Chapter 15
The MDC Procedure

Chapter Contents

Chapter 15
The MDC Procedure

Overview

The MDC (Multinomial Discrete Choice) procedure analyzes models where 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. Since the term *multinomial logit* is often used instead of *conditional logit* in econometrics literature, the term *simple multinomial logit* is used here to denote the model used by Schmidt and Strauss (1975), while *multinomial logit* is used as a synonym of *conditional logit*.

Unordered multiple choices are observed in wide areas of applications. For example, choices of where to live, political party affiliation, automobile, and mode of transportation are all unordered multiple choices. Economics and Psychology models often explain observed choices 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. 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 is 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. Different assumptions on the distribution of the error components give rise to different classes of models.

The features of discrete choice models available in the MDC procedure are summarized in the following table.

Table 15.1. 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 non-identical
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 non-identical

IEV stands for type I Extreme Value (or log-Weibull) distribution and has the form $F(\boldsymbol{\epsilon}_i)= \exp[-\sum_{j\in C_i}\exp(-\epsilon_{ij})]$; HEV stands for Heteroscedastic Extreme Value distribution and has the form $F(\boldsymbol{\epsilon}_i)= \exp[-\sum_{j\in C_i}\exp(-\frac{\epsilon_{ij}}{\theta_j})]$; GEV stands for Generalized Extreme Value distribution; MVN represents Multivariate Normal distribution; and ξ_{ij} is an error component. See the "Details" section later in this chapter for more information on ξ_{ij}.

Getting Started

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 CDF of the stochastic component is $F(\epsilon_{ij})$ =$(\exp(-\exp(-\epsilon_{ij}))$. The following statements are used to estimate this conditional logit model:

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

Note that the MDC procedure does not include the intercept term automatically like other regression procedures. The dependent variable decision takes value 1 when a specific alternative is chosen; otherwise it takes value 0. Each individual is allowed to choose one and only one of the possible alternatives. In other words, the variable decision takes value 1 one time only per 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) contains travel time (ttime1-ttime3) and choice (choice) variables.

```
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
   11.679  12.668  23.104  1  23.237  10.356  21.346  2
   13.236  16.019  10.087  3  20.052  16.861  14.168  3
   18.917  14.764  21.564  2  18.2    6.868   19.095  2
   10.777  16.554  15.938  1  20.003  6.377   9.314   2
   19.768  8.523   18.96   2  8.151   13.845  17.643  2
   22.173  18.045  15.535  1  13.134  11.067  19.108  2
   14.051  14.247  15.764  1  14.685  10.811  12.361  3
   11.666  10.758  16.445  1  17.211  15.201  17.059  3
   13.93   16.227  22.024  1  15.237  14.345  19.984  2
   10.84   11.071  10.188  1  16.841  11.224  13.417  2
   13.913  16.991  26.618  3  13.089  9.822   19.162  2
   16.626  10.725  15.285  3  13.477  15.509  24.421  2
   20.851  14.557  19.8    2  11.365  12.673  22.212  2
```

```
13.296  10.076  17.81   2  15.417  14.103  21.05   1
15.938  11.18   19.851  2  19.034  14.125  19.764  2
10.466  12.841  18.54   1  15.799  16.979  13.074  3
12.713  15.105  13.629  2  16.908  10.958  19.713  2
17.098  6.853   14.502  2  18.608  14.286  18.301  2
11.059  10.812  20.121  1  15.641  10.754  24.669  2
7.822   18.949  16.904  1  12.824  5.697   19.183  2
11.852  12.147  15.672  2  15.557  8.307   22.286  2
;
```

A new data set (newdata) is created since PROC MDC requires that each individual decision maker has one case per each alternative in his decision 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.

```
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 are as follows:

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

Figure 15.1. Transformed Modal Choice Data

The decision variable, decision, must have one non-zero value for each decision maker corresponding to the actual choice. When the RANK option is specified, the decision variable may contain rank data. For more details, see the "MODEL Statement" section later in this chapter. The following SAS statements estimate the conditional multinomial logit model using maximum likelihood:

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

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 (e.g., 1 3 6 8), you need to use the CHOICE= option.

The OPTMETHOD=QN option specifies the quasi-Newton optimization technique. The covariance matrix of the parameter estimates is obtained from the Hessian matrix since COVEST=HESS is specified. You may also specify COVEST=OP or COVEST=QML. See the “MODEL Statement” section for more details.

The MDC procedure produces a summary of model estimation displayed in Figure 15.2. Since there are multiple observations for each individual, the “Number of Cases” (150), is larger than the number of individuals, “Number of Observations” (50). Figure 15.3 shows the frequency distribution of the three choice alternatives. In this example, mode 2 is most frequently chosen.

```
                      The MDC Procedure

                 Conditional Logit Estimates

                      Model Fit Summary

Dependent Variable                         decision
Number of Observations                           50
Number of Cases                                 150
Log Likelihood                            -33.32132
Maximum Absolute Gradient                2.97024E-6
Number of Iterations                              6
Optimization Method                Dual Quasi-Newton
AIC                                        68.64265
Schwarz Criterion                          70.55467
```

Figure 15.2. Estimation Summary Table

```
                      The MDC Procedure

                 Conditional Logit Estimates

                  Discrete Response Profile

   Index     CHOICE      Frequency      Percent

       0          1             14        28.00
       1          2             29        58.00
       2          3              7        14.00
```

Figure 15.3. Choice Frequency

The MDC procedure computes nine goodness-of-fit measures for the discrete choice model. Seven of them are R^2 measures based on the null hypothesis, $\beta = \mathbf{0}$ (Figure 15.4). McFadden's likelihood ratio index (LRI) is the smallest in value.

```
                              The MDC Procedure

                          Conditional Logit Estimates

              Goodness-of-Fit Measures for Discrete Choice Models

   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                 0.6510    (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
```

Figure 15.4. Likelihood Ratio Test and R^2 Measures

Finally, the parameter estimate is displayed in Figure 15.5.

```
                              The MDC Procedure

                          Conditional Logit Estimates

                             Parameter Estimates

                                          Standard                 Approx
Parameter        DF      Estimate            Error    t Value    Pr > |t|     Gradient

ttime             1       -0.3572           0.0776      -4.60      <.0001      2.97E-6
```

Figure 15.5. Parameter Estimate of Conditional Logit

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 is 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.

```
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 15.6. It is expected that a decision maker will choose mode "1" based on predicted probabilities for all modes.

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

Figure 15.6. Out-Of-Sample Mode Choice Forecast

Nested Logit Modeling

A more general model can be specified using the nested logit model. Since the public transportation modes, 1 and 2, tend to be correlated, these two choices can be grouped. A decision tree displayed in Figure 15.7 is constructed.

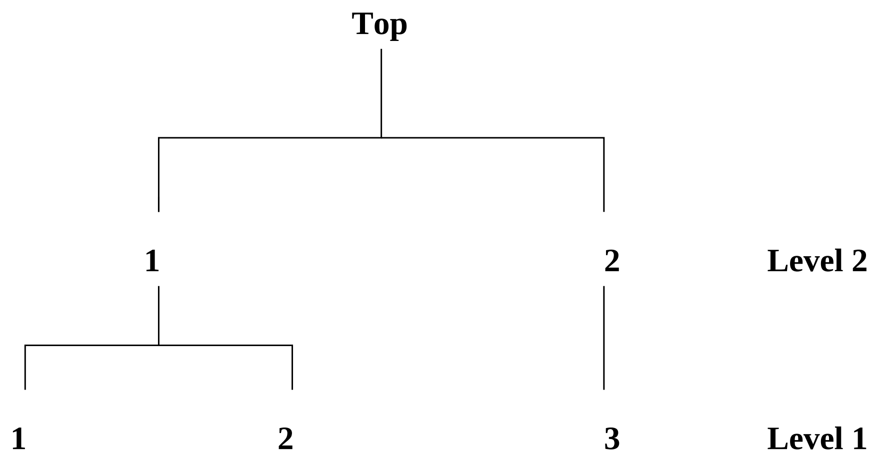


Figure 15.7. Decision Tree for Modal Choice

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.

```
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 15.8, parameter estimates of the inclusive values, INC_L2G1C1 and INC_L2G1C2, are indicative of a nested model structure. See the “Details” section later in this chapter for more details on the inclusive values.

The MDC Procedure

Nested Logit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
ttime_L1	1	-0.4040	0.1241	-3.25	0.0011	-6.07E-6
INC_L2G1C1	1	0.8016	0.4352	1.84	0.0655	-0.00003
INC_L2G1C2	1	0.8087	0.3591	2.25	0.0243	-0.00002

Figure 15.8. Two-Level Nested Logit Estimates

The nested logit model is estimated with a restriction INC_L2G1C1=INC_L2G1C2 by specifying the SAMESCALE option. The estimation result is displayed in Figure 15.9.

```
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 MDC Procedure

Nested Logit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
ttime_L1	1	-0.4025	0.1217	-3.31	0.0009	-4.43E-8
INC_L2G1	1	0.8209	0.3019	2.72	0.0066	2.422E-9

Figure 15.9. Nested Logit Estimates with One Dissimilarity Parameter

The nested logit model is equivalent to the conditional logit model if INC_L2G1C1 = INC_L2G1C2 = 1. You can verify this relationship by estimating the constrained nested logit model. Estimates are displayed in Figure 15.10.

```
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 fixedparm=(. 1 1);
run;
```

The MDC Procedure

Nested Logit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
ttime_L1	1	-0.3572	0.0776	-4.60	<.0001	-8.03E-7
INC_L2G1C1	0	1.0000	.	.	.	.
INC_L2G1C2	0	1.0000	.	.	.	.

Figure 15.10. Constrained Nested Logit Estimates

Multivariate Normal Utility Function

Consider the following random utility function:

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

where

$$\begin{bmatrix} \epsilon_{i1} \\ \epsilon_{i2} \\ \epsilon_{i3} \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} 1 & \rho_{12} & 0 \\ \rho_{12} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

The correlation coefficient (ρ_{12}) between U_{i1} and U_{i2} represents common neglected attributes of public transportation modes, 1 and 2. The following SAS statements estimate this trinomial probit model:

```
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 error component related to each of these choices has unit variance. The results of this constrained multinomial probit model estimation are displayed in Figure 15.11 and Figure 15.12. The test for ttime = 0 is rejected at the 1% significance level.

```
                         The MDC Procedure

                    Multinomial Probit Estimates

                         Model Fit Summary

          Dependent Variable                         decision
          Number of Observations                           50
          Number of Cases                                 150
          Log Likelihood                            -33.88604
          Maximum Absolute Gradient                 0.0002380
          Number of Iterations                              8
          Optimization Method                Dual Quasi-Newton
          AIC                                        71.77209
          Schwarz Criterion                          75.59613
```

Figure 15.11. Constrained Probit Estimation Summary

The MDC Procedure

Multinomial Probit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
ttime	1	-0.2307	0.0472	-4.89	<.0001	-0.00024
RHO_21	1	0.4820	0.3135	1.54	0.1242	2.564E-6

Figure 15.12. Multinomial Probit Estimates with Unit Variances

HEV and Multinomial Probit: Heteroscedastic Utility Function

When the stochastic error components are heteroscedastic and independent, you can model the data 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 CDF 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 be one. The following SAS statements estimate an HEV model under a unit scale restriction for mode "1" ($\theta_1 = 1$):

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

```
                        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 15.13. HEV Estimation Summary

```
                        The MDC Procedure

           Heteroscedastic Extreme Value Model Estimates

                       Parameter Estimates

                                  Standard                 Approx
Parameter      DF     Estimate       Error    t Value    Pr > |t|    Gradient

ttime           1      -0.4407      0.1798      -2.45      0.0143    0.000014
SCALE2          1       0.7765      0.4348       1.79      0.0741    3.972E-6
SCALE3          1       0.5753      0.2752       2.09      0.0366    -0.00002
```

Figure 15.14. HEV Parameter Estimates

It is worth mentioning that the estimate of the HEV model is not always stable since 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 since HEV models include the conditional logit as a special case, but 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 15.13 and Figure 15.2. 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 unit scale for all modes, since the HEV model with the unit variance for all modes reduces to the conditional logit model.

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

Figure 15.15 shows that the ttime coefficient is not close to that of the conditional logit model.

```
                              The MDC Procedure

               Heteroscedastic Extreme Value Model Estimates

                             Parameter Estimates

                                            Standard                      Approx
Parameter          DF        Estimate          Error     t Value       Pr > |t|       Gradient

ttime               1         -0.2926         0.0438       -6.68         <.0001        -6.8E-9
```

Figure 15.15. HEV Estimates with All Unit Scale Parameters

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 the adaptive Romberg-type integration method for analysis of the HEV model since this method requires excessive CPU time and hinders fitting the parameters. The following SAS statements produce the HEV estimates for this integration method displayed in Figure 15.17.

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

```
                              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 15.16. HEV Estimation Summary Using Alternative Integration Method

```
                           The MDC Procedure

             Heteroscedastic Extreme Value Model Estimates

                          Parameter Estimates

                                     Standard                 Approx
Parameter        DF      Estimate       Error    t Value    Pr > |t|     Gradient

ttime             1       -0.4580      0.1861      -2.46      0.0139      0.00012
SCALE2            1        0.7757      0.4283       1.81      0.0701     0.000035
SCALE3            1        0.6908      0.3384       2.04      0.0412     -0.00005
```

Figure 15.17. HEV Estimates Using Alternative Integration Method

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 15.16 and Figure 15.2.

When you impose unit scale restrictions on all choices, the HEV model gives the same estimates as the conditional logit. See Figure 15.18.

```
proc mdc data=newdata;
   model decision = ttime / type=hev nchoice=3
      hev=(unitscale=1 2 3 integrate=hardy) covest=hess;
   id pid;
run;
```

```
                           The MDC Procedure

             Heteroscedastic Extreme Value Model Estimates

                          Parameter Estimates

                                     Standard                 Approx
Parameter        DF      Estimate       Error    t Value    Pr > |t|     Gradient

ttime             1       -0.3572      0.0776      -4.60      <.0001     -0.00009
```

Figure 15.18. Alternative HEV Estimates with Unit Scale Restrictions

For comparison, we estimate a heteroscedastic multinomial probit model by imposing a zero restriction on the correlation parameter, $\rho_{31} = 0$. The MDC procedure requires at least one normalization of the error variance. The correlation parameters associated with a unit normalized variance are restricted to be zero. When multiple choices are specified in the UNITVARIANCE= option, the zero restriction on correlation coefficients applies to the last choice of the list. In the following example, the variances of the first and second choices are normalized. The default is UNITVARIANCE=(3). The UNITVARIANCE=(1 2) option imposes additional restrictions that $\rho_{32} = \rho_{21} = 0$. The utility function can be defined as

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

where

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

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

The MDC Procedure

Multinomial Probit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
ttime	1	-0.3206	0.0920	-3.49	0.0005	-5.6E-6
STD_3	1	1.6913	0.6906	2.45	0.0143	-4.67E-7
RHO_31	0	0	.	.	.	.

Figure 15.19. Heteroscedastic Multinomial Probit Estimates

Parameter Heterogeneity: Mixed Logit

One way of modeling heterogeneity of multinomial choices is to use the mixed logit model. The probability of choosing alternative j is written

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

where $\boldsymbol{\beta}$ is allowed to vary randomly.

For example, you can specify the distribution of the parameter, β, as follows:

$$\beta \sim N(\beta^*, \sigma_\beta^2)$$

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

When you estimate the model with normal random coefficients of ttime, you can use the following SAS statements:

```
proc mdc data=one type=mixedlogit;
   model decision = ttime / nchoice=3
         mixed=(normalparm=ttime);
   id pid;
run;
```

Let β^m and β^s be mean and scale parameters 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. 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 the non-negative parameters are estimated. The NORMALPARM=, UNIFORMPARM=, and LOGNORMALPARM= variables must be included in the right-hand-side of the MODEL statement. See the "Mixed Logit Model" section later in this chapter for more details. To estimate a mixed logit model using the transportation mode choice data, the MDC procedure requires MIXED= option for random components. Results of mixed logit estimation are displayed in Figure 15.20.

The MDC Procedure

Mixed Multinomial Logit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
ttime_M	1	-0.5342	0.1861	-2.87	0.0041	1.144E-6
ttime_S	1	0.2843	0.1715	1.66	0.0974	8.097E-7

Figure 15.20. Mixed Model Parameter Estimates

Syntax

The MDC procedure is controlled by the following statements:

PROC MDC *options* **;**
ID *variable* **;**
BY *variables* **;**
MODEL *dependent variables = regressors / options* **;**
UTILITY *U() = variables, . . ., U() = variables* **;**
NEST *LEVEL(value) = ((values)@(value),. . ., (values)@(value))* **;**
NLOPTIONS *options* **;**
RESTRICT *options* **;**
OUTPUT *options* **;**

Functional Summary

The statements and options used with the MDC procedure are summarized in the following table:

Description	Statement	Option
Data Set Options		
specify the input data set	MDC	DATA=
write parameter estimates to an output data set	MDC	OUTEST=
include covariances in the OUTEST= data set	MDC	COVOUT
write linear predictors and predicted probabilities to an output data set	OUTPUT	OUT=
Declaring the Role of Variables		
specify the ID variable	ID	
specify BY-group processing variables	BY	
Printing Control Options		
request all printing options	MODEL	ALL
print correlation matrix of the estimates	MODEL	CORRB
print covariance matrix of the estimates	MODEL	COVB
Model Estimation Options		
specify the choice variables	MODEL	CHOICE=()
specify the convergence criterion	MODEL	CONVERGE=
specify the type of covariance matrix	MODEL	COVEST=
specify the starting point of the Halton sequence	MODEL	HALTONSTART=
specify options specific to the HEV model	MODEL	HEV=()
set the initial values of parameters used by the iterative optimization algorithm	MODEL	INITIAL=()

Description	**Statement**	**Option**
specify the maximum number of iterations	MODEL	MAXITER=
specify the estimation method	MODEL	METHOD=
specify the options specific to mixed logit	MODEL	MIXED=()
specify the number of choices for each person	MODEL	NCHOICE=
specify the number of simulations	MODEL	NSIMUL=
specify the optimization technique	MODEL	OPTMETHOD=
specify the type of random number generators	MODEL	RANDNUM=
specify that initial values are generated using random numbers	MODEL	RANDINIT
specify the rank dependent variable	MODEL	RANK
specify optimization restart options	MODEL	RESTART=
specify a restriction on inclusive parameters	MODEL	SAMESCALE
specify a seed of pseudo-random number generation	MODEL	SEED=
specify a stated preference data restriction on inclusive parameters	MODEL	SPSCALE
specify the type of the model	MODEL	TYPE=
specify normalization restrictions on multinomial probit error variances	MODEL	UNITVARIANCE=
NESTED Logit Related Options		
specify the tree structure	NEST	LEVEL()=
specify the type of utility function	UTILITY	U()=
Output Control Options		
output predicted values	OUTPUT	P=
output estimated linear predictor	OUTPUT	XBETA=

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. See “OUTEST= Data Set” later in this chapter for information on the contents of this data set.

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 every MODEL statement: ALL, CONVERGE=, CORRB, COVB, COVEST=, HALTONSTART=, ITPRINT, MAXITER=, METHOD=, NOPRINT, NSIMUL=, OPTMETHOD=, RANDINIT, RANK, RESTART=, SAMESCALE, SEED=, SPSCALE, TYPE=, and UNITVARIANCE=.

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.

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.

MODEL Statement

MODEL *dependent = regressors / options ;*

The MODEL statement specifies the dependent variable and independent regressor variables for the regression model. When the nested 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. Multiple choice variables are only allowed for nested logit models. If all possible alternatives are written with the variable name, the MDC procedure checks all values of the choice variable. The CHOICE=(X 1 2 3) specification implies that the value of X should be 1, 2, or 3. On the other hand, the CHOICE=(X) specification considers all distinctive non-missing values of X as elements of the choice set.

NCHOICE= *number*

specifies the number of choices for multinomial choice models when all individuals have the same choice set. 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=200. 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 are 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 (1000 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 will add a generated random number u to the initial value and compute an objective function value 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 contains 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.

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

SPSCALE
specifies that the parameters of the inclusive values are 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 the stated preference data.

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

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*)
UNITVARIANCE= (*number-list*)
specifies normalization restrictions on error variances of multinomial probit for the choices whose numbers are given in the list.

HEV Estimation Options

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 statements:

```
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 Logit Estimation Options

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 log-normally distributed.

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.

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.

Restart Options

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.

```
model y = x1 x2 / type=clogit
   restart=(addvalue=(.01 .01 .01 .001 .001));
```

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 in 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 number of maximum iterations for the second stage of the estimation.

ADDRANDOM

ADDRANDOM= *value*

specifies random added values to the estimates in the initial stage. With the ADDRANDOM option, $U(-1, 1)$ random numbers are created and added to the estimate obtained in the initial stage. When the ADDRANDOM=r option is specified, $U(-r, r)$ random numbers generated. The restart initial value is determined based on the given number of random searches (1000 for conditional or nested logit, and 500 for other models).

ADDVALUE= (*value-list*)

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

Printing Options

ALL

requests all printing options.

COVB

prints the estimated covariances of the parameter estimates.

CORRB

prints the estimated correlation matrix of the parameter estimates.

COVEST= *value*

The COVEST= option specifies the type of covariance matrix. Possible values are OP, HESSIAN, and QML. 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 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. The supported covariance types are

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

ITPRINT

prints the objective function and parameter estimates at each iteration. The objective function is the full log-likelihood function for the maximum likelihood method. When the ITPRINT option is specified in the presence of the NLOPTIONS statement, all printing options in the NLOPTIONS statement are ignored. The ITPRINT option is equivalent to the PALL option in the NLOPTIONS statement.

NOPRINT

suppresses all printed output.

Estimation Control Options

You can also specify detailed optimization options in the NLOPTIONS statement. The OPTMETHOD= option overrides the TECH= option in the NLOPTIONS statement. Note that 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 as the parameter estimates are printed 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 default is MAXITER=100.

OPTMETHOD= *value*

The OPTMETHOD= option specifies the optimization technique when the estimation method uses non-linear optimization.

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 TECH=NRA option in the NLOPTIONS statement. For the conditional and nested logit models the default is OPTMETHOD=NR. For other models OPTMETHOD=QN is the default.

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
```

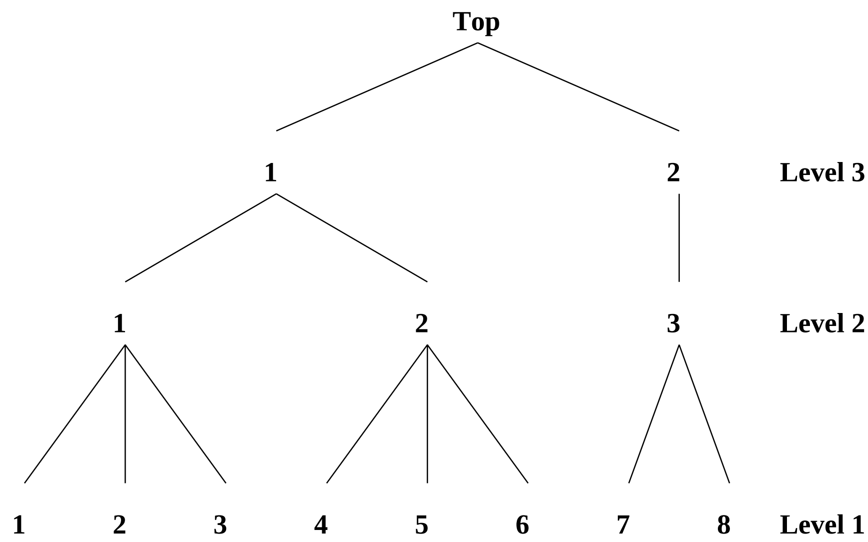


Figure 15.21. Three-Level Tree

You can use the NEST statement to specify the tree structure displayed in Figure 15.21.

```
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);
```

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 15.22.

```
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);
```

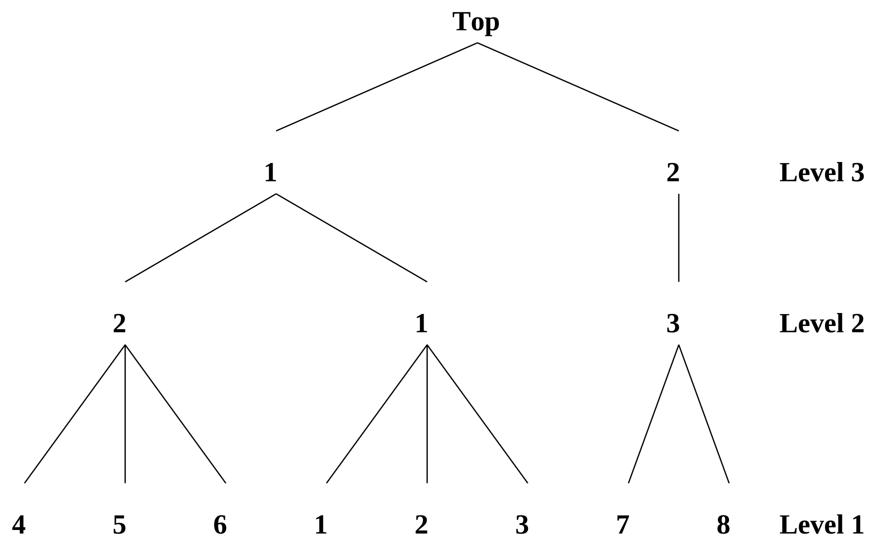


Figure 15.22. Alternative Three-Level Tree

However, the SAS statement with a different sequence of choice specification at higher levels builds the same tree as displayed in Figure 15.21 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 data sequence in 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.

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

The tree is constructed as in Figure 15.23.

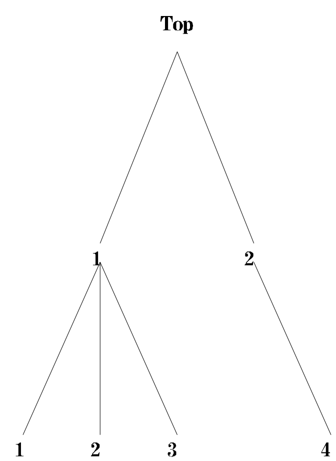


Figure 15.23. Two-Level Tree

Of course, another model is estimated if you specify the decision tree as in Figure 15.24. The different nested tree structure is specified in the following SAS statement:

```
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);
```

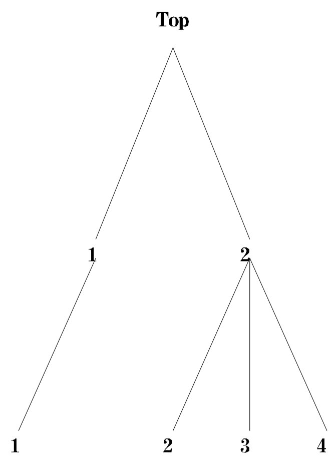


Figure 15.24. Two-Level Tree

NLOPTIONS Statement

NLOPTIONS *options* **;**

The NLOPTIONS statement specifies non-linear optimization options. The specified options are similar to those in PROC NLP.

ABSCONV | ABSTOL=*r*

specifies the absolute function convergence criterion:

- For minimization, termination requires

$$f^{(k)} = f(x^{(k)}) \leq r$$

- For maximization, termination requires

$$f^{(k)} = f(x^{(k)}) \geq r$$

The default value of ABSTOL is:

- For minimization: the negative square root of the largest double precision value,
- For maximization: the positive square root of the largest double precision value.

ABSFCONV | ABSFTOL=*r <n>*

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

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

The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

ABSGCONV | ABSGTOL=*r* <*n*>

specifies the absolute gradient convergence criterion. Termination requires the maximum absolute gradient element to be small:

$$\max_j |g_j(x^{(k)})| \le r$$

The default value is r=1E-5. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

ABSXCONV | ABSXTOL=*r* <*n*>

specifies the absolute parameter convergence criterion. For all techniques, termination requires a small Euclidean distance between successive parameter vectors,

$$\| x^{(k)} - x^{(k-1)} \|_2 \le r$$

The default value is r=0. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

DAMPSTEP | DS =*r*

specifies that the initial step size value $\alpha^{(0)}$ for each line search (used by QUANEW, CONGRA, or NEWRAP) cannot be larger than r times the step size value used in the former iteration. If DAMPSTEP is specified but not factor r, the default is $r = 2$. The DAMPSTEP=r option can prevent the line-search algorithm from repeatedly stepping into regions where some objective functions are difficult to compute or where they could lead to floating point overflows during the computation of objective functions and their derivatives. The DAMPSTEP=r option can save time-consuming function calls during the line searches of objective functions that result in very small step sizes.

FCONV | FTOL=*r* <*n*>

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

$$\frac{|f(x^{(k)}) - f(x^{(k-1)})|}{\max(|f(x^{(k-1)})|, FSIZE)} \le r$$

where $FSIZE$ is defined by the FSIZE= option. The default value is r=$10^{-FDIGITS}$, where FDIGITS is either specified or is set by default to $-log10(\epsilon)$ where ϵ is the machine precision. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FSIZE=*r*

specifies the $FSIZE$ parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. For more details, see the FCONV= and GCONV= options.

GCONV | GTOL=*r* <*n*>

specifies the relative gradient convergence criterion. For all techniques, termination requires that the normalized predicted function reduction is small:

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

where $FSIZE$ is defined by the FSIZE= option. For the CONGRA technique (where a reliable Hessian estimate G is not available)

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

is used. The default value is r=1E-8. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

LINESEARCH | LIS=*i*

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques. The value of i can be $1, \ldots, 8$. For CONGRA, QUANEW, and NEWRAP, the default is $i = 2$.

LIS=1	specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is similar to one used by the Harwell subroutine library.
LIS=2	specifies a line-search method that needs more function than gradient calls for quadratic and cubic interpolation and cubic extrapolation.
LIS=3	specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation.
LIS=4	specifies a line-search method that needs the same number of function and gradient calls for stepwise extrapolation and cubic interpolation.
LIS=5	specifies a line-search method that is a modified version of LIS=4.
LIS=6	specifies a golden section line search, which uses only function values for linear approximation.
LIS=7	specifies a bisection line search, which uses only function values for linear approximation.
LIS=8	specifies an Armijo line-search technique, which uses only function values for linear approximation.

PALL | ALL

prints all optional output.

PHISTORY | PHIS

prints the optimization history.

TECHNIQUE | TECH=*name*

specifies the optimization technique. Valid values for *name* are as follows:

CONGRA | CG chooses one of four different conjugate-gradient optimization algorithms that can be more precisely defined with the UPDATE= option and modified with the LINESEARCH= option. The conjugate-gradient techniques need only $O(n)$ memory, compared to the $O(n^2)$ memory for the other optimization techniques, where n is the number of parameters. On the other hand, the conjugate gradient techniques can be significantly slower than other optimization techniques and should be used only when insufficient memory is available for more efficient techniques. For $n \geq 400$, this is the default optimization technique. The algorithm uses only first-order derivatives.

DBLDOG | DD performs a version of double dogleg optimization, which uses the gradient to update an approximation of the Cholesky factor of the Hessian. This technique is in many aspects very similar to the dual quasi-Newton method but does not use line search.

NEWRAP | NRA performs a usually stable but, for large dense problems, memory- and time-consuming Newton-Raphson optimization technique. The algorithm combines a line-search algorithm with ridging. The line-search algorithm LIS=2 is the default. This algorithm uses second-order derivatives.

NRRIDG | NRR performs a usually stable but, for large problems, memory- and time-consuming Newton-Raphson optimization technique. For fewer than 40 parameters, this is the default technique for general optimization. The algorithm uses second-order derivatives. Since TECH=NRRIDG uses an orthogonal decomposition of the approximate Hessian, each iteration of TECH=NRRIDG can be slower than that of TECH=NEWRAP, which works with Cholesky decomposition. However, usually TECH=NRRIDG needs fewer iterations than TECH=NEWRAP.

QUANEW | QN chooses one of four quasi-Newton optimization algorithms that can be defined more precisely with the UPDATE= option and modified with the LINESEARCH= option. When you choose this option, UPDATE= DBFGS by default. The QUANEW technique is the default optimization technique if there are more than 40 and fewer than 400 parameters to estimate. The QUANEW algorithm uses only first-order derivatives of the objective function and, if available, of the nonlinear constraint functions.

TRUREG | TR performs a usually very stable but, for large problems, memory- and time-consuming trust region optimization technique. The algorithm uses second-order derivatives.

UPDATE | UPD=*name*

specifies the update method for the (dual) quasi-Newton, double dogleg, or conjugate-gradient optimization technique.

For TECHNIQUE=QUANEW, the following updates can be used (the default is DBFGS):

BFGS	performs original BFGS (Broyden, Fletcher, Goldfarb, and Shanno) update of the inverse Hessian matrix.
DBFGS	performs the dual BFGS (Broyden, Fletcher, Goldfarb, and Shanno) update of the Cholesky factor of the Hessian matrix. This is the default.
DDFP	performs the dual DFP (Davidon, Fletcher, and Powell) update of the Cholesky factor of the Hessian matrix.
DFP	performs the original DFP (Davidon, Fletcher, and Powell) update of the inverse Hessian matrix.

For TECHNIQUE=DBLDOG, the following updates can be used (the default is DBFGS):

DBFGS	performs the dual BFGS (Broyden, Fletcher, Goldfarb, and Shanno) update of the Cholesky factor of the Hessian matrix. This is the default.
DDFP	performs the dual DFP (Davidon, Fletcher, and Powell) update of the Cholesky factor of the Hessian matrix.

For TECHNIQUE=CONGRA, the following updates can be used (the default is DBFGS):

DBFGS	performs the dual BFGS (Broyden, Fletcher, Goldfarb, and Shanno) update of the Cholesky factor of the Hessian matrix. This is the default.
DDFP	performs the dual DFP (Davidon, Fletcher, and Powell) update of the Cholesky factor of the Hessian matrix.

XCONV | XTOL=*r* <*n*>

specifies the relative parameter convergence criterion. For all techniques, termination requires a small relative parameter change in subsequent iterations:

$$\frac{\max_j |x_j^{(k)} - x_j^{(k-1)}|}{\max(|x_j^{(k)}|, |x_j^{(k-1)}|, XSIZE)} \leq r$$

The default value is r=0. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

XSIZE=*r*

specifies the $XSIZE$ parameter of the relative parameter termination criterion. The default value is $r = 0$. r must be greater than or equal to zero. For more detail see the XCONV= option.

OUTPUT Statement

OUTPUT *options* **;**

The MDC procedure supports the OUTPUT statement. 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).

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

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 ($\mathbf{x}'\boldsymbol{\beta}$) values. However, the XBETA= option is not supported in the nested logit model.

RESTRICT Statement

RESTRICT *options* **;**

The RESTRICT statement specifies simple parameter restrictions. The sequence of elements in the FIXEDPARM=, LBOUND=, and UBOUND= option must correspond to the printed sequence of parameter estimates. A RESTRICT statement can be specified for each MODEL statement.

FIXEDPARAMETER= (*value-list***)**
FIXEDPARM= (*value-list***)**
specifies the fixed values of parameters. When the LBOUND= or UBOUND= option is specified, the values specified in the FIXEDPARM= option must satisfy the specified boundary condition.

LOWERBOUND= (*value-list***)**
LBOUND= (*value-list***)**
specifies the lower bounds of parameters. When there is a FIXEDPARM= option present and the corresponding element in the FIXEDPARM= option does not have a missing value, the relevant element of the LBOUND= option is ignored.

UPPERBOUND= (*value-list***)**
UBOUND= (*value-list***)**
specifies the upper bounds of parameters. When there is a FIXEDPARM= option present and the corresponding element in the FIXEDPARM= option does not have a missing value, the relevant element of the UBOUND= option is ignored.

ALL
specifies that the single element of the FIXEDPARM=, LBOUND=, and UBOUND= options is expanded to all parameters. For example, a model with four parameters can have non-negative boundary constraints if the following RESTRICT

statement is specified:

```
restrict lbound=(0) / all;
```

However, only the first parameter is bounded below by 0 if the ALL option is not specified.

The RESTRICT statement is experimental for Version 8.2. The current syntax will change in later releases.

UTILITY Statement

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

The UTILITY statement can be used in estimating a nested logit model. The U()= option can have two arguments. The first argument contains level information while the second argument is related to choice information. The second argument can be omitted. 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

```
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);
```

Of course, you also can specify

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

The variable, cost, should be listed in the MODEL statement. When the additional explanatory variable, dummy, is included at level 2, another U()= option needs to be specified.

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

Details

Multinomial Discrete Choice Modeling

When the dependent variable takes multiple discrete values, you can use multinomial discrete choice modeling to analyze the data. Let the random utility function be defined by

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

where V_{ij} is a non-stochastic utility function and ϵ_{ij} is a random component. In multinomial discrete choice models, the utility function is assumed to be linear, so that $V_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta}$. The multinomial probit model is derived when the error disturbances, ϵ_{ij}, have non-identical and non-independent normal distribution. In conditional logit models, the error disturbances have independent type I extreme value distribution function, $\exp(-\exp(-\epsilon_{ij}))$, also known as Gumbel. The nested logit model is derived when the random components have identical and non-independent Gumbel distribution. The heteroscedastic extreme value (HEV) model assumes non-identical and independent distribution of the error disturbance.

A 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 in the estimation process. In addition, a 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 as follows:

$$U_{ij} \geq \mathbf{max}_{k \in C_i, k \neq j} U_{ik}$$

where C_i is a choice set of individual i. Using properties of the Gumbel (type I extreme value) distribution, the probability of choosing an alternative j among n_i choices for individual i can be written

$$\begin{aligned} P_i(j) &= P[\mathbf{x}'_{ij}\boldsymbol{\beta} + \epsilon_{ij} \geq \mathbf{max}_{k \in C_i}(\mathbf{x}'_{ik}\boldsymbol{\beta} + \epsilon_{ik})] \\ &= \frac{\exp(\mathbf{x}'_{ij}\boldsymbol{\beta})}{\sum_{k \in C_i} \exp(\mathbf{x}'_{ik}\boldsymbol{\beta})} \end{aligned}$$

Simple Multinomial and Conditional Logit

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

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

where y_i is the choice variable that contains possible alternatives of each decision maker. For model identification, one often assumes that $\boldsymbol{\beta}_0 = 0$. The simple multinomial logit model is reduced to the binary logit model if $M = 1$. The log-odds ratio of alternative j and k is

$$\ln\left[\frac{P_{ij}}{P_{ik}}\right] = \mathbf{x}_i'(\boldsymbol{\beta}_j - \boldsymbol{\beta}_k)$$

This type of multinomial choice modeling has a couple of weaknesses; it has too many parameters and it is difficult to interpret. The log-likelihood function of the simple multinomial logit model is written

$$\ell = \sum_{i=1}^{N}\sum_{j=0}^{M} d_{ij}\ln P[y_i = j]$$

where

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

The simple multinomial logit model can be estimated with TYPE=CLOGIT in the MDC procedure.

The conditional logit model, also called multinomial logit model, is similarly defined when choice specific data are available.

$$P[y_i = j] = \frac{\exp(\mathbf{x}_{ij}'\boldsymbol{\beta})}{\sum_{k\in C_i}\exp(\mathbf{x}_{ik}'\boldsymbol{\beta})}$$

where there are n_i choices in each individual's choice set, C_i. The log-likelihood function is written

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

Using properties of type I extreme value distribution, the probability of choosing an alternative j from n_i choices of individual i can be defined as follows:

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

The problematic aspect of the conditional 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.

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

It is evident that the probability ratio is only affected by choices j and l. Note that this IIA property is caused by an assumption of an independent and identical distribution of the random utility function.

Heteroscedastic Extreme Value Model

A heteroscedastic extreme value (HEV) model is used when the random components of the utility function are assumed to be non-identical. The HEV model is derived from the type I extreme value error distribution. 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 of choosing an alternative j 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

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

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

Therefore, the log-likelihood function can be written as

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

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

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

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, the utility function of each decision maker can be decomposed into a deterministic component (linear combination of observed variables, $\mathbf{x}'_{ij}\boldsymbol{\beta}$) and stochastic components $\xi_{ij} + \epsilon_{ij}$ with zero mean.

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

where the error component ξ_{ij} can be correlated among alternatives and heteroscedastic for each individual while ϵ_{ij} is independently and identically distributed. The conditional logit model is derived if you assume ϵ_{ij} has an *iid* Gumbel distribution, and $V(\xi_{ij}) = 0$. The error component has a specific type of distribution with probability density function $f(\xi_{ij}|\boldsymbol{\gamma})$, where $\boldsymbol{\gamma}$ is a parameter vector of the distribution of ξ_{ij}. The choice probability of an alternative j is written as

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

where the conditional probability is

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

In general, the mixed logit model does not have an exact likelihood function since 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

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

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 $\boldsymbol{\mu}$ has zero mean with density function $\psi(\boldsymbol{\mu}|\boldsymbol{\gamma})$. The observed data vector ($\mathbf{z}_{ij}$) of the error component may contain $\mathbf{x}_{ij}$. 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, the parameters β^* are random parameters such that

$$\beta^*_m = b_m + s_m \epsilon_\beta$$

where b_m and s_m are parameters and

$$\epsilon_\beta \sim (0, \sigma^2)$$

The observed data vector, $\mathbf{w}_{ij}$, is a subset of $\mathbf{x}_{ij}$. Three types of distributions are supported:

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

$$\epsilon_\beta \sim U(-1,1)$$

$$\ln(\epsilon_\beta) \sim N(0,1)$$

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

Multinomial Probit

Consider the random utility function

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

where

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

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

The dimension of the error covariance is determined by the number of alternatives. The choice probability of the multinomial probit model is written as

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

Since evaluation of the 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 Geweke, Hajivassiliou, and Keane (GHK) perform well. For example, Hajivassiliou et al. (1996) compare 13 simulators using 11 different simulation methods and conclude that the GHK simulation method is most reliable.

For model identification of the multinomial probit model, one of diagonal elements (σ_k=1) of Σ is normalized to 1 and it is assumed that $\sigma_{jk} = \sigma_{kj} = 0 (j = 1, \cdots, J)$. 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 $\rho_{jk} = \frac{\sigma_{jk}}{\sigma_j \sigma_k}$. Therefore, the error covariance matrix can be denoted as $\Sigma = DRD$. To compute the probability of the multivariate normal distribution, the recursive simulation method is used. Refer to Hajivassiliou (1993) for more details on GHK simulators.

Nested Logit

When a decision maker has an n-dimensional choice set $C = C_1 \times C_2 \times \cdots C_n$, multinomial discrete choice models do not produce correct predictions, since they are affected by the undesirable IIA property that is described in the preceding "Simple Multinomial and Conditional Logit" section and derives from the assumption of independence of the utility across choices. The nested logit model overcomes this problem by assuming that decisions are taken sequentially following a decision tree. The assumption of independence of choices is retained only at each single node, or decision step, of the tree.

Nested logit models can be described analytically following the notation of McFadden (1981). Assume that there are L levels, with 1 representing the lowest branch of the tree and L denoting the top level of the tree structure. The index of a node at level h is represented by $(j_h, \cdots, j_L)$. Let π_h denote a node at level

$h+1$, where $\pi_h = (j_{h+1}, \cdots, j_L)$. The choice set C_{π_h} contains choices that belong to branches below the node π_h. The notation C_{π_h} can also be used to express a set of indices below π_h. Note that C_{π_0} is a set with a single element while C_{π_L} represents a choice set containing all possible alternatives. As example, consider the circled node at level 1 in Figure 15.25. Since it stems from node 11, $\pi_{h+1} = 11$, and, since it is the second node stemming from 11, $j_h = 2$, so that its index is $\pi_h = (j_h, \pi_{h+1}) = 211$. Similarly, $C_{11} = \{111, 211, 311\}$ contains all the possible choices below 11.

It may be worth noting that, while this notation is useful for writing closed form solutions for probabilities, the MDC procedure allows a more flexible definition of indeces. See the NEST statement in the "Syntax" section for more details on how to describe decision trees within the MDC procedure.

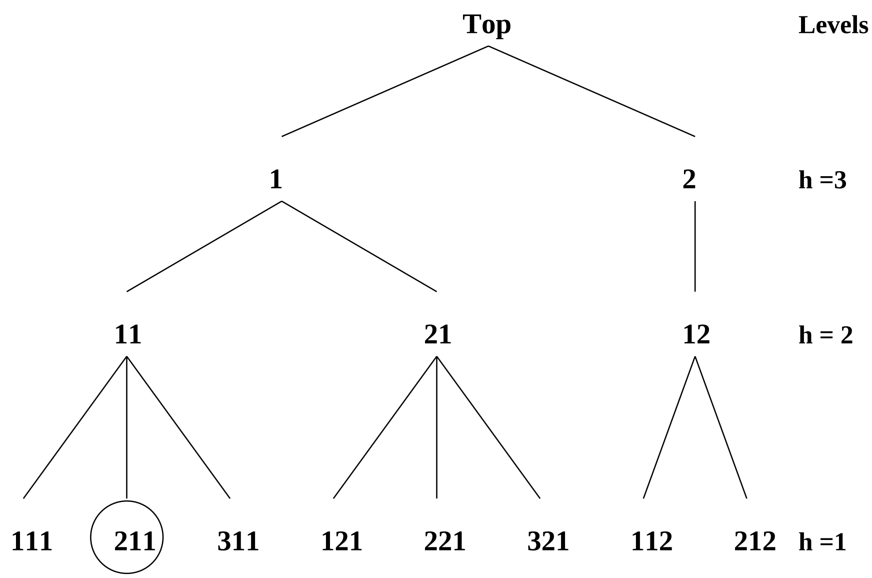


Figure 15.25. Node Indices for a Three-Level Tree

The probability of choice at level h has a closed form solution and is written

$$P_i(j_h|\pi_h) = \frac{\exp[\mathbf{x}_{i,j_h\pi_h}^{h'}\boldsymbol{\beta}^h + \sum_{k\in C_{j_h\pi_h}} I_{k,j_h\pi_h}\boldsymbol{\theta}_{k,j_h\pi_h}]}{\sum_{j\in C_{\pi_h}} \exp[\mathbf{x}_{i,j\pi_h}^{h'}\boldsymbol{\beta}^h + \sum_{k\in C_{j_h\pi_h}} I_{k,j\pi_h}\theta_{k,j\pi_h}]}, \quad h = 2, \cdots, L$$

where I_{π_h} is the *inclusive value* at level $h+1$ and is defined recursively as follows:

$$I_{\pi_h} = \ln \sum_{j\in C_{\pi_h}} \exp[\mathbf{x}_{i,j\pi_h}^{h'}\boldsymbol{\beta}^h + \sum_{k\in C_{j_h\pi_h}} I_{k,j\pi_h}\boldsymbol{\theta}_{k,j\pi_h}]$$

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

The dissimilarity parameters ($\theta_{k,j\pi_h}$) 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, i.e. $I_{\pi_0} = 0$. Therefore, the conditional probability is

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

Let $\mathbf{x}_{i,\pi_{h-1}}^h$ be the vector of variables for the subject (individual) i related to the node π_{h-1} and $n_{i,\pi_{h-1}}$ denote the number of elements of choice set $C_{i,\pi_{h-1}}$. The log-likelihood function at level h can be written

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

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:

$$\ell = \sum_{h=1}^{L} \ell^h$$

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

Decision Tree and Nested Logit

You can view choices as a decision tree and model it using the nested logit model. You need to use the 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. As an example, think of a two-level decision tree. The tree structure is displayed in Figure 15.26.

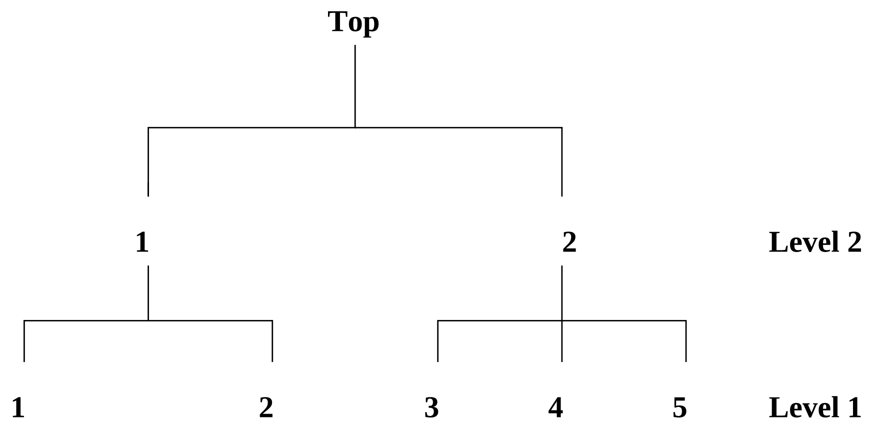


Figure 15.26. Two-Level Decision Tree

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

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

All model variables, x1 –x5, are specified in the utility statement. It is required that entries denoted as # have values for model estimation. Entries denoted as 0 are not used even though there are non-zero values. The choice variable for level 2 (upmode) should be placed before the first-level choice variable (mode) when the CHOICE= option is given. Alternatively, the NEST statement can be used to specify the decision tree. The following SAS statement is used to fit a 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 UTILITY option, U(1, 3 4 5 @ 2), 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 any level share the same variables, you can omit the second argument of the U()= option. However, U(1,) = x1 x2 is not equivalent to

```
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, the stated preference data are widely used. The stated preference data are attractive for market researchers since attribute variations can be controlled. Hensher (1993) explored the advantage of combining revealed preference (market data) and stated preference data. The scale factor $(\mathrm{V}_{rp}/\mathrm{V}_{sp})$ can be estimated using the nested logit model with the decision tree structure displayed in Figure 15.27.

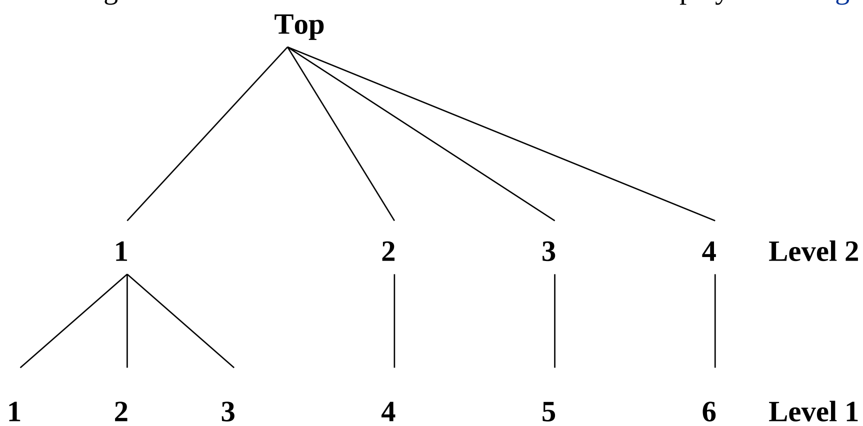


Figure 15.27. Decision Tree for Revealed and Stated Preference Data

Example SAS statements read

```
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 2, 3, and 4 at level 2 are the same. When you specify the SAMESCALE option, the MDC procedure imposes the same coefficient of inclusive values for choices 1–4.

Goodness-of-Fit Measures

McFadden (1974) suggested a likelihood ratio index that is analogous to the R^2 in the linear regression model.

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

where L is the value of the maximum likelihood function at the maximum and L_0 is a likelihood function when regression 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

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

Estrella suggests an alternative measure

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

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:

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

OUTEST= Data Set

The OUTEST= data set contains all the parameters estimated in any MODEL statement. The OUTEST= option can be used when PROC MDC 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 (e.g., SCALE1 or STD1) are generated automatically in the MDC procedure. Therefore, the following SAS statement is not valid when the OUTEST= option is given:

```
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 provided.

```
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 on possible choices is required. Therefore, the following is a correct SAS statement:

```
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 given.

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 that is specified in the METHOD= option. Since only METHOD=ML is supported, _METHOD_ always contains ML.
MODEL	the label of the MODEL statement if one is given, or blank otherwise
STATUS	convergence status for optimization
NAME	the name of the row of the covariance matrix for the parameter estimate, if the COVOUT option is specified
LIKL	the log-likelihood value
STDERR	standard error of the parameter estimate, if the COVOUT option is specified

TYPE PARMS for observations containing parameter estimates, or COV for observations containing 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 of parameter estimates matrix. 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 following table.

Table 15.2. ODS Tables Produced in PROC MDC

ODS Table Name	Description	Option
	ODS Tables Created by the Model Statement	
FitSummary	Summary of Nonlinear Estimation	default
ResponseProfile	Response Profile	default
GoodnessOfFit	Pseudo-R^2 Measures	default
ParameterEstimates	Parameter Estimates	default
CovB	Covariance of Parameter Estimates	COVB
CorrB	Correlation of Parameter Estimates	CORRB

Examples

Example 15.1. Binary Data Modeling

The MDC procedure supports various multinomial choice models. However, binary choice models such as binary logit and probit can also be estimated using PROC MDC since 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), knowledge of the material (tuce), a dummy variable for the new teaching method (psi), and the final course grade (grade). grade is recorded as 1 if a student earned a letter grade "A"; otherwise, 0.

The binary logit can be estimated using the conditional logit model. In order to use the MDC procedure, the data are converted 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
4.00      21      0        1
2.86      17      0        0
2.76      17      0        0
2.87      21      0        0
3.03      25      0        0
3.92      29      0        1
2.63      20      0        0
3.32      23      0        0
3.57      23      0        0
3.26      25      0        1
3.53      26      0        0
2.74      19      0        0
2.75      25      0        0
2.83      19      0        0
3.12      23      1        0
3.16      25      1        1
2.06      22      1        0
3.62      28      1        1
2.89      14      1        0
3.51      26      1        0
3.54      24      1        1
2.83      27      1        1
3.39      17      1        1
2.67      24      1        0
3.65      21      1        1
4.00      23      1        1
3.10      21      1        0
2.39      19      1        1
```

```
;

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 15.1.1. The variables related to grade=0 are omitted since these are not used for binary choice model estimation.

Output 15.1.1. 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 this example, it is assumed that $\mathbf{x}_{i1} = \mathbf{0}$. The binary logit model is estimated and displayed in Output 15.1.2.

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

Output 15.1.2. Binary Logit Estimates

The MDC Procedure

Conditional Logit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
choice2	1	-13.0213	4.9313	-2.64	0.0083	-6.26E-8
gpa_2	1	2.8261	1.2629	2.24	0.0252	-1.62E-7
tuce_2	1	0.0952	0.1416	0.67	0.5014	-1.15E-6
psi_2	1	2.3787	1.0646	2.23	0.0255	-1.55E-9

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 probability of choosing an alternative between 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 15.1.3. You do not get the same estimates as that of the usual binary probit model. The probabilities 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 $Var(\epsilon_{i2} - \epsilon_{i1}) = \sigma_1^2 + \sigma_2^2$ if ϵ_{i1} and ϵ_{i2} are independent ($\sigma_{12} = 0$). In this example, unit variance restrictions are imposed on choices 1 and 2. Therefore, the binary probit estimate can be obtained by multiplying the multinomial probit estimate by $1/\sqrt{2}$.

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

Output 15.1.3. Binary Probit Estimates

The MDC Procedure

Multinomial Probit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
choice2	1	-10.5392	3.5971	-2.93	0.0034	4.102E-6
gpa_2	1	2.2992	0.9815	2.34	0.0191	0.000011
tuce_2	1	0.0732	0.1187	0.62	0.5376	0.00015
psi_2	1	2.0171	0.8415	2.40	0.0165	-5.88E-6

Example 15.2. Conditional Logit and Data Conversion

In this example, a data preparation macro is introduced. 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 (or individual-choice) combination. Simple binary choice data are obtained from Ben-Akiva and Lerman (1985).

```
data travel;
   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
18.5  84.0 Auto
82.0  38.0 Auto
 8.6  1.6  Transit
22.5  74.1 Auto
51.4  83.8 Auto
81.0  19.2 Transit
51.0  85.0 Auto
62.2  90.1 Auto
95.1  22.2 Transit
```

```
41.6  91.5 Auto
;
```

The travel time is stored in two variables, auto and transit. In addition, alternatives are stored in a character variable, mode. The choice variable needs to be converted as a numeric variable since the MDC procedure only supports numeric variables. The original data travel is converted and estimates are displayed in Output 15.2.1.

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

Output 15.2.1. 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|	Gradient
autodum	1	-0.2376	0.7505	-0.32	0.7516	-234E-14
ttime	1	-0.0531	0.0206	-2.57	0.0101	-259E-12

In order to handle more general cases, you can use the data conversion macro program, mdcdata. The mdcdata macro generates choice specific dummy variables and creates multiple observations for each individual. The two variables, auto and transit, are reduced to one variable x_1, but %mdcdata generates extra observations that correspond to the eliminated variable.

```
/*--------------------------------------------------------------
 * name: %mdcdata
 * note: original data contains one observation for each
 *       decision maker (or id). The choice specific data are
 *       stored in multiple variables. The created variables are
 *       x_1 ... x_n. Choice dummy variables are created as
 *       dum_1 ... dum_J. In addition, the id_var, alt_var,
 *       and dec_var are created.
 * purpose: prepare data to use in the mdc procedure
 *-------------------------------------------------------------*/

%macro mdcdata(indata=,outdata=,varlist=,dumlist=,
               altset=,nchoice=0,nvar=0,select=,
               id=id_var,choice=alt_var,decision=dec_var);

%if &indata = or
    &outdata = or
        &select = %then %do;
        %put ERROR: Not enough information is supplied.;
        %put MDCDATA macro cannot be processed.;
        %goto end_macro;
%end;

data &outdata(drop=&varlist _i _j _chk _id _chosen _decide);
set &indata;
array _var{&nvar,&nchoice} &varlist ;
array _newvar{&nvar} x_1 - x_%left(&nvar) ;
array _dum{&nchoice} %if &dumlist = %then %do;
                        dum_1 - dum_%left(&nchoice)
                                             %end;
                                             %else %do;
                        &dumlist
                                             %end; ;
array _set{&nchoice} $ _temporary_ (
                     %do i = 1 %to &nchoice;
                         "%scan(&altset,&i)"
                     %end; ) ;
_chk = 0;
_id = _n_;
do _i = 1 to &nchoice;
   %do j = 1 %to &nvar;
          _newvar{&j} = _var{&j,_i};
       %end;
   %if &id = %then %do;
           id_var = _id;
       %end;
       %else %do;
           &id = _id;
       %end;
   %if &choice = %then %do;
           alt_var = _i;
       %end;
       %else %do;
          &choice = _i;
```

```
          %end;
          /*-- choice variable is numeric --*/
          %if &altset = %then %do;
             _decide = ( &select = i );
                 if ( _decide ) then _chk = _chk + 1;
          %if &decision = %then %do;
                 dec_var = _decide;
             %end;
             %else %do;
                &decision = _decide;
             %end;
          %end;
          /*-- choice variable is alphanumeric --*/
          %else %do;
             _chosen = 0;
             do _j = 1 to &nchoice;
                if ( upcase(_set{_j}) = upcase(&select) ) then
               _chosen = _j;
             end;
             if ( _chosen = 0 ) then
                _decide = 0;
             else _decide = ( _i = _chosen );
             if ( _decide ) then _chk = _chk + 1;
             %if &decision = %then %do;
                 dec_var = _decide;
             %end;
             %else %do;
                 &decision = _decide;
             %end;
          %end;
      do _j = 1 to &nchoice;
         if ( _i = _j ) then
            _dum{_j} = 1;
         else _dum{_j} = 0;
      end;
      output;
   end;
   /*- check if any decision is not made -*/
   if ( _chk ^= 1 ) then
      put "WARNING: No choices are given for id =" _id ;
   run;
   %end_macro:
   %mend mdcdata;
```

The following macro invocation produces a new data set. Its first 10 observations are displayed in Output 15.2.2.

```
%mdcdata(indata=travel,outdata=base,
         varlist=auto transit,
         dumlist=autodum trdum,nchoice=2,nvar=1,
         altset=%str(auto transit),
         select=mode)
```

Output 15.2.2. Converted Data Using MDCDATA Macro

id_var	alt_var	dec_var	autodum	x_1
1	1	0	1	52.9
1	2	1	0	4.4
2	1	0	1	4.1
2	2	1	0	28.5
3	1	1	1	4.1
3	2	0	0	86.9
4	1	0	1	56.2
4	2	1	0	31.6
5	1	0	1	51.8
5	2	1	0	20.2

The converted variable is created as x_1, . . ., x_n. When you do not specify the ID statement, CHOICE=, or DECISION= option, id_var, alt_var, and dec_var are automatically produced. Finally, the conditional logit model is estimated. Estimates displayed in Output 15.2.3 are equivalent to Output 15.2.1.

```
proc mdc data=new type=clogit;
   model dec_var = autodum x_1 / nchoice=2;
   id id_var;
run;
```

Output 15.2.3. Conditional Logit Estimates

The MDC Procedure

Conditional Logit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
autodum	1	-0.2376	0.7505	-0.32	0.7516	-234E-14
x_1	1	-0.0531	0.0206	-2.57	0.0101	-259E-12

Example 15.3. Correlated Choice Modeling

It is not realistic to assume that all choices are independent. To analyze correlated data, trinomial choice data (1000 observations) is created using the pseudo-random number generator. The random utility function is

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

where

$$\epsilon_{ij} \sim N\left(0, \begin{bmatrix} .64 & .12 & 0 \\ .12 & .36 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

```
%let ndim = 3;
%let nobs = 1000;

 /*-- generate simulated series --*/
   data one;
      array error{&ndim} e1-e3;
      array vtemp{&ndim} _temporary_;
      array lm{6} _temporary_ (.8 0.15 .5809475 0 0 1);
      retain nseed 345678 useed 223344;

      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 * ranuni(useed);
         x2 = 1.2 + 2 * ranuni(useed);
         x3 = 1.5 + 1.2 * ranuni(useed);
         util1 = 2 * x1 + e1;
         util2 = 2 * x2 + e2;
         util3 = 2 * 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. Results show that standard deviation and correlation estimates are close to parameter values. Note that $\rho_{12} = \frac{.12}{\sqrt{(.64)(.36)}} = .25$. See Output 15.3.1.

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

Output 15.3.1. Trinomial Probit Model Estimation

The MDC Procedure

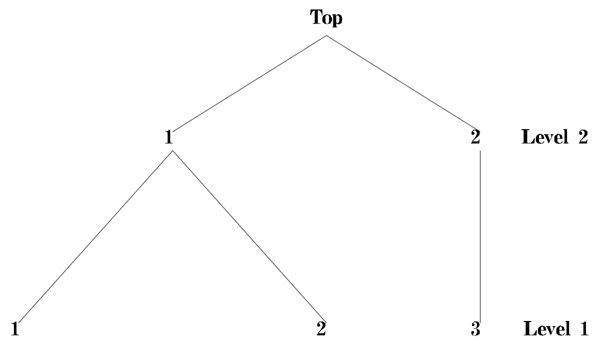
Multinomial Probit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
x	1	1.8764	0.1125	16.67	<.0001	-0.00001
STD_1	1	0.8017	0.0771	10.40	<.0001	0.000086
STD_2	1	0.6317	0.0896	7.05	<.0001	-0.00001
RHO_21	1	0.2680	0.1011	2.65	0.0080	-0.00005

The nested model is also estimated based on a two-level decision tree. See Output 15.3.2. The estimated result shows that the data supports the nested tree model since the inclusive parameter estimates are significant and are less than 1.

Output 15.3.2. Nested Tree Structure




```
proc mdc data=one;
   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;
```

Output 15.3.3. Two-Level Nested Logit

The MDC Procedure

Nested Logit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
x_L1	1	3.9586	0.3232	12.25	<.0001	0.000066
INC_L2G1C1	1	0.6834	0.0748	9.14	<.0001	0.009671
INC_L2G1C2	1	0.6903	0.0799	8.64	<.0001	-0.00856

Example 15.4. Testing for Homoscedastic Utility Function

A HEV model is analyzed in the “Getting Started” section using Daganzo’s trinomial choice data. The HEV estimates are displayed in Figure 15.14. The inverted scale estimates for mode “2” and mode “3” show that the conditional logit might be misleading. The HEV estimation summary is shown in Output 15.4.1.

Output 15.4.1. HEV Estimation Summary ($\theta_1 = 1$)

```
                          The MDC Procedure

             Heteroscedastic Extreme Value Model Estimates

                          Model Fit Summary

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

The HEV model with unit scale restrictions is estimated, and the estimation summary is displayed in Output 15.4.2.

Output 15.4.2. 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             6.79509E-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 statement computes the test statistic (1.4276) and and its *p* value (.4898):

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

However, the test statistic displayed in Output 15.4.1 and Output 15.4.2 fails to reject the null hypothesis of equal scale parameters, which implies that the random utility function is homoscedastic.

The heteroscedastic utility function can also be analyzed using multinomial probit. Consider the following utility function:

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

where

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

A multinomial probit model is estimated and the estimation summary is displayed in Output 15.4.3.

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

Output 15.4.3. 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
              Maximum Absolute Gradient              5.60275E-6
              Number of Iterations                            8
              Optimization Method             Dual Quasi-Newton
              AIC                                      71.77209
              Schwarz Criterion                        75.59613
```

The multinomial probit model with unit variances ($\sigma_1 = \sigma_2 = \sigma_3 = 1$) is estimated and an estimation summary is given in Output 15.4.4.

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

Output 15.4.4. 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
           Maximum Absolute Gradient               1.37303E-7
           Number of Iterations                             5
           Optimization Method                Dual Quasi-Newton
           AIC                                       71.08505
           Schwarz Criterion                         72.99707
```

The test for homoscedasticity (σ_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 $\chi^2_{.05,1} = 3.84$. The marginal probability or *p* value computed from the PROBCHI function is $.2518$.

```
data _null_;
   /*-- test for H0: sigma3 = 1 --*/
   /*   log L(max) = -33.88545    */
   /*   log L(0)   = -34.54448    */
   stat = -2 * ( -34.54448 + 33.88545 );
   df = 1;
   p_value = 1 - probchi(stat, df);
   put stat p_value;
run;
```

Example 15.5. Choice of Time for Work Trips: Nested Logit Analysis

A sample data of 527 automobile commuters in San Francisco Bay Area has been analyzed by Small (1982) and 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. Refer to Small (1982) for more details on this data.

Brownstone and Small (1989) analyzed a two-level nested logit model displayed in Output 15.5.1. The probability of choosing j at level 2 is written

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

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

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

The probability of choosing an alternative k is denoted as

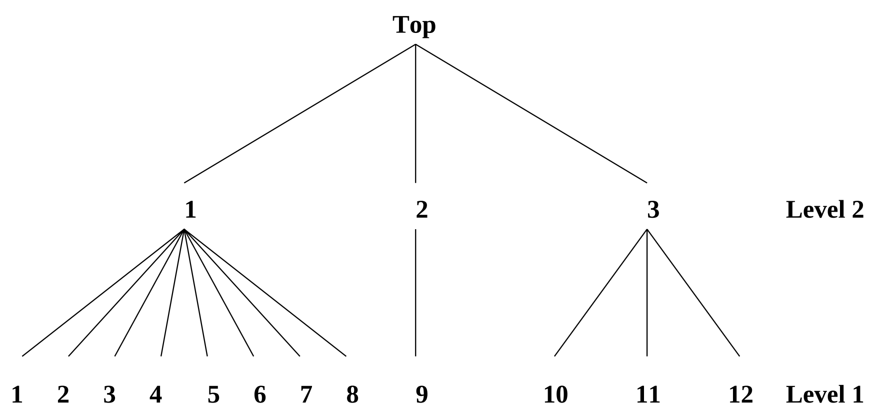
$$P_i(k|j) = \frac{\exp(\mathbf{x}'_{ik}\boldsymbol{\beta})}{\sum_{k' \in C_j} \exp(\mathbf{x}'_{ik'}\boldsymbol{\beta})}$$

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

$$\ell = \sum_{i=1}^{N} \sum_{j=1}^{J} d_{ij} \left[\log(P_i(k|j)) + \log(P_i(j))\right]$$

where $d_{ij} = 1$ if a decision maker i chooses j; otherwise, $d_{ij} = 0$. The FIML estimates are given in Output 15.5.4.

Output 15.5.1. Decision Tree for Two-Level Nested Logit




```
proc mdc data=mylib.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;
```

Output 15.5.2. 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
Maximum Absolute Gradient             4.93868E-6
Number of Iterations                          18
Optimization Method               Newton-Raphson
AIC                                         2006
Schwarz Criterion                           2057
```

Output 15.5.3. Discrete Choice Characteristics

```
                    The MDC Procedure

                  Nested Logit Estimates

                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 15.5.4. Nested Logit Estimates

```
                              The MDC Procedure

                            Nested Logit Estimates

                             Parameter Estimates

                                       Standard                  Approx
Parameter       DF      Estimate          Error    t Value     Pr > |t|     Gradient

r15_L1           1        1.1034         0.1221       9.04       <.0001     4.196E-8
r10_L1           1        0.3931         0.1194       3.29       0.0010     3.427E-8
ttime_L1         1       -0.0465         0.0235      -1.98       0.0474     -1.06E-7
ttime_cp_L1      1       -0.0498         0.0305      -1.63       0.1028      2.42E-8
sde_L1           1       -0.6618         0.0833      -7.95       <.0001     1.257E-7
sde_cp_L1        1        0.0519         0.1278       0.41       0.6850     -4.63E-9
sdl_L1           1       -2.1006         0.5062      -4.15       <.0001      -5.8E-7
sdlx_L1          1       -3.5240         1.5346      -2.30       0.0217     -4.94E-6
d2l_L1           1       -1.0941         0.3273      -3.34       0.0008     -3.74E-7
INC_L2G1C1       1        0.6762         0.2754       2.46       0.0141     -2.59E-7
INC_L2G1C2       1        1.0906         0.3090       3.53       0.0004     -2.34E-7
INC_L2G1C3       1        0.7622         0.1649       4.62       <.0001     -1.74E-9
```

Brownstone and Small (1989) estimated the two-level nested logit model with equal scale parameter constraints, $\tau_1 = \tau_2 = \tau_3$. Replication of their model results is displayed in Output 15.5.5 and Output 15.5.6. The parameter estimates and standard errors are almost identical to those in Brownstone and Small (1989, p. 69).

```
proc mdc data=mylib.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;
```

Output 15.5.5. Nested Logit Estimation Summary with Equal Dissimilar 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
                  Maximum Absolute Gradient          2.97172E-6
                  Number of Iterations                       16
                  Optimization Method            Newton-Raphson
                  AIC                                      2009
                  Schwarz Criterion                        2051
```

Output 15.5.6. Nested Logit Estimates with Equal Dissimilar Parameters

The MDC Procedure

Nested Logit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Gradient
r15_L1	1	1.1345	0.1092	10.39	<.0001	2.318E-8
r10_L1	1	0.4194	0.1081	3.88	0.0001	1.836E-8
ttime_L1	1	-0.1626	0.0609	-2.67	0.0076	-3.04E-8
ttime_cp_L1	1	0.1285	0.0853	1.51	0.1319	3.264E-8
sde_L1	1	-0.7548	0.0669	-11.28	<.0001	4.637E-8
sde_cp_L1	1	0.2292	0.0981	2.34	0.0195	8.206E-9
sdl_L1	1	-2.0719	0.4860	-4.26	<.0001	-3.34E-7
sdlx_L1	1	-2.8216	1.2560	-2.25	0.0247	-2.97E-6
d2l_L1	1	-1.3164	0.3474	-3.79	0.0002	-3.93E-7
INC_L2G1	1	0.8059	0.1705	4.73	<.0001	-1.35E-7

However, the test statistic for $H_0 : \tau_1 = \tau_2 = \tau_3$ rejects the null hypothesis at 5% significance level since the $-2 * (\log L(0) - \log L)(7.15) > \chi^2_2(.05)(5.99)$. The *p* value is computed as $.0280$.

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

Acknowledgments

Professor Kenneth Small has provided the work trip data that is used in the "Examples" section. This data was collected for the urban travel demand forecasting project, which was carried out by McFadden et al. (1977). The project was supported by the National Science Foundation, Research Applied to National Needs Program through grants GI-43740 and APR74-20392, and Alfred P. Sloan Foundation, through grant 74-21-8.

References

Abramowitz, M. and Stegun, A. (1970), *Handbook of Mathematical Functions*, New York: Dover Press.

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

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

Ben-Akiva, M. and Lerman, S.R. (1985), *Discrete Choice Analysis*, Cambridge: 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 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.

Geweke, J., Keane, M., 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.

Green, W.H. (1997), *Econometric Analysis*, Upper Saddle River, N.J.: Prentice Hall.

Hajivassiliou, V.A. (1993), "Simulation Estimation Methods for Limited Dependent Variable Models," in *Handbook of Statistics*, vol. 11, ed. G.S. Maddala, C.R. Rao, and H.D. Vinod, New York: Elseviour Science Publishing.

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

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

Luce, R.D. (1959), *Individual Choice Behaviour*: *A Theoretical Analysis*, New York: Wiley.

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

McFadden, D. (1974), "Conditional Logit Analysis of Qualitative Choice Behaviour," in *Frontiers in Econometrics*, ed. P. Zarembka, New York: Academic Press.

McFadden, D. (1981), "Econometric Models of Probabilistic Choice," in *Structural Analysis of Discrete Data with Econometric Applications*, ed. C.F. Manski and D. McFadden, Cambridge: MIT Press.

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

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

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

Schmidt, P. and Strauss, R. (1975), "Estimation of Models with Jointly Dependent Qualitative Variables: A Simultaneous Logit Approach," *Econometrica*, 43, 745–755.

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), "Distingushing Taste Variation from Error Structure in Discrete Choice Data," *Transportation Research Part B*, 34, 1–15.

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