(1) = (1 2 3 4 5 6 7 8 @ 1, 9 @ 2, 10 11 12 @ 3),
        level(2) = (1 2 3 @ 1);
run;
```

Similarly, IIA could be tested for commuters who arrive approximately on time (alternative 8, 9, 10), as follows:

```
%IIA( indata=small,
      varlist=r15 r10 ttime ttime_cp sde sde_cp sdl sdlx d2l,
      nchoice=12,
      choice=alt,
      nvar=9,
      nIIA=3,
      IIA=(8 9 10),
      id=id,
      decision=decision );
```

Based on this test, independence of irrelevant alternatives is not rejected for this subgroup ($\chi^2 = 10.3$ and p-value = 0.326), and it is concluded that a more complex nested logit model with commuters who arrive approximately on time in one subcategory is not needed. Since the two Hausman's specification tests just performed did not reject IIA, it might be a good idea to test whether the nested logit model is even needed. This is done using the likelihood ratio test in the next example.

Example 23.7: Likelihood Ratio Test

This example is an extension of Example 23.6 and uses the same data.[3] It performs another specification test, the likelihood ratio test (LR). Suppose you are interested in testing whether the nested logit model (Output 23.30) with three subgroups that represent commuters who arrive early, on time, and late is more appropriate than the standard multinomial logit. This can be done by adding the TEST statement to the model as follows:

```
/*-- Restricted Model with Inclusive Value Parameters
     Constrained to One --*/
proc mdc data=small maxit=200 outest=a;
   model decision = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l /
            type=nlogit
            choice=(alt);
   id id;
   utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                    sdl sdlx d2l;
   nest level(1) = (1 2 3 4 5 6 7 8 @ 1, 9 @ 2, 10 11 12 @ 3),
        level(2) = (1 2 3 @ 1);
   test INC_L2G1C1=1, INC_L2G1C2=1, INC_L2G1C3=1 /LR;
run;
```

Output 23.7.1 Likelihood Ratio Test

The MDC Procedure

Nested Logit Estimates

Test Results

Test	Type	Statistic	Pr > ChiSq	Label
Test0	L.R.	8.11	0.0438	INC_L2G1C1 = 1 , INC_L2G1C2 = 1 , INC_L2G1C3 = 1

Based on this test, you can conclude that the inclusive values, INC_L2G1C1, INC_L2G1C2, and INC_L2G1C3 are jointly statistically different from the value 1 at the 5% level and therefore the nested logit is a more appropriate model. The LR test can be used to test other types of restrictions in the nested logit setting as long as one model can be nested within another.

[3]These data were provided by Professor Kenneth Small. They were collected for the urban travel demand forecasting project, which was carried out by McFadden, Talvitie, and Associates (1977). The project was supported by the National Science Foundation, Research Applied to National Needs Program, through grants GI-43740 and APR74-20392 and by the Alfred P. Sloan Foundation through grant 74-21-8.

References

Abramowitz, M., and Stegun, I. A., eds. (1970). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* 9th printing. New York: Dover.

Amemiya, T. (1981). "Qualitative Response Models: A Survey." *Journal of Economic Literature* 19:483–536.

Amemiya, T. (1985). *Advanced Econometrics*. Cambridge, MA: Harvard University Press.

Ben-Akiva, M., and Lerman, S. R. (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*. Cambridge, MA: MIT Press.

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

Brownstone, D., and Small, K. A. (1989). "Efficient Estimation of Nested Logit Models." *Journal of Business and Economic Statistics* 7:67–74.

Brownstone, D., and Train, K. (1999). "Forecasting New Product Penetration with Flexible Substitution Patterns." *Journal of Econometrics* 89:109–129.

Daganzo, C. (1979). *Multinomial Probit: The Theory and Its Application to Demand Forecasting*. New York: Academic Press.

Daganzo, C., and Kusnic, M. (1993). "Two Properties of the Nested Logit Model." *Transportation Science* 27:395–400.

Estrella, A. (1998). "A New Measure of Fit for Equations with Dichotomous Dependent Variables." *Journal of Business and Economic Statistics* 16:198–205.

Gallant, A. R. (1987). *Nonlinear Statistical Models*. New York: John Wiley & Sons.

Geweke, J. (1989). "Bayesian Inference in Econometric Models Using Monte Carlo Integration." *Econometrica* 57:1317–1339.

Geweke, J., Keane, M. P., and Runkle, D. (1994). "Alternative Computational Approaches to Inference in the Multinomial Probit Model." *Review of Economics and Statistics* 76:609–632.

Godfrey, L. G. (1988). *Misspecification Tests in Econometrics*. Cambridge: Cambridge University Press.

Greene, W. H. (1997). *Econometric Analysis*. 3rd ed. Upper Saddle River, NJ: Prentice-Hall.

Gregory, A. W., and Veall, M. R. (1985). "On Formulating Wald Tests for Nonlinear Restrictions." *Econometrica* 53:1465–1468.

Hajivassiliou, V. A. (1993). "Simulation Estimation Methods for Limited Dependent Variable Models." In *Econometrics*, edited by G. S. Maddala, C. R. Rao, and H. D. Vinod. Vol. 11 of Handbook of Statistics, 519–543. New York: Elsevier Science.

Hajivassiliou, V. A., McFadden, D. L., and Ruud, P. A. (1996). "Simulation of Multivariate Normal Rectangle Probabilities and Their Derivatives: Theoretical and Computational Results." *Journal of Econometrics* 72:85–134.

Hausman, J. A., and McFadden, D. (1984). "Specification Tests for the Multinomial Logit Model." *Econometrica* 52:1219–1240.

Hensher, D. A. (1986). "Sequential and Full Information Maximum Likelihood Estimation of a Nested Logit Model." *Review of Economics and Statistics* 68:657–667.

Hensher, D. A. (1993). "Using Stated Response Choice Data to Enrich Revealed Preference Discrete Choice Models." *Marketing Letters* 4:139–151.

Keane, M. P. (1994). "A Computationally Practical Simulation Estimator for Panel Data." *Econometrica* 62:95–116.

Keane, M. P. (1997). "Current Issues in Discrete Choice Modeling." *Marketing Letters* 8:307–322.

LaMotte, L. R. (1994). "A Note on the Role of Independence in *t* Statistics Constructed from Linear Statistics in Regression Models." *American Statistician* 48:238–240.

Luce, R. D. (1959). *Individual Choice Behavior: A Theoretical Analysis*. New York: John Wiley & Sons.

Maddala, G. S. (1983). *Limited-Dependent and Qualitative Variables in Econometrics*. New York: Cambridge University Press.

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. (1981). "Econometric Models of Probabilistic Choice." In *Structural Analysis of Discrete Data with Econometric Applications*, edited by C. F. Manski and D. McFadden, 2–50. Cambridge, MA: MIT Press.

McFadden, D., Talvitie, A. P., and Associates (1977). *The Urban Travel Demand Forecasting Project: Phase I Final Report Series.* Vol. 5. Berkeley: Institute of Transportation Studies, University of California, Berkeley.

McFadden, D. L., and Ruud, P. A. (1994). "Estimation by Simulation." *Review of Economics and Statistics* 76:591–608.

Phillips, C. B., and Park, J. Y. (1988). "On Formulating Wald Tests of Nonlinear Restrictions." *Econometrica* 56:1065–1083.

Powers, D. A., and Xie, Y. (2000). *Statistical Methods for Categorical Data Analysis*. San Diego: Academic Press.

Schmidt, P., and Strauss, R. (1975). "The Prediction of Occupation Using Multiple Logit Models." *International Economic Review* 16:471–486.

Small, K. (1982). "The Scheduling of Consumer Activities: Work Trips." *American Economic Review* 72:467–479.

Spector, L., and Mazzeo, M. (1980). "Probit Analysis and Economic Education." *Journal of Economic Education* 11:37–44.

Swait, J., and Bernardino, A. (2000). "Distinguishing Taste Variation from Error Structure in Discrete Choice Data." *Transportation Research Part B* 34:1–15.

Theil, H. (1969). "A Multinomial Extension of the Linear Logit Model." *International Economic Review* 10:251–259.

Train, K. E., Ben-Akiva, M., and Atherton, T. (1989). "Consumption Patterns and Self-Selecting Tariffs." *Review of Economics and Statistics* 71:62–73.

Subject Index

Syntax Index