NLOGIT

Version 4.0

Student Reference Guide

by

William H. Greene
Econometric Software, Inc.

© 1986 - 2010 Econometric Software, Inc. All rights reserved.

This software product, including both the program code and the accompanying documentation, is copyrighted by, and all rights are reserved by Econometric Software, Inc. No part of this product, either the software or the documentation, may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without prior written permission of Econometric Software, Inc.

LIMDEP® and *NLOGIT*™® are registered trademarks of Econometric Software, Inc. All other brand and product names are trademarks or registered trademarks of their respective companies.

Econometric Software, Inc.
15 Gloria Place
Plainview, NY 11803
USA
Tel: +1 516-938-5254
Fax: +1 516-938-2441
Email: sales@limdep.com
Websites: www.limdep.com and www.nlogit.com

Econometric Software, Australia
215 Excelsior Avenue
Castle Hill, NSW 2154
Australia
Tel: +61 (0)4-1843-3057
Fax: +61 (0)2-9899-6674
Email: hgroup@optusnet.com.au

End-User License Agreement

This is a contract between you and Econometric Software, Inc. The *software product* refers to the computer software and documentation as well as any upgrades, modified versions, copies or supplements supplied by Econometric Software. By installing, downloading, accessing or otherwise using the software product, you agree to be bound by the terms and conditions of this agreement.

Copyright, Trademark, and Intellectual Property

This software product is copyrighted by, and all rights are reserved by Econometric Software, Inc. No part of this software product, either the software or the documentation, may be reproduced, distributed, downloaded, stored in a retrieval system, transmitted in any form or by any means, sold or transferred without prior written permission of Econometric Software. You may not modify, adapt, translate, or change the software product. You may not reverse engineer, decompile, dissemble, or otherwise attempt to discover the source code of the software product.

LIMDEP™ and *NLOGIT*™ are trademarks of Econometric Software, Inc. The software product is licensed, not sold. Your possession, installation and use of the software product does not transfer to you any title and intellectual property rights, nor does this license grant you any rights in connection with software product trademarks.

Use of the Software Product

You have only the non-exclusive right to use this software product. A single user license is registered to one specific individual, and is not intended for access by multiple users on one machine, or for installation on a network or in a computer laboratory. For a single user license only, the registered single user may install the software on a primary stand alone computer and one home or portable secondary computer for his or her exclusive use. However, the software may not be used on the primary computer by another person while the secondary computer is in use. For a multi-user site license, the specific terms of the site license agreement apply for scope of use and installation.

Limited Warranty

Econometric Software warrants that the software product will perform substantially in accordance with the documentation for a period of ninety (90) days from the date of the original purchase. To make a warranty claim, you must notify Econometric Software in writing within ninety (90) days from the date of the original purchase and return the defective software to Econometric Software. If the software does not perform substantially in accordance with the documentation, the entire liability and your exclusive remedy shall be limited to, at Econometric Software's option, the replacement of the software product or refund of the license fee paid to Econometric Software for the software product. Proof of purchase from an authorized source is required. This limited warranty is void if failure of the software product has resulted from accident, abuse, or misapplication. Some states and jurisdictions do not allow limitations on the duration of an implied warranty, so the above limitation may not apply to you. To the extent permissible, any implied warranties on the software product are limited to ninety (90) days.

Econometric Software does not warrant the performance or results you may obtain by using the software product. To the maximum extent permitted by applicable law, Econometric Software disclaims all other warranties and conditions, either express or implied, including, but not limited to, implied warranties of merchantability, fitness for a particular purpose, title, and non-infringement with respect to the software product. This limited warranty gives you specific legal rights. You may have others, which vary from state to state and jurisdiction to jurisdiction.

Limitation of Liability

Under no circumstances will Econometric Software be liable to you or any other person for any indirect, special, incidental, or consequential damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, computer failure or malfunction, loss of business information, or any other pecuniary loss) arising out of the use or inability to use the software product, even if Econometric Software has been advised of the possibility of such damages. In any case, Econometric Software's entire liability under any provision of this agreement shall not exceed the amount paid to Econometric Software for the software product. Some states or jurisdictions do not allow the exclusion or limitation of liability for incidental or consequential damages, so the above limitation may not apply to you.

Preface

NLOGIT is a major suite of programs for the estimation of discrete choice models. It is built on the original **DISCRETE CHOICE** command in *LIMDEP* Version 6.0 which provided some of the features that are described with the estimator presented in Chapter N13 of this reference guide. *NLOGIT*, itself, began in 1996 with the development of the nested logit command, originally an extension of the multinomial logit model. With the additions of the multinomial probit model and the mixed logit model among several others, *NLOGIT* has now grown to a self standing superset of *LIMDEP*. The focus of most of the recent development is the random parameters logit model, or 'mixed logit' model as it is frequently called in the literature. *NLOGIT* is now the only generally available package that contains panel data (repeated measures) versions of this model, in random effects and autoregressive forms. We note, the technology used in the random parameters model, originally proposed by Dan McFadden and Kenneth Train, has proved so versatile and robust, that we have been able to extend it into most of the other modeling platforms that are contained in *LIMDEP*. They, like *NLOGIT*, now contain random parameters versions. Finally, a major feature of *NLOGIT* is the simulation package. With this program, you can use any model that you have estimated to do 'what if' sorts of simulations to examine the effects on predicted behavior of changes in the attributes of choices in your model.

NLOGIT Version 4.0 is the result of an ongoing (since 1985) collaboration of William Greene (Econometric Software, Inc.) and David Hensher (Econometric Software, Australia.) Recent developments, especially the random parameters logit in its cross section and panel data variants have also benefited from the suggestions of Kenneth Train of UC Berkeley. Version 4.0 has also been greatly improved by the enthusiastic collaboration of John Rose (Econometric Software, Australia).

We note, the recently published work *Applied Choice Analysis: A Primer* (Hensher, D., Rose, J. and Greene, W., Cambridge University Press, 2005) is a wide ranging introduction to discrete choice modeling that contains numerous applications developed with Versions 3.0 and 4.0 of *NLOGIT*. This book should provide a useful companion to the documentation for *NLOGIT*.

William H. Greene
Econometric Software, Inc.
15 Gloria Place
Plainview, NY 11803
January 2007

Table of Contents

Table of Contents vi

Chapter 1: Introduction to *NLOGIT* 9
1.1 Discrete Choice Modeling with *NLOGIT* 9
1.2 *NLOGIT* and *LIMDEP* 10

Chapter 2: Discrete Choice Models 11
2.1 Introduction 11
2.2 Random Utility Models 11
2.3 Binary Choice Models 12
2.4 Multinomial Logit Model 14
2.5 Conditional Logit Model 16
2.6 Nested Logit Model 17
2.7 Random Parameters Logit Models 18
2.8 Multinomial Probit Model 20

Chapter 3: Model and Command Summary for Discrete Choice Models 22
3.1 Introduction 22
3.2 Model Summary 22
3.3 Basic Discrete Choice Models 22
3.4 Multinomial Logit Models 23
3.4.1 Multinomial Logit 23
3.4.2 Conditional Logit 23
3.5 NLOGIT Extensions of Conditional Logit 24
3.5.1 Nested 24
3.5.2 Random Parameters Logit 24
3.5.3 Multinomial Probit 25
3.6 Command Summary 25
3.7 Subcommand Summary 26

Chapter 4: The Basic Multinomial Logit Model 30
4.1 Introduction 30
4.2 The Multinomial Logit Model 31
4.3 Model Command for the Multinomial Logit Model 32
4.4 Robust Covariance Matrices 32
4.5 Output for the Multinomial Logit Model 34
4.6 Marginal Effects 37
4.7 Computing Predicted Probabilities 40

Chapter 5: Data Setup for *NLOGIT* 41
5.1 Introduction 41
5.2 Basic Data Setup for *NLOGIT* 41
5.3 Fixed and Variable Numbers of Choices 42
5.4 Types of Data on the Choice Variable 46
5.5 Data for the Applications 47

Chapter 6: *NLOGIT* Commands and Results 49
6.1 Introduction 49
6.2 *NLOGIT* Commands 49
6.2.1 Other Optional Specifications on *NLOGIT* Commands 52
6.2.2 Specifying the Choice Variable and the Choice Set 53
6.2.3 Restricting the Choice Set 54
6.2.4 Specifying the Utility Functions with Rhs and Rh2 55
6.2.5 Building the Utility Functions 59
6.3 Standard Model Results 67
6.3.1 Retained Results 69
6.3.2 Robust Standard Errors 70
6.3.3 Descriptive Statistics for Alternatives 71
6.4 Marginal Effects and Elasticities 72
6.5 Predicted Probabilities and Inclusive Values 76
6.5.1 In Sample Predicted Probabilities and Inclusive Values 76
6.5.2 Computing Out of Sample Model Probabilities 78
6.6 Testing Hypotheses 80
6.6.1 Testing the Assumption of Independence from Irrelevant Alternatives (IIA) 80
6.6.2 Lagrange Multiplier, Wald, and Likelihood Ratio Tests 83

Chapter 7: Simulating Probabilities in Discrete Choice Models 84
7.1 Introduction 84
7.2 Essential Subcommands 85
7.3 Multiple Attribute Specifications and Multiple Scenarios 86
7.4 Simulation Commands 87
7.4.1 Observations Used for the Simulations 87
7.4.2 Variables Used for the Simulations 87
7.4.3 Choices Simulated 87
7.4.4 Other *NLOGIT* Options 87
7.4.5 Observations Used for the Simulations 87
7.5 Applications 88

Chapter 8: The Multinomial Logit Model 94
8.1 Introduction 94
8.2 Command for the Multinomial Logit Model 95
8.3 Results for the Multinomial Logit Model 97
8.4 Application 97
8.5 Marginal Effects 101

Chapter 9: The Nested Logit Model 103
9.1 Introduction 103
9.2 Mathematical Specification of the Model 104
9.3 Commands for FIML Estimation 105
9.3.1 Data Setup 105
9.3.2 Tree Definition 106
9.3.3 Utility Functions 107
9.3.4 Setting and Constraining Inclusive Value Parameters 108

9.3.5 Command Builder 110
9.4 Marginal Effects and Elasticities 112
9.5 Inclusive Values, Utilities, and Probabilities 114
9.6 Application of a Nested Logit Model 115

Chapter 10: The Random Parameters Logit Model 120
10.1 Introduction 120
10.2 Random Parameters (Mixed) Logit Models 120
10.3 Command for the Random Parameters Logit Models 124
10.3.1 Distributions of Random Parameters in the Model 125
10.3.2 Spreads, Scaling Parameters and Standard Deviations 128
10.3.3 Alternative Specific Constants 132
10.3.4 Heterogeneity in the Means of the Random Parameters 132
10.3.5 Correlated Parameters 133
10.3.6 Command Builders for the RPL Models 135
10.4 Heteroscedasticity and Heterogeneity in the Variances 137
10.5 Controlling the Simulations 138
10.5.1 Number and Initiation of the Random Draws 138
10.5.2 Halton Draws and Random Draws for Simulations 139
10.6 Model Estimates 139
10.7 Individual Specific Estimates 143
10.7.1 Computing Individual Specific Parameter Estimates 144
10.7.2 Examining the Distribution of the Parameters 149
10.7.3 Conditional Confidence Intervals for Parameters 153
10.7.4 Willingness to Pay Estimates 155
10.8 Applications 156
10.9 Panel Data 159

Chapter 11: The Multinomial Probit Model 161
11.1 Introduction 161
11.2 Model Command 162
11.3 An Application 164
11.4 Testing IIA with a Multinomial Probit Model 166

Chapter 12: Diagnostics and Error Messages 167
12.1 Introduction 167
12.2 Discrete Choice (CLOGIT) and NLOGIT 168

Chapter 1: Introduction to *NLOGIT*

1.1 Discrete Choice Modeling with *NLOGIT*

NLOGIT is a set of tools for building models of discrete choice among multiple alternatives. The essential building block that underlies the set of programs is the random utility model of consumer choice,

U(choice 1) = f_1 (attributes of choice 1, characteristics of the consumer, ε_1,**v**,**w**)
...
U(choice J) = f_J (attributes of choice J, characteristics of the consumer, ε_J,**v**,**w**)

where the functions on the right hand side describe the utility to a consumer – decision maker – of J possible choices, as functions of the attributes of the choices, the characteristics of the consumer, random choice specific elements of preferences, ε_j, that may be known to the chooser but are unobserved by the analyst and random elements **v** and **w**, that will capture the unobservable heterogeneity across individuals. Finally, a crucial element of the underlying theory is the assumption of utility maximization,

The choice made is alternative j such that U(choice j) > U(choice q) $\forall\ q \neq j$.

The tools provided by *NLOGIT* are a complete suite of estimators beginning with the simplest binary logit model for choice between two alternatives and progressing through the most recently developed models for multiple choices, including random parameters, mixed logit models with individual specific random effects for repeated observation choice settings and the multinomial probit model.

Background theory and applications for the programs described here can be found in many sources. For a primer that develops the theory in detail and presents many examples and applications, all using *NLOGIT*, we suggest

Applied Choice Analysis: A Primer, Hensher, D., Rose, J. and Greene, W., Cambridge University Press, Cambridge, 2005.

It is not possible (nor even desirable) to present all of the necessary econometric methodology in a manual of this sort. The econometric background needed for *Applied Choice Analysis* as well as for use of the tools to be described here can be found in many graduate econometrics books. One popular choice is

Econometric Analysis, 7th Edition, Greene, W., Prentice Hall, Englewood Cliffs, 2011.

Finally, this guide is primarily focused on the specialized tools in *NLOGIT* for extensions of the multinomial logit model. Users will find the *LIMDEP* documentation, the *LIMDEP Reference Guide* and Volumes 1 and 2 of the *LIMDEP Econometric Modeling Guide*, essential for effective use of this program.

It is assumed throughout that you are already a user of *LIMDEP*. The *NLOGIT Reference Guide*, by itself, will not be sufficient documentation for you to use *NLOGIT* unless you are already familiar with the program platform, *LIMDEP*, on which *NLOGIT* is placed.

1.2 *NLOGIT* and *LIMDEP*

This *Reference Guide* describes *NLOGIT* Version 4.0. *NLOGIT* is a suite of programs for estimating discrete choice models that are built around the logit and multinomial logit form. This is a superset of *LIMDEP*'s models – *NLOGIT* 4.0 is all of *LIMDEP* 9.0 plus the set of tools and estimators described in this manual. *LIMDEP* 9.0 contains the **CLOGIT** command and the estimator for the 'conditional logit' (or multinomial logit) model. **CLOGIT** is the same as the most basic form of the **NLOGIT** command described in Chapter 6. This manual will describe the tools and estimators that extend the multinomial logit model. These include, for example, extensions of the multinomial logit model such as the nested logit, mixed logit and multinomial probit models.

We emphasize, *NLOGIT* Version 4.0 is a superset of *LIMDEP* 9.0. It is created by adding certain features to *LIMDEP* Version 9.0. As such, the full set of features of *LIMDEP* 9.0 is part of this package as well. We assume that you will use the other parts of *LIMDEP* as part of your analysis. More to the point, this manual is primarily oriented to the commands added to *LIMDEP* that request the set of discrete choice estimators.

To use *NLOGIT*, you will need to be familiar with the *LIMDEP* platform. At various points in your operation of the program, you will encounter *LIMDEP*, rather than *NLOGIT* as the program name, for example in certain menus, dialog boxes, window headers, diagnostics, and so on. Once again, these result from the fact that in obtaining *NLOGIT*, you have installed *LIMDEP* plus some additional capabilities. If you are uncertain which program is actually installed on your computer, go to the About box in the main menu. It will clearly indicate which program you are operating.

Chapter 2: Discrete Choice Models

2.1 Introduction

This chapter will provide a short, thumbnail sketch of the discrete choice models discussed in this manual. *NLOGIT* supports a large array of models for both discrete and continuous variables, including regression models, survival models, models for counts and, of relevance to this setting, models for discrete outcomes. The group of models described in this manual are those that arise naturally from a random utility framework, that is, those that arise from a consumer choice setting in which the model is of an individual's selection among two or more alternatives. This includes several of the models described in the *LIMDEP* manual, such as the binary logit and probit models, but also excludes some others, including the models for count data and some of the loglinear models such as the geometric regression model.

2.2 Random Utility Models

The random utility framework starts with a structural model,

$$U(\textit{choice } 1) = f_1\,(\textit{attributes of choice } 1, \textit{characteristics of the consumer}, \varepsilon_1, \mathbf{v}, \mathbf{w}),$$
$$\ldots$$
$$U(\textit{choice } J) = f_J\,(\textit{attributes of choice } J, \textit{characteristics of the consumer}, \varepsilon_J, \mathbf{v}, \mathbf{w}),$$

where $\varepsilon_1,\ldots,\varepsilon_J$ denote the random elements of the random utility functions and in our later treatments, **v** and **w** will represent the unobserved individual heterogeneity built into models such as the error components and random parameters (mixed logit) models. The assumption that the choice made is alternative j such that

$$U(\textit{choice } j) > U(\textit{choice } q)\ \forall\ q \neq j.$$

The observed outcome variable is then

$$y = \text{the index of the observed choice.}$$

The econometric model that describes the determination of y is then built around the assumptions about the random elements in the utility functions that endow the model with its stochastic characteristics. Thus, where Y is the random variable that will be the observed discrete outcome,

$$\text{Prob}(Y = j) = \text{Prob}(U(\text{choice } j) > U(\text{choice } q)\ \forall\ q \neq j).$$

The objects of estimation will be the parameters that are built into the utility functions including possibly those of the distributions of the random components and, with estimates of the parameters in hand, useful characteristics of consumer behavior that can be derived from the model, such as partial effects and measures of aggregate behavior.

To consider the simplest example, that will provide the starting point for our development, consider a consumer's random utility derived over a single choice situation, say whether to make a purchase. The two outcomes are 'make the purchase' and 'do not make the purchase.' The random utility model is simply

$$U(\textit{not purchase}) = \boldsymbol{\beta}_0'\mathbf{x}_0 + \varepsilon_0,$$
$$U(\textit{purchase}) = \boldsymbol{\beta}_1'\mathbf{x}_1 + \varepsilon_1.$$

Assuming that ε_0 and ε_1 are random, the probability that the analyst will observe a purchase is

$$\begin{aligned}\text{Prob}(\textit{purchase}) &= \text{Prob}(U(\textit{purchase}) > U(\textit{not purchase}))\\ &= \text{Prob}(\boldsymbol{\beta}_1'\mathbf{x}_1 + \varepsilon_1 > \boldsymbol{\beta}_0'\mathbf{x}_0 + \varepsilon_0)\\ &= \text{Prob}(\varepsilon_1 - \varepsilon_0 < \boldsymbol{\beta}_1'\mathbf{x}_1 - \boldsymbol{\beta}_0'\mathbf{x}_0)\\ &= F(\boldsymbol{\beta}_1'\mathbf{x}_1 - \boldsymbol{\beta}_0'\mathbf{x}_0),\end{aligned}$$

where $F(z)$ is the CDF of the random variable $\varepsilon_1 - \varepsilon_0$. The model is completed and an estimator, generally maximum likelihood, is implied by an assumption about this probability distribution. For example, if ε_0 and ε_1 are assumed to be normally distributed, then the difference is also, and the familiar probit model emerges.

The sections to follow will outline the models described in this manual in the context of this random utility model. The different models derive from different assumptions about the utility functions and the distributions of their random components.

2.3 Binary Choice Models

Continuing the example in the previous section, the choice of alternative 1 (*purchase*) reveals that $U_1 > U_0$, or that

$$\varepsilon_1 - \varepsilon_0 < \boldsymbol{\beta}_1'\mathbf{x}_1 - \boldsymbol{\beta}_0'\mathbf{x}_0.$$

Let $\varepsilon = \varepsilon_1 - \varepsilon_0$ and $\boldsymbol{\beta}'\mathbf{x}$ represent the difference on the right hand side of the inequality – $\mathbf{x}$ is the union of the two sets of covariates, and $\boldsymbol{\beta}$ is constructed from the two parameter vectors with zeros in the appropriate locations if necessary. Then, a binary choice model applies to the probability that $\varepsilon \le \boldsymbol{\beta}'\mathbf{x}$. Two of the parametric model formulations in *NLOGIT* for binary choice models are the probit model based on the normal distribution:

$$F = \int_{-\infty}^{\beta'\mathbf{x}_i} \frac{\exp(-t^2/2)}{\sqrt{2\pi}}dt = \Phi(\boldsymbol{\beta}'\mathbf{x}_i),$$

and the logit model based on the logistic distribution

$$F = \frac{\exp(\boldsymbol{\beta}'\mathbf{x}_i)}{1+\exp(\boldsymbol{\beta}'\mathbf{x}_i)} = \Lambda(\boldsymbol{\beta}'\mathbf{x}_i).$$

Numerous variations on the model can be obtained. A model with multiplicative heteroscedasticity is obtained with the additional assumption

$$\varepsilon_i \sim \text{ normal or logistic with variance} \propto [\exp(\boldsymbol{\gamma}'\mathbf{z}_i)]^2,$$

where $\mathbf{z}_i$ is a set of observed characteristics of the individual. A model of sample selection can be extended to the probit and logit binary choice models. In both cases, we depart from

$$\text{Prob}(y_i = 1 \,|\mathbf{x}_i) = F(\boldsymbol{\beta}'\mathbf{x}_i),$$

where $F(t) = \Phi(t)$ for the probit model and $\Lambda(t)$ for the logit model,

$$d_i^* = \boldsymbol{\alpha}'\mathbf{z}_i + u_i,\ u_i \sim N[0,1],\ d_i = 1(d_i^* > 0),$$

y_i, $\mathbf{x}_i$ observed only when $d_i = 1$.

where $\mathbf{z}_i$ is a set of observed characteristics of the individual. In both cases, as stated, there is no obvious way that the selection mechanism impacts the binary choice model of interest. We modify the models as follows: For the probit model,

$$y_i^* = \boldsymbol{\beta}'\mathbf{x}_i + \varepsilon_i,\ \varepsilon_i \sim N[0,1],\ y_i = 1(y_i^* > 0),$$

which is the structure underlying the probit model in any event, and

$$u_i,\ \varepsilon_i \sim N_2[(0,0),(1,\rho,1)].$$

(We use N_P to denote the P-variate normal distribution, with the mean vector followed by the definition of the covariance matrix in the succeeding brackets.) For the logit model, a similar approach does not produce a convenient bivariate model. The probability is changed to

$$\text{Prob}(y_i = 1 \mid \mathbf{x}_i,\varepsilon_i) = \frac{\exp(\boldsymbol{\beta}'\mathbf{x}_i + \sigma\varepsilon_i)}{1+\exp(\boldsymbol{\beta}'\mathbf{x}_i + \sigma\varepsilon_i)}.$$

With the selection model for $\mathbf{z}_i$ as stated above, the bivariate probability for y_i and $\mathbf{z}_i$ is a mixture of a logit and a probit model. The log likelihood can be obtained, but it is not in closed form, and must be computed by approximation. We do so with simulation.

There are several formulations for extensions of the binary choice models to panel data setting. These include

- Fixed effects: $\text{Prob}(y_{it} = 1) = F(\boldsymbol{\beta}'\mathbf{x}_{it} + \alpha_i)$, α_i correlated with $\mathbf{x}_{it}$.
- Random effects: $\text{Prob}(y_{it} = 1) = \text{Prob}(\boldsymbol{\beta}'\mathbf{x}_{it} + \varepsilon_{it} + u_i > 0)$, u_i uncorrelated with $\mathbf{x}_{it}$.
- Random parameters: $\text{Prob}(y_{it} = 1) = F(\boldsymbol{\beta}_i'\mathbf{x}_{it})$,
  $\boldsymbol{\beta}_i \mid i \sim h(\boldsymbol{\beta}|i)$ with mean vector $\boldsymbol{\beta}$ and covariance matrix $\boldsymbol{\Sigma}$.
- Latent class: $\text{Prob}(y_{it} = 1|\text{class } j) = F(\boldsymbol{\beta}_j'\mathbf{x}_{it})$,
  $\text{Prob}(\text{class} = j) = G_j(\boldsymbol{\theta},\mathbf{z}_i)$,

where $\mathbf{z}_i$ is a set of observed characteristics of the individual. Other variations include simultaneous equations models and semiparametric formulations.

2.4 Multinomial Logit Model

The canonical random utility model is as follows:

$$U(\text{alternative } 0) = \boldsymbol{\beta}_0'\mathbf{x}_{i0} + \varepsilon_{i0},$$

$$U(\text{alternative } 1) = \boldsymbol{\beta}_1'\mathbf{x}_{i1} + \varepsilon_{i1},$$

$$\ldots$$

$$U(\text{alternative } J) = \boldsymbol{\beta}_J'\mathbf{x}_{iJ} + \varepsilon_{iJ},$$

$$\text{Observed } y_i = \text{choice } j \text{ if } U_i(\text{alternative } j) > U_i(\text{alternative } q)\ \forall\ q \neq j.$$

The 'disturbances' in this framework (individual heterogeneity terms) are assumed to be independently and identically distributed with identical type 1extreme value distribution; the CDF is

$$F(\varepsilon_j) = \exp(-\exp(-\varepsilon_j)).$$

Based on this specification, the choice probabilities are

$$\begin{aligned}\text{Prob(choice } j) &= \text{Prob}(U_j > U_q),\ \forall\ q \neq j \\ &= \frac{\exp(\boldsymbol{\beta}_j'\mathbf{x}_{ij})}{\sum_{q=0}^{J}\exp(\boldsymbol{\beta}_q'\mathbf{x}_{iq})},\ j = 0,\ldots,J.\end{aligned}$$

At this point we make a purely semantic distinction between two cases of the model. When the observed data consist of individual choices and (only) data on the characteristics of the individual, identification of the model parameters will require that the parameter vectors differ across the utility functions, as they do above. The study on labor market decisions by Schmidt and Strauss (1975) is a classic example. For the moment, we will call this the *multinomial logit model*. When the data also include attributes of the choices that differ across the alternatives, then the forms of the utility functions can change slightly – and the coefficients can be generic, that is the same across alternatives. Again, only for the present, we will call this the *conditional logit model*. (It will emerge that the multinomial logit is a special case of the conditional logit model, though the reverse is not true.) The conditional logit model is defined in Section 2.5.

The general form of the *multinomial logit* model is

$$\text{Prob(choice } j) = \frac{\exp(\boldsymbol{\beta}_j'\mathbf{x}_i)}{\sum_{q=0}^{J}\exp(\boldsymbol{\beta}_q'\mathbf{x}_i)},\ j = 0,\ldots,J.$$

A possible $J + 1$ *unordered* outcomes can occur. In order to identify the parameters of the model, we impose the normalization $\boldsymbol{\beta}_0 = \mathbf{0}$. This model is typically employed for individual or grouped data in which the '$\mathbf{x}$' variables are characteristics of the observed individual(s), not the choices.

The data will appear as follows:

- Individual data: y_i coded 0, 1, ..., J,
- Grouped data: $y_{i0}, y_{i1},...,y_{iJ}$ give proportions or shares.
  The structural equations of the multinomial logit model are

$$U_{ijt} = \boldsymbol{\beta}_j'\mathbf{x}_{it} + \varepsilon_{ijt},\ t = 1,...,T_i,\ j = 0,1,...,J, i=1,...,N,$$

where U_{ijt} gives the utility of choice j by person i in period t – we assume a panel data application with $t = 1,...,T_i$. The model about to be described can be applied to cross sections, where $T_i = 1$. Note also that as usual, we assume that panels may be unbalanced. We also assume that ε_{ijt} has a type 1 extreme value distribution and that the J random terms are independent. Finally, we assume that the individual makes the choice with maximum utility. Under these (IIA inducing) assumptions, the probability that individual i makes choice j in period t is

$$P_{ijt} = \frac{\exp(\boldsymbol{\beta}_j'\mathbf{x}_{it})}{\sum_{q=0}^{J}\exp(\boldsymbol{\beta}_q'\mathbf{x}_{it})}.$$

We now suppose that individual i has latent, unobserved, time invariant heterogeneity that enters the utility functions in the form of a random effect, so that

$$U_{ijt} = \boldsymbol{\beta}_j'\mathbf{x}_{it} + \alpha_{ij} + \varepsilon_{ijt},\ t = 1,...,T_i,\ j = 0,1,...,J, i=1,...,N.$$

The resulting choice probabilities, conditioned on the random effects, are

$$P_{ijt} \mid \alpha_{i1},...,\alpha_{iJ} = \frac{\exp(\boldsymbol{\beta}_j'\mathbf{x}_{it} + \alpha_{ij})}{\sum_{q=0}^{J}\exp(\boldsymbol{\beta}_q'\mathbf{x}_{it} + \alpha_{iq})}.$$

To complete the model, we assume that the heterogeneity is normally distributed with zero means and $(J+1)\times(J+1)$ covariance matrix, $\boldsymbol{\Sigma}$. For identification purposes, one of the coefficient vectors, $\boldsymbol{\beta}_q$, must be normalized to zero and one of the α_{iq}s is set to zero. We normalize the first element – subscript 0 – to zero. For convenience, this normalization is left implicit in what follows. It is automatically imposed by the software. To allow the remaining random effects to be freely correlated, we write the $J\times1$ vector of nonzero αs as

$$\boldsymbol{\alpha}_i = \boldsymbol{\Gamma}\,\mathbf{v}_i$$

where $\boldsymbol{\Gamma}$ is a lower triangular matrix to be estimated and $\mathbf{v}_i$ is a standard normally distributed (mean vector $\mathbf{0}$, covariance matrix, $\mathbf{I}$) vector.

2.5 Conditional Logit Model

If the utility functions are conditioned on observed individual, choice invariant characteristics, $\mathbf{z}_i$, as well as the attributes of the choices, $\mathbf{x}_{ij}$, then we write

$$U(\textit{choice } j \textit{ for individual } i) \;=\; U_{ij} \;=\; \boldsymbol{\beta}'\mathbf{x}_{ij} + \boldsymbol{\gamma}_j'\mathbf{z}_i + \;\varepsilon_{ij},\; j = 1,\ldots,J_i.$$

(For this model, which uses a different part of *NLOGIT*, we number the alternatives $1,\ldots,J_i$ rather than $0,\ldots,J_i$. There is no substantive significance to this – it is purely for convenience in the context of the model development for the program commands.) The random, individual specific terms, $(\varepsilon_{i1},\varepsilon_{i2},\ldots,\varepsilon_{iJ})$ are once again assumed to be independently distributed across the utilities, each with the same type 1 extreme value distribution

$$F(\varepsilon_{ij}) \;=\; \exp(-\exp(-\varepsilon_{ij})).$$

Under these assumptions, the probability that individual t chooses alternative j is

$$\text{Prob}(U_{ij} > U_{iq}) \text{ for all } q \neq j.$$

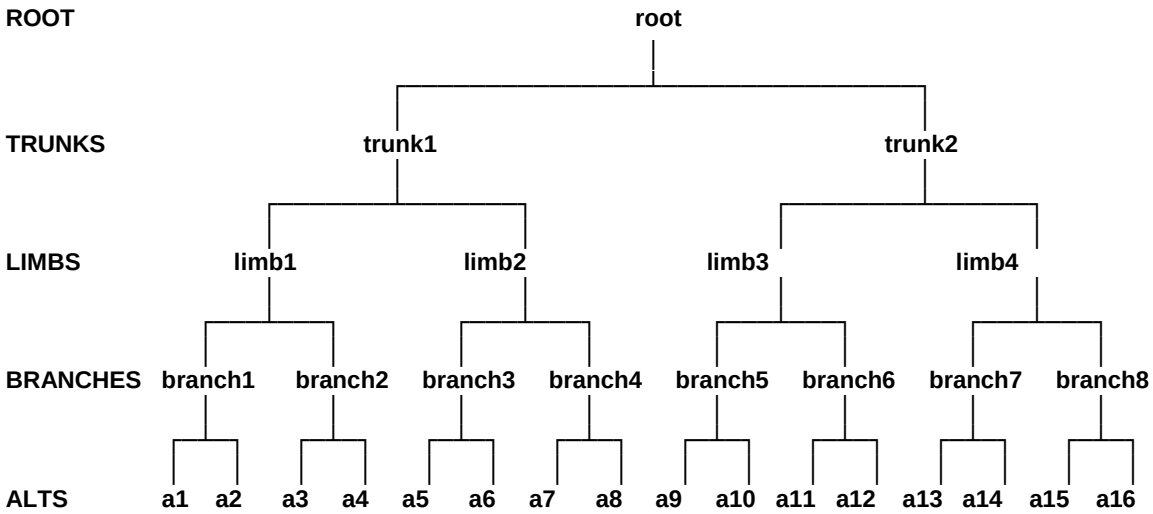
It has been shown that for independent type 1 extreme value distributions, as above, this probability is

$$\text{Prob}(y_i = j) \;=\; \frac{\exp\left(\boldsymbol{\beta}'\mathbf{x}_{ij} + \boldsymbol{\gamma}_j'\mathbf{z}_i\right)}{\sum_{q=1}^{J_i}\exp\left(\boldsymbol{\beta}'\mathbf{x}_{iq} + \boldsymbol{\gamma}_q'\mathbf{z}_i\right)}$$

where y_i is the index of the choice made. We note at the outset that the IID assumptions made about ε_j are quite stringent, and induce the 'Independence from Irrelevant Alternatives' or IIA features that characterize the model. This is functionally identical to the multinomial logit model of Section 2.4. Indeed, the earlier model emerges by the simple restriction $\boldsymbol{\gamma}_j = \mathbf{0}$. We have distinguished it in this fashion because the nature of the data suggests a different arrangement than for the multinomial logit model and, second, the models in the section to follow are formulated as extensions of this one.

2.6 Nested Logit Model

The nested logit model is an extension of the conditional logit model. The models supported by *NLOGIT* are based on variations of a four level tree structure such as the following:



The choice probability under the assumption of the nested logit model is defined to be the conditional probability of alternative j in branch b, limb l, and trunk r, $j|b,l,r$:

$$P(j|b,l,r) = \frac{\exp(\boldsymbol{\beta}'\mathbf{x}_{j|b,l,r})}{\sum_{q|b,l,r}\exp(\boldsymbol{\beta}'\mathbf{x}_{q|b,l,r})} = \frac{\exp(\boldsymbol{\beta}'\mathbf{x}_{j|b,l,r})}{\exp(J_{b|l,r})},$$

where $J_{b|l,r}$ is the inclusive value for branch b in limb l, trunk r, $J_{b|l,r} = \log \Sigma_{q|b,l,r}\exp(\boldsymbol{\beta}'\mathbf{x}_{q|b,l,r})$. At the next level up the tree, we define the conditional probability of choosing a particular branch in limb l, trunk r,

$$P(b|l,r) = \frac{\exp(\boldsymbol{\alpha}'\mathbf{y}_{b|l,r} + \tau_{b|l,r}J_{b|l,r})}{\sum_{s|l,r}\exp(\boldsymbol{\alpha}'\mathbf{y}_{s|l,r} + \tau_{s|l,r}J_{s|l,r})} = \frac{\exp(\boldsymbol{\alpha}'\mathbf{y}_{b|l,r} + \tau_{b|l,r}J_{b|l,r})}{\exp(I_{l|r})},$$

where $I_{l|r}$ is the inclusive value for limb l in trunk r, $I_{l|r} = \log \Sigma_{s|l,r}\exp(\boldsymbol{\alpha}'\mathbf{y}_{s|l,r} + \tau_{s|l,r}J_{s|l,r})$. The probability of choosing limb l in trunk r is

$$P(l|r) = \frac{\exp(\boldsymbol{\delta}'\mathbf{z}_{l|r} + \sigma_{l|r}I_{l|r})}{\sum_{s|r}\exp(\boldsymbol{\delta}'\mathbf{z}_{q|r} + \sigma_{s|r}I_{s|r})} = \frac{\exp(\boldsymbol{\delta}'\mathbf{z}_{l|r} + \sigma_{l|r}I_{l|r})}{\exp(H_r)},$$

where H_r is the inclusive value for trunk r, $H_r = \log \Sigma_{s|l}\exp(\boldsymbol{\delta}'\mathbf{z}_{s|r} + \sigma_{s|r}I_{s|r})$. Finally, the probability of choosing a particular limb is

$$P(r) \;=\; \frac{\exp(\boldsymbol{\theta}'\mathbf{h}_r + \phi_r H_r)}{\sum_s \exp(\boldsymbol{\theta}'\mathbf{h}_s + \phi_s H_s)}.$$

By the laws of probability, the unconditional probability of the observed choice made by an individual is

$$P(j,b,l,r) \;=\; P(j|b,l,r) \times P(b|l,r) \times P(l|r) \times P(r).$$

This is the contribution of an individual observation to the likelihood function for the sample.

The 'nested logit' aspect of the model arises when any of the $\tau_{b|l,r}$ or $\sigma_{l|r}$ or ϕ_r differ from 1.0. If all of these deep parameters are set equal to 1.0, the unconditional probability reduces to

$$P(j,b,l,r) \;=\; \frac{\exp(\boldsymbol{\beta}'\mathbf{x}_{j|b,l,r} + \boldsymbol{\alpha}'\mathbf{y}_{b|l,r} + \boldsymbol{\delta}'\mathbf{z}_{l|r} + \boldsymbol{\theta}'\mathbf{h}_r)}{\sum_r \sum_l \sum_b \sum_j \exp(\boldsymbol{\beta}'\mathbf{x}_{j,b,l,r} + \boldsymbol{\alpha}'\mathbf{y}_{b,l,r} + \boldsymbol{\delta}'\mathbf{z}_{l,r} + \boldsymbol{\theta}'\mathbf{h}_r)},$$

which is the probability for a one level conditional (multinomial) logit model.

2.7 Random Parameters Logit Models

In its most general form, we write the multinomial logit probability as

$$P(j \mid \mathbf{v}_i) = \frac{\exp(\alpha_{ji} + \boldsymbol{\theta}'_j\mathbf{z}_i + \boldsymbol{\phi}'_j\mathbf{f}_{ji} + \boldsymbol{\beta}'_{ji}\mathbf{x}_{ji})}{\sum_{q=1}^{J}\exp(\alpha_{qi} + \boldsymbol{\theta}'_q\mathbf{z}_i + \boldsymbol{\phi}'_q\mathbf{f}_{qi} + \boldsymbol{\beta}'_{qi}\mathbf{x}_{qi})},$$

where $U(j,i) \;=\; \alpha_{ji} + \boldsymbol{\theta}'_j\mathbf{z}_i + \boldsymbol{\phi}'_j\mathbf{f}_{ji} + \boldsymbol{\beta}'_{ji}\mathbf{x}_{ji}$, $j = 1,\ldots,J_i$ alternatives in individual i's choice set,

α_{ji} is an alternative specific constant which may be fixed or random, $\alpha_{Ji} = 0$,

$\boldsymbol{\theta}_j$ is a vector of nonrandom (fixed) coefficients, $\boldsymbol{\theta}_{Ji} = \mathbf{0}$,

$\boldsymbol{\phi}_j$ is a vector of nonrandom (fixed) coefficients,

$\boldsymbol{\beta}_{ji}$ is a coefficient vector that is randomly distributed across individuals; $\mathbf{v}_i$ enters $\boldsymbol{\beta}_{ji}$,

$\mathbf{z}_i$ is a set of choice invariant individual characteristics such as age or income,

$\mathbf{f}_{ji}$ is a vector of M individual and choice varying attributes of choices, multiplied by $\boldsymbol{\phi}_j$,

$\mathbf{x}_{ji}$ is a vector of L individual and choice varying attributes of choices, multiplied by $\boldsymbol{\beta}_{ji}$.

The term 'mixed logit' is often used in the literature for this model. The choice specific constants, α_{ji} and the elements of $\boldsymbol{\beta}_{ji}$ are distributed randomly across individuals such that for each random coefficient, ρ_{ki} = any (not necessarily all of) α_{ji} or β_{jki}, the coefficient on attribute x_{jik}, $k = 1,...,K$,

$$\rho_{jki} = \alpha_{ji} \text{ or } \beta_{jki} = \rho_{jk} + \boldsymbol{\delta}_k'\mathbf{w}_i + \sigma_k v_{ki},$$

or

$$\rho_{jki} = \alpha_{ji} \text{ or } \beta_{jki} = \exp(\rho_{jk} + \boldsymbol{\delta}_{jk}'\mathbf{w}_i + \sigma_{jk} v_{jki}).$$

The vector $\mathbf{w}_i$ (which does not include *one*) is a set of choice invariant characteristics that produce individual heterogeneity in the means of the randomly distributed coefficients; ρ_{jk} is the constant term and $\boldsymbol{\delta}_{jk}$ is a vector of 'deep' coefficients which produce an individual specific mean. The random term, v_{jki} is normally distributed (or distributed with some other distribution) with mean 0 and standard deviation 1, so σ_{jk} is the standard deviation of the marginal distribution of ρ_{jki}. The v_{jki}s are individual and choice specific, unobserved random disturbances – the source of the heterogeneity. Thus, as stated above, in the population

$$\alpha_{ji} \text{ or } \beta_{jki} \sim \text{Normal or Lognormal } [\rho_{jk} + \boldsymbol{\delta}_{jk}'\mathbf{w}_i, \sigma_{jk}^2].$$

(Other distributions may be specified.) For the full vector of K random coefficients in the model, we may write

$$\boldsymbol{\rho}_i = \boldsymbol{\rho} + \boldsymbol{\Delta}\mathbf{w}_i + \boldsymbol{\Gamma}\mathbf{v}_i$$

where $\boldsymbol{\Gamma}$ is a diagonal matrix which contains σ_k on its diagonal. A nondiagonal $\boldsymbol{\Gamma}$ allows the random parameters to be correlated. Then, the full covariance matrix of the random coefficients is $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Gamma}'$. The standard case of uncorrelated coefficients has $\boldsymbol{\Gamma} = \text{diag}(\sigma_1,\sigma_2,\ldots,\sigma_k)$. If the coefficients are freely correlated, $\boldsymbol{\Gamma}$ is a full, unrestricted, *lower triangular* matrix and $\boldsymbol{\Sigma}$ will have nonzero off diagonal elements. An additional level of flexibility is obtained by allowing the distributions of the random parameters to be heteroscedastic,

$$\sigma_{ijk}^2 = \sigma_{jk}^2 \times \exp(\boldsymbol{\gamma}_{jk}'\mathbf{h}_i).$$

This is now built into the model by specifying

$$\boldsymbol{\rho}_i = \boldsymbol{\rho} + \boldsymbol{\Delta}\mathbf{w}_i + \boldsymbol{\Gamma}\boldsymbol{\Omega}_i\mathbf{v}_i$$

where

$$\boldsymbol{\Omega}_i = \text{diag}[\sigma_{ijk}^2]$$

and now, $\boldsymbol{\Gamma}$ is a lower triangular matrix of constants with ones on the diagonal. Finally, autocorrelation can also be incorporated by allowing the random components of the random parameters to obey an autoregressive process,

$$v_{ki,t} = \tau_{ki}\, v_{ki,t-1} + c_{ki,t}$$

where $c_{ki,t}$ is now the random element driving the random parameter.

This produces, then, the full random parameters logit model

$$P(j \mid \mathbf{v}_i) = \frac{\exp(\alpha_{ji} + \boldsymbol{\beta}_i' \mathbf{x}_{ji})}{\sum_{m=1}^{J} \exp(\alpha_{mi} + \boldsymbol{\beta}_i' \mathbf{x}_{mi})},$$

$$\boldsymbol{\beta}_i = \boldsymbol{\beta} + \Delta \mathbf{z}_i + \boldsymbol{\Gamma}\, \boldsymbol{\Omega}_i\, \mathbf{v}_i$$

$\mathbf{v}_i$ ~ with mean vector $\mathbf{0}$ and covariance matrix $\mathbf{I}$.

The specific distributions may vary from one parameter to the next. We also allow the parameters to be lognormally distributed so that the preceding specification applies to the logarithm of the specific parameter.

2.8 Multinomial Probit Model

In this model, the individual's choice among J alternatives is the one with maximum utility, where the utility functions are

$$U_{ji} = \boldsymbol{\beta}' \mathbf{x}_{ji} + \varepsilon_{ji}$$

where U_{ji} = utility of alternative j to individual i

$\mathbf{x}_{jit}$ = union of all attributes that appear in all utility functions. For some alternatives, $x_{jit,k}$ may be zero by construction for some attribute k which does not enter their utility function for alternative j.

The multinomial logit model specifies that ε_{ji} are draws from independent extreme value distributions (which induces the IIA condition). In the multinomial probit model, we assume that ε_{ji} are normally distributed with standard deviations Sdv[ε_{ji}] = σ_j and correlations Cor[ε_{ji}, ε_{qi}] = ρ_{jq} (the same for all individuals). Observations are independent, so Cor[ε_{ji},ε_{qs}] = 0 if i is not equal to s, for all j and q. A variation of the model allows the standard deviations and covariances to be scaled by a function of the data, which allows some heteroscedasticity across individuals.

The correlations ρ_{jq} are restricted to $-1 < \rho_{jq} < 1$, but they are otherwise unrestricted save for a necessary normalization. The correlations in the last row of the correlation matrix must be fixed at zero. The standard deviations are unrestricted with the exception of a normalization – two standard deviations are fixed at 1.0 – *NLOGIT* fixes the last two.

This model may also be fit with panel data. In this case, the utility function is modified as follows:

$$U_{ji,t} = \boldsymbol{\beta}' \mathbf{x}_{ji,t} + \varepsilon_{ji,t} + v_{ji,t}$$

where 't' indexes the periods or replications. There are two formulations for $v_{ji,t}$,

Random effects $v_{ji,t} = v_{ji,t}$ (the same in all periods)

First order autoregressive $v_{ji,t} = \alpha_j\, v_{ji,t-1} + a_{ji,t}$.

It is assumed that you have a total of T_i observations (choice situations) for person i. Two situations might lend themselves to this treatment. If the individual is faced with a set of choice situations that are similar and occur close together in time, then the random effects formulation is likely to be appropriate. However, if the choice situations are fairly far apart in time, or if habits or knowledge accumulation are likely to influence the latter choices, then the autoregressive model might be the better one.

You can also add a form of individual heterogeneity to the disturbance covariance matrix. The model extension is

$$\mathrm{Var}[\boldsymbol{\varepsilon}_i] = \exp[\boldsymbol{\gamma}'\mathbf{h}_i] \times \boldsymbol{\Sigma}$$

where $\boldsymbol{\Sigma}$ is the matrix defined earlier (the same for all individuals), and $\mathbf{h}_i$ is an individual (not alternative) specific set of variables not including a constant.

Chapter 3: Model and Command Summary for Discrete Choice Models

3.1 Introduction

The chapters to follow will provide details on the various discrete choice models you can estimate with *NLOGIT* and on the model commands you will use to request the estimates. This chapter will provide a brief summary listing of the models and model commands. The variety of logit models now use a set of specific names, rather than qualifiers to more general model classes as in earlier versions. For example, the model name **OLOGIT** can be used instead of **ORDERD ; Logit**. The earlier formats remain available, but the newer ones may prove more convenient. The full listing of these commands is also given below. The commands below specify the essential parts needed to fit the model. The numerous options and different forms are discussed in the chapters to follow.

3.2 Model Summary

The descriptions below present the different discrete choice models that are the main feature of *NLOGIT*. Note, once again, *NLOGIT* contains all of *LIMDEP*, so all of the models documented in the *Econometric Modeling Guide*, including the regression models, limited dependent variable models, generalized linear models, sample selection models, and so on are supported in *NLOGIT*, as well as the ancillary tools including **MATRIX**, etc.

3.3 Basic Discrete Choice Models

The binomial probit and logit models and the ordered probit and logit models are the primary model frameworks for single equation, single decision, discrete choice models. The ordered choice and the bivariate and multivariate probit models are multivariate extensions of the simple probit model.

There are five binary choice models, probit, logit, complementary log log, Gompertz and Burr. The ones that interest us here are the binary probit and logit models. The probit model is requested with

```
PROBIT      ; Lhs = dependent variable
            ; Rhs = independent variables $
```

The binary logit model may be invoked with

```
BLOGIT      ; Lhs = dependent variable
            ; Rhs = independent variables $
```

In earlier versions, you would use the **LOGIT** command, which is still useable. **LOGIT** is the same as **BLOGIT** when the data on the dependent variable are either binary (zeros and ones) or proportions (strictly between zero and one).

3.4 Multinomial Logit Models

The 'multinomial logit model' is an early, restrictive version of the conditional logit model, which, itself, is the gateway model to the main model extensions described in Section 3.5.

3.4.1 Multinomial Logit

The multinomial logit model is invoked with

MLOGIT ; Lhs = dependent variable
; Rhs = independent variables $

Data for the MLOGIT model consist of an integer valued variable taking the values 0, 1, ..., *J*. This model may also be fit with proportions data. In that case, you will provide the names of *J*+1 Lhs variables that will be strictly between zero and one, and will sum to one at every observation. The **MLOGIT** command is the same as **LOGIT**. The program inspects the command (Lhs) and the data, and determines internally whether **BLOGIT** or **MLOGIT** is appropriate. Note, on proportions data, if you want to fit a binary logit model with proportions data, you will supply a single proportions variable, not two. (What would be the second one is just one minus the first.) If you want to fit a multinomial logit model with proportions data with three or more outcomes, you must provide the full set of proportions. Thus, you would never supply two Lhs variables in a **LOGIT**, **BLOGIT** or **MLOGIT** command.

3.4.2 Conditional Logit

The command for the conditional model, and the commands in the sections to follow, are variants of the **NLOGIT** command. This is a full class of estimators based on the conditional logit form. There are several forms of the essential command for fitting the conditional logit model with *NLOGIT*. The simpler one is

CLOGIT ; Lhs = dependent variable
; Choices = the names of the J alternatives
; Rhs = list of choice specific attributes
; Rh2 = list of choice invariant individual characteristics $

As discussed in Chapter 5, the data for this estimator consist of a set of *J* observations, one for each alternative. (The observation resembles a group in a panel data set.) The command just given assumes that every individual in the sample chooses from the same size choice set, *J*. The choice sets may have different numbers of choices, in which case, the command is changed to

; Lhs = dependent variable, choice set size variable

The second Lhs variable is structured exactly the same as a **; Pds** variable for a panel data estimator. In the second form of the model command, the utility functions are specified directly, symbolically.

The **; Rhs** and **; Rh2** specifications can be replaced with

; Model: ... specification of the utility functions

This is discussed in Chapter 6.

The **CLOGIT** command is the same as **DISCRETE CHOICE.** It is also the same as **NLOGIT** when the only information given in the command is that specified above, that is when none of the specifications that invoke the model extensions that are described in the sections to follow are provided.

3.5 NLOGIT Extensions of Conditional Logit

3.5.1 Nested

The nested logit model is the default form of the **NLOGIT** command. Request the nested logit model with

```
NLOGIT      ; Tree = specification of the tree structure
            ; Choices = the names of the J alternatives
            ; Rhs  = list of choice specific attributes
```

3.5.2 Random Parameters Logit

The random parameters logit model (mixed logit model) is requested by specifying a conditional logit model, and adding the specification of the random parameters. The model command is

```
RPLOGIT     ; Lhs  = dependent variable
            ; Choices = the names of the J alternatives
            ; Rhs  = list of choice specific attributes
            ; Rh2 = list of choice invariant individual characteristics
            ; Fcn  = the specifications of the random parameters
            ; ... other specifications for the random parameters model $
```

Once again, variable choice set sizes and utility function specifications are specified as in the **CLOGIT** command. This command is the same as

```
NLOGIT      ; RPL
            ; ... the rest of the command $
```

There is one modification that might be necessary. If you are providing variables that affect the means of the random parameters, you would generally use

```
NLOGIT      ; RPL = the list of variables
            ; ... the rest of the command $
```

The RPL specification may still be used this way. The command can be **NLOGIT** as above, or

```
RPLOGIT     ; RPL = the list of variables
            ; ... the rest of the command $
```

These are identical.

The random parameters model may also include an error components specification defined in the next section. The command will be

```
RPLOGIT    ; Lhs = dependent variable
           ; Choices = the names of the J alternatives
           ; Rhs = list of choice specific attributes
           ; Rh2 = list of choice invariant individual characteristics
           ; Fcn = the specifications of the random parameters
           ; ... other specifications for the random parameters model
           ; ECM = specification $
```

3.5.3 Multinomial Probit

The multinomial probit model is described in Chapter 11. The essential command is

```
MNPROBIT   ; Lhs = dependent variable
           ; Choices = the names of the J alternatives
           ; Rhs = list of choice specific attributes
           ; Rh2 = list of choice invariant individual characteristics $
```

Variable choice set sizes and utility function specifications are specified as in the **CLOGIT** command. This command is the same as

```
NLOGIT     ; MNP
           ; ... the rest of the command $
```

3.6 Command Summary

The following lists the current and where applicable, alternative forms of the discrete choice model commands. The two sets of commands are identical, and for each model, in *NLOGIT* 4.0, either command may be used for that model.

Models	Command	Alternative Command Form
Binary Choice Models		
Binary Probit	**PROBIT**	**PROBIT**
Binary Logit	**BLOGIT**	**LOGIT**
Multinomial Logit Models		
Multinomial Logit	**MLOGIT**	**LOGIT**
Conditional Logit	**CLOGIT**	**DISCRETE CHOICE**
Conditional Logit Extensions		
Conditional Logit	**CLOGIT**	**CLOGIT**
Multinomial Logit	**NLOGIT**	**NLOGIT** (Same as **CLOGIT**)
Nested Logit	**NLOGIT ; Tree = ...**	**NLOGIT ; Tree = ...**
Random Parameters Logit	**RPLOGIT**	**NLOGIT ; RPL**
Multinomial Probit	**MNPROBIT**	**NLOGIT ; MNP**

3.7 Subcommand Summary

The following subcommands are used in *NLOGIT* model commands. The **BLOGIT**, **BPROBIT**, **BVPROBIT**, **MVPROBIT**, **OLOGIT** and **OPROBIT** commands have additional specifications that are documented in the *LIMDEP Econometric Modeling Guide* for these specific models. The specifications below are those that may appear in the **NLOGIT** command or the conditional logit extensions described above.

General Model Specification and Data Setup

Data on Dependent Variable

; Ranks indicates that data are in the form of ranks, possibly ties at last place.
; Shares indicates that data are in the form of proportions or shares.
; Frequencies indicates that data are in the form of frequencies or counts.
; Checkdata checks validity of the data before estimation.
; Wts = weighting variable uses a weighting variable. (Noscale is not used here.)
; Scale (list of variables) = values for scaling loop specifies scaling of certain variables during iterations.
; Pds = specification used by RPL, LCM, ECM, MNP and by binary choice models to indicate a panel data set. Indicates multiple choice situations for individuals.

Specification of the Dependent Variable

; Lhs = list of variables used by all models to name the dependent variable. Second Lhs variable indicates variable choice set size. Third Lhs variable indicates specific choices in a universal choice set. First variable is a set of utilities if **; MCS** is used.
; MCS requests data generated by Monte Carlo simulation.
; Choices = list lists names for alternatives.

Specification of Utility Functions

; Rhs = list of variables lists choice varying attribute variables.
; Rh2 = list of variables lists choice invariant characteristic variables.
; Model: alternative way to specify utility functions, followed by definitions of utility functions.
; Fix = list lists names of and values for coefficients that are to be fixed.
; Uset (list of alternatives) = list of values or [list of values] alternative method of specifying starting values or fixed coefficients.
; Lambda = value specifies coefficient to use for Box-Cox transformation.
; Attr= list of names names for attributes used in one line entry format.

Output Control

List and Retain Variables and Results

; Prob = variable name keeps predicted probabilities from estimated model as variable.
; Keep = variable name keeps predicted values from estimated model as variable. Used by **PROBIT** and **BLOGIT** only.
; Utility = name keeps predicted utilities as variable.
; List lists predicted probabilities and predicted outcomes with model results.
; Parameters retains additional parameters as matrices. With RPL and LCM, keeps matrices of individual specific parameter means.
; WTP = list of specifications retains computations of willingness to pay.

Covariance Matrices

; Printvc displays estimated covariance matrix with model output.
; Robust computes robust sandwich estimator for asymptotic covariance matrix.
; Cluster = specification computes robust cluster corrected asymptotic covariance matrix.

Display of Estimation Results

; Show displays model specification and tree structure.
; Describe lists descriptive statistics for attributes by alternative.
; Odds includes odds ratios in estimation results. Used only by **BLOGIT**.
; Crosstab includes crosstabulation of predicted and actual outcomes.
; Table = name adds model results to stored tables.

Marginal Effects

; Effects: specification displays estimated marginal effects. Used by **NLOGIT**.
; Marginal Effects displays estimated marginal effects. Used by **PROBIT**, **BLOGIT**, **BVPROBIT**, **MVPROBIT**, **OLOGIT**, **OPROBIT**.
; Means computes marginal effects using data means. Uses average partial effects if this is not specified.
; Pwt uses probability weights to compute average partial effects.

Hypothesis Testing

; Wald: specification computes Wald test statistic for specified linear restrictions.
; Test: specification same as **Wald: specification**.
; IAS = list of choices used with **CLOGIT** to test IIA assumption.

Optimization

Iterations Controls

; Alg = algorithm specifies optimization method.
; Maxit = value specifies maximum iterations.
; Tlg = value tolerance for convergence on gradient.
; Tlb = value tolerance for convergence on change in parameters.
; Tlg = value tolerance for convergence on change in function.
; Set keeps settings of tolerance values.
; Output = value displays technical output during iterations.

Starting Values

; Start = list of values provides starting values for all model parameters.
; PR0 = list of values provides starting values for free parameters only. (Generally not used.)

Constrained Estimation

; CML: specification constrained maximum likelihood estimator.
; Rst = list of values and symbols imposes fixed value and equality constraints.
; Calibrate fixes parameters at previously estimated values.
; ASC initially fit model with just ASCs.

Criterion Function for CLOGIT

; GME [= number of support points] generalized maximum entropy. Used by **MLOGIT** and **CLOGIT**.
; Sequential sequential two step estimator for nested logit. (Generally not used.)
; Conditional conditional estimator for two step nested logit. (Generally not used.)

Simulation Based Estimation

; Pts = number of replications number of replications for simulation estimator. Used by ECM and MNP. (Also used by LCM to specify number of latent classes.)
; Shuffled uses shuffled uniform draws to compute draws for simulations.
; Halton uses Halton sequences for simulation based estimators.

Simulation Processor (BINARY CHOICE Command for PROBIT and BLOGIT)

; Simulation [= list of choices] simulates effect of changes in attributes on aggregate outcomes.
; Scenarios specifies changes in attributes for simulations.
; Arc computes arc elasticities during simulations.
; Merge merges revealed and stated preference data during simulations.

Specific NLOGIT Model Commands

; LCM [= list of variables] specifies latent class model. Optionally, specifies variables that enter the class probabilities. (Command is also **LCLOGIT**.) Also used by **PROBIT** and **BLOGIT**.
; ECM = list of specifications specifies error components logit model. (Command is also **ECLOGIT**.)
; HEV specifies heteroscedastic extreme value model. (Command is also **HCLOGIT**.)

Nested Logit Model

; Tree = specification specifies tree structure in nested logit model.
; GNL specifies generalized nested logit model. (Command is also **GNLOGIT**.)
; RU1 specifies parameterization of second and third levels of the tree.
; RU2 specifies parameterization of second and third levels of the tree.
; RU3 specifies parameterization of second and third levels of the tree.
; IVSET: specifications imposes constraints on inclusive value parameters.
; IVB= variable name keeps branch level inclusive values as a variable.
; IVL= name for limb IV keeps limb level inclusive values as a variable.
; IVT= name for trunk IV keeps trunk level inclusive values as a variable.
; Prb = name keeps branch level probabilities as a variable.
; Cprob = name keeps conditional probabilities for alternatives.

Random Parameters Logit Model

; RPL [= list of variables] requests mixed logit model. Optionally specifies variables to enter means of random parameters.
; AR1 AR(1) structure for random terms in random parameters.
; Fcn: defines names and types of random parameters.
; Correlation specifies that random parameters are correlated.
; Hfr = list of variables defines variables in heteroscedasticity. Also used by HEV and covariance heterogeneity.

Multinomial Probit

; MNP specifies multinomial probit model. (Command is also **MNPROBIT**.)
; EQC = list of choices specifies a set of choices whose pairwise correlations are all equal.
; RCR = list of specifications specifies configurations for correlations for multinomial probit model. Also used by RPL.
; SDV = list of specifications specifies diagonal elements of covariance matrix. Also used by RPL and HEV.
; REM specifies random effects form of the model.

Chapter 4: The Basic Multinomial Logit Model

4.1 Introduction

This chapter will describe a basic form of the 'multinomial logit' model. These models are also known variously as 'conditional logit,' 'discrete choice,' and 'universal logit' models, among other names. All of them can be viewed as special cases of a general model of utility maximization: An individual is assumed to have preferences defined over a set of alternatives (travel modes, occupations, food groups, etc.)

$$U_i(\text{alternative } 0) = \boldsymbol{\beta}_0'\mathbf{x}_{i0} + \varepsilon_{i0},$$

$$U_j(\text{alternative } 1) = \boldsymbol{\beta}_1'\mathbf{x}_{i1} + \varepsilon_{i1},$$

$$\ldots$$

$$U_i(\text{alternative } J) = \boldsymbol{\beta}_J'\mathbf{x}_{iJ} + \varepsilon_{iJ},$$

$$\text{Observed } Y_i = j \text{ if } U_i(\text{ alternative } j\,) > U_i(\text{ alternative } q\,)\ \forall\ q \neq j.$$

The 'disturbances' in this framework (individual heterogeneity terms) are assumed to be independently and identically distributed with identical type 1 extreme value distribution; the CDF is

$$F(\varepsilon_j) = \exp(-\exp(-\varepsilon_j)).$$

Based on this specification, the choice probabilities,

$$\text{Prob(choice } j\,) = \text{Prob}(U_j > U_q),\ \forall\ q \neq j$$

$$= \frac{\exp(\boldsymbol{\beta}_j'\mathbf{x}_{ji})}{\sum_{m=0}^{J}\exp(\boldsymbol{\beta}_m'\mathbf{x}_{mi})},\ j = 0,\ldots,J,$$

where 'i' indexes the observation, or individual, and 'j' and 'm' index the choices. The IID assumptions made about ε_j are quite stringent, and lead to the 'Independence from Irrelevant Alternatives' or IIA implications that characterize the model. Much (perhaps all) of the research on forms of this model consists of development of alternative functional forms and stochastic specifications that avoid this feature. The observed data consist of the Rhs vectors, $\mathbf{x}_{jt}$, and the outcome, or choice, y_t. (We also consider a number of variants.)

This chapter will examine what we call, for the present, the *multinomial logit* model. In this model, it is assumed that the Rhs variables consist of a set of individual specific characteristics, such as age, education, marital status, etc. These are the same for all choices, so the choice subscript on **x** in the formula above is dropped. The observation setting is the individual's choice among a set of alternatives, where it is assumed that the determinant of the choice is the *characteristics* of the individual. An example might be a model of choice of occupation. The remaining chapters of this manual after this one will examine what we call (again only for convenience) the *discrete choice* model and, also, to differentiate the command, the *conditional logit* model. In this framework, we observe the *attributes* of the choices, rather than the characteristics of the individual. A well known

example is travel mode choice. Samples of observations often consist of the attributes of the different modes and the choice actually made. Usually, no characteristics of the individuals are observed beyond their actual choice. Models may also contain mixtures of the two types of choice determinants. These are considered in the later chapters as well. (We emphasize, these naming distinctions are meaningless in the modeling framework – we just use them here only to organize the applicable parts of *NLOGIT*.

4.2 The Multinomial Logit Model

The general form of the *multinomial logit* model is

$$\text{Prob(choice } j) = \frac{\exp(\boldsymbol{\beta}_j'\mathbf{x}_t)}{\sum_{m=1}^{J}\exp(\boldsymbol{\beta}_m'\mathbf{x}_t)}, j = 0,...,J.$$

A possible J+1 *unordered* outcomes can occur. In order to identify the parameters of the model, we impose the normalization $\boldsymbol{\beta}_0 = \mathbf{0}$. This model is typically employed for individual or grouped data in which the '**x**' variables are characteristics of the observed individual(s), not the choices. For present purposes, that is the main distinction between this and the *discrete choice* model described in Chapter 8. The characteristics are the same across all outcomes.

The data will appear as follows:

- Individual data: y_i coded 0, 1, ..., J,
- Grouped data: $y_{0i}, y_{1i},...,y_{Ji}$ give proportions or shares.

In the grouped data case, a weighting variable, n_t, may also be provided if the observations happen to be frequencies. The proportions variables must range from zero to one and sum to one at each observation. The full set must be provided, even though one is redundant. The data are inspected to determine which specification is appropriate. The number of Lhs variables given and the coding of the data provide the full set of information necessary to estimate the model, so no additional information about the dependent variable is needed.

This model proliferates parameters. There are $J\times K$ nonzero parameters in all, since there is a vector $\boldsymbol{\beta}_j$ for each probability except the first. Consequently, even moderately sized models quickly become very large ones if your outcome variable, y, takes many values. The maximum number of parameters which can be estimated in a model is 150 as usual with the standard configuration. However, if you are able to forego certain other optional features, the number of parameters can increase to 300. (This is the only model in *NLOGIT* that extends the 150 parameter limit.) The model size is detected internally. If your configuration contains more than 150 parameters, the following options and features become unavailable:

- marginal effects
- choice based sampling
- **; Rst = list** for imposing restrictions
- **; CML:** specification for imposing linear constraints
- **; Hold** for using the multinomial logit model as a sample selection equation

In addition, if your model size exceeds 150 parameters, the matrices *b* and *varb* cannot be retained.

4.3 Model Command for the Multinomial Logit Model

The command for fitting this form of multinomial logit model is

```
MLOGIT      ; Lhs = y    or    y0,y1,...yJ
            ; Rhs = regressors $
```

(The command may also be **LOGIT**, which is what has always been used in previous versions of *LIMDEP and NLOGIT*.) All general options for controlling output and iterations are available except **; Keep = name**. (A program which can be used to obtain the fitted probabilities is listed below.) There are internally computed predictions for the multinomial logit model.

The **; Rst = list** form of restrictions is supported for imposing constraints on model parameters, either fixed value or equality. One possible application of the constrained model involves making the entire vector of coefficients in one probability equal that in another. You can do this as follows:

```
NAMELIST    ; x = the entire set of Rhs variables $
CALC        ; k = Col(x) $
LOGIT             ; Lhs     = y
            ; Rhs = x
            ; Rst  = k_b, k_b, ... , k_b $
```

This would force the corresponding coefficients in all probabilities to be equal. You could also apply this to some, but not all of the outcomes, as in

```
            ; Rst = k_b, k_b, k_b2, k_b3
```

HINT: The coefficients in this model are not the marginal effects. But, forcing the coefficient on a characteristic in probability *j* to equal its counterpart in probability *m* also forces the two marginal effects to be equal.

4.4 Robust Covariance Matrices

It has become common in the literature to compute a 'robust covariance matrix' for the MLE. (The misspecification to which the matrix is robust is left unspecified in most cases.) The desired robust covariance matrix would result in the preceding computation if w_i equals one for all observations. This suggests a simple way to obtain it, just by specifying **; Choice Based ; Wts = one**. Alternatively, just use

```
            ; Robust
```

which is equivalent.

A related calculation is used when observations occur in groups which may be correlated. This is rather like a panel; one might use this approach in a random effects kind of setting in which observations have a common latent heterogeneity. The parameter estimator is unchanged in this case, but an adjustment is made to the estimated asymptotic covariance matrix. The calculation is done as follows: Suppose the n observations are assembled in C clusters of observations, in which the number of observations in the cth cluster is n_c. Thus,

$$\sum_{c=1}^{C} n_c = n.$$

Denote by $\boldsymbol{\beta}$ the full set of model parameters, $[\boldsymbol{\beta}_1', ..., \boldsymbol{\beta}_J']'$. Let the observation specific gradients and Hessians for individual i in cluster c be

$$\mathbf{g}_{ic} = \frac{\partial \log L_{ic}}{\partial \boldsymbol{\beta}},$$

$$\mathbf{H}_{ic} = \frac{\partial^2 \log L_{ic}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'}.$$

The uncorrected estimator of the asymptotic covariance matrix based on the Hessian is

$$\mathbf{V}_H = -\mathbf{H}^{-1} = \left(-\sum_{c=1}^{C} \sum_{i=1}^{n_c} \mathbf{H}_{ic}\right)^{-1}.$$

The corrected asymptotic covariance matrix is

$$\text{Est.Asy.Var}\left[\hat{\boldsymbol{\beta}}\right] = \mathbf{V}_H \frac{C}{C-1}\left[\sum_{c=1}^{C}\left(\sum_{i=1}^{n_c} \mathbf{g}_{ic}\right)\left(\sum_{i=1}^{n_c} \mathbf{g}_{ic}\right)'\right] \mathbf{V}_H .$$

Note that if there is exactly one observation per cluster, then this is $C/(C\text{-}1)$ times the sandwich (robust) estimator discussed above. Also, if you have fewer clusters than parameters, then this matrix is singular – it has rank equal to the minimum of C and JK, the number of parameters. This estimator is requested with

; Cluster = specification

where the specification is either a fixed number of observations per cluster, or an identifier that distinguishes clusters, such as an identification number. This estimator can also be extended to stratified as well as clustered data, using

; Stratum = specification

4.5 Output for the Multinomial Logit Model

Initial ordinary least squares results are used for the starting values for this model. For individual data, J binary variables are implied by the model. These are used in a least squares regression. For the grouped data case, a minimum chi squared, generalized least squares estimate is obtained by the weighted regression of

$$o_{ij} = \log(P_{ij} / P_{i0})$$

on the regressors, with weights $h_{ij} = (n_i P_{ij} P_{i0})^{1/2}$ (n_i may be 1.0). (Note that the dependent variables in these regressions are the 'odds ratios.') The OLS estimates based on the individual data are inconsistent, but the grouped data estimates are consistent (and, in the binomial case, efficient). The least squares estimates are included in the displayed results by including

; OLS

in the model command. The iterations are followed by the maximum likelihood estimates with the usual diagnostic statistics. An example is shown below.

NOTE: Minimum chi squared (MCS) is an estimator, not a model. Moreover, the MCS estimator has the same properties as, but is different from the maximum likelihood estimator. Since the MCS estimator in *NLOGIT* is not iterated, it should not be used as the final result of estimation. Without iteration, the MCS estimator is not a fixed point – the weights are functions only of the sample proportions, not the parameters. For current purposes, these are only useful as starting values.

Standard output for the logit model will begin with a table such as the following which results from estimation of a model in which the dependent variable takes values 0,1,2,3,4,5:

LOGIT ; Lhs = newhsat ; Rhs = one,educ,hhninc,age,hhkids $

```
+---------------------------------------------+
| Multinomial Logit Model                     |
| Maximum Likelihood Estimates                |
| Dependent variable              NEWHSAT     |
| Weighting variable                 None     |
| Number of observations             8140     |
| Iterations completed                  5     |
| Log likelihood function       -11246.97     |
| Number of parameters                 25     |
| Info. Criterion: AIC =          2.76953     |
|   Finite Sample: AIC =          2.76955     |
| Info. Criterion: BIC =          2.79104     |
| Info. Criterion:HQIC =          2.77688     |
| Restricted log likelihood     -11308.02     |
| McFadden Pseudo R-squared      .0053989     |
| Chi squared                    122.1013     |
| Degrees of freedom                   20     |
| Prob[ChiSqd > value] =         .0000000     |
+---------------------------------------------+
```

This is based on the health satisfaction variable analyzed in the preceding chapter. We reduced the sample to those with *newhsat* reported zero to five. We would note, though these make for a fine numerical example, the multinomial logit model would be inappropriate for these ordered data.) The restricted log likelihood is computed for a model in which *one* is the only Rhs variable. In this case,

$$\log L_0 \;=\; \Sigma_j\, n_j \log P_j,$$

where n_j is the number of individuals who choose outcome j and $P_j = n_j/n$ = the jth sample proportion. The chi squared statistic is $2(\log L - \log L_0)$. If your model does not contain a constant term, this statistic need not be positive, in which case it is not reported. But, even if it is, the statistic is meaningless if your model does not contain a constant.

The diagnostic statistics are followed by the coefficient estimates: These are $\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_J$. Recall $\boldsymbol{\beta}_0$ is normalized to zero, and not reported.

```
+--------+--------------+----------------+--------+--------+----------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]| Mean of X|
+--------+--------------+----------------+--------+--------+----------+
---------+Characteristics in numerator of Prob[Y = 1]
 Constant|   -1.77566023       .69486152    -2.555   .0106
 EDUC    |     .07325707       .04476186     1.637   .1017   10.8759203
 HHNINC  |     .28572052       .58129003      .492   .6231    .32998942
 AGE     |     .00565832       .00838172      .675   .4996   46.9925061
 HHKIDS  |     .27187563       .19642471     1.384   .1663    .33169533
---------+Characteristics in numerator of Prob[Y = 2]
 Constant|    -.54216913       .54865993     -.988   .3231
 EDUC    |     .06151644       .03616780     1.701   .0890   10.8759203
 HHNINC  |     .85929376       .44943471     1.912   .0559    .32998942
 AGE     |    -.00089766       .00650574     -.138   .8903   46.9925061
 HHKIDS  |     .13920984       .15529658      .896   .3700    .33169533
---------+Characteristics in numerator of Prob[Y = 3]
 Constant|    -.25432932       .49206457     -.517   .6053
 EDUC    |     .10995580       .03246796     3.387   .0007   10.8759203
 HHNINC  |    1.54516927       .40166793     3.847   .0001    .32998942
 AGE     |    -.00955207       .00583708    -1.636   .1017   46.9925061
 HHKIDS  |     .08177804       .14014086      .584   .5595    .33169533
---------+Characteristics in numerator of Prob[Y = 4]
 Constant|     .09378185       .48301274      .194   .8461
 EDUC    |     .10453491       .03201865     3.265   .0011   10.8759203
 HHNINC  |    1.74362305       .39382043     4.427   .0000    .32998942
 AGE     |    -.01430375       .00571476    -2.503   .0123   46.9925061
 HHKIDS  |     .19548647       .13659829     1.431   .1524    .33169533
---------+Characteristics in numerator of Prob[Y = 5]
 Constant|    1.58458651       .45170179     3.508   .0005
 EDUC    |     .07526768       .03034831     2.480   .0131   10.8759203
 HHNINC  |    1.64030015       .37209397     4.408   .0000    .32998942
 AGE     |    -.01481141       .00525964    -2.816   .0049   46.9925061
 HHKIDS  |     .19988328       .12654882     1.579   .1142    .33169533
```

The prediction for any observation is the cell with the largest predicted probability for that observation.

NOTE: If you have more than three outcomes, it is very common, as occurred above, for the model to predict zero outcomes in one or more of the cells. Even in a model with very high t ratios and great statistical significance, it takes a very well developed model to make predictions in all cells.

The **; List** specification produces a listing such as the following:

Observation	Observed Y	Predicted Y	Residual	MaxPr(i)	Prob[Y*=y]
1	2.0000	.00000	.0000	.2905	.1443
2	.00000	.00000	.0000	.2538	.2538
3	.00000	.00000	.0000	.2866	.2866
4	5.0000	3.0000	.0000	.2532	.1088
5	4.0000	3.0000	.0000	.2535	.2452
6	4.0000	3.0000	.0000	.2584	.2503
7	4.0000	4.0000	.0000	.2568	.2568
8	5.0000	.00000	.0000	.2354	.1440
9	.00000	4.0000	.0000	.2596	.2045
10	1.0000	.00000	.0000	.2554	.1027

In the listing, the MaxPr(i) is the probability attached to the outcome with the largest predicted probability; the outcome is shown as the Predicted Y. The last column shows the predicted probability for the observed outcome. Residuals are not computed – there is no significance to the reported zero.

The results kept for further use are:

Matrices: *b* and *varb*.

An additional matrix named *b_logit* is created which is $(J+1)\times K$. This matrix contains the parameters arranged so that $\boldsymbol{\beta}_j'$ is the *j*th row. The first row is zero. This matrix can be used to obtain fitted probabilities, as discussed below.

Scalars: *kreg*, *nreg*, *logl*, and *exitcode*.

Labels for **WALD** are constructed from the outcome and variable numbers. For example, if there are three outcomes and **; Rhs = one,x1,x2**, the labels will be

Last Model: [*b*1_1,*b*1_2,*b*1_3,*b*2_1,*b*2_2,*b*2_3].

4.6 Marginal Effects

The marginal effects in this model are

$$\boldsymbol{\delta}_j = \partial P_j/\partial \mathbf{x},\ j = 0,1,\ldots,J.$$

For the present, ignore the normalization $\boldsymbol{\beta}_0 = \mathbf{0}$. The notation P_j is used for Prob[y = j]. After some tedious algebra, we find

$$\boldsymbol{\delta}_j = P_j(\boldsymbol{\beta}_j - \bar{\boldsymbol{\beta}}),$$

where $$\bar{\boldsymbol{\beta}} = \sum_{j=0}^{J} P_j \boldsymbol{\beta}_j.$$

It follows that neither the sign nor the magnitude of $\boldsymbol{\delta}_j$ need bear any relationship to those of $\boldsymbol{\beta}_j$. (This is worth bearing in mind when reporting results.) The asymptotic covariance matrix for the estimator of $\boldsymbol{\delta}_j$ would be computed using

$$\text{Asy.Var.}\left[\hat{\boldsymbol{\delta}}_j\right] = \mathbf{G}_j \,\text{Asy.Var}\left[\hat{\boldsymbol{\beta}}_j\right]\mathbf{G}_j',$$

where $\boldsymbol{\beta}$ is the full parameter vector. It can be shown that

$$\text{Asy.Var.}\left[\hat{\boldsymbol{\delta}}_j\right] = \Sigma_l \Sigma_m\ \mathbf{V}_{jl}\,\text{Asy.Cov.}[\ \hat{\boldsymbol{\beta}}_l, \hat{\boldsymbol{\beta}}_m']\mathbf{V}_{jm}',\ j=0,\ldots,J,$$

where $$\mathbf{V}_{jl} = [\mathbf{1}(j = l) - P_l]\{P_j\mathbf{I} + \boldsymbol{\delta}_j\mathbf{x}'\} - P_j\boldsymbol{\delta}_l\mathbf{x}',$$

and $$\mathbf{1}(j = l) = 1 \text{ if } j = l, \text{ and } 0 \text{ otherwise.}$$

This full set of results is produced automatically when your **LOGIT** command includes

; Marginal Effects

There is no conditional mean function in this model, so marginal effects are interpreted a bit differently from the usual case. What is reported are the derivatives of the probabilities. (Note this is the same as the ordered probability models.) These derivatives are saved in a matrix named *partials* which has J+1 rows and K columns. Each row is the vector of partial effects of the corresponding probability. Since the probabilities will always sum to one, the column sums in this matrix will always be zero. That is,

MATRIX ; List ; 1 ' partials $

will display a row matrix of zeros. The elasticities of the probabilities, $(\partial P_j/\partial x_k)\times(x_k/Pj)$ are placed in a $(J+1)\times K$ matrix named *elast_ml*. The format of the results is illustrated in the example below.

```
+-------------------------------------------+
| Partial derivatives of probabilities with |
| respect to the vector of characteristics. |
| They are computed at the means of the Xs. |
| Observations used for means are All Obs.  |
| A full set is given for the entire set of |
| outcomes, NEWHSAT  = 0 to NEWHSAT  =  5.  |
| Probabilities at the mean vector are      |
|  0= .052 1= .030 2= .078 3= .145 4= .171  |
|  5= .523                                  |
+-------------------------------------------+
+--------+--------------+----------------+--------+--------+----------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|Elasticity|
+--------+--------------+----------------+--------+--------+----------+
---------+Marginal effects on Prob[Y = 0]
 Constant|    -.03681271       .02185753    -1.684   .0921
 EDUC    |    -.00415059       .00144841    -2.866   .0042   -.87310224
 HHNINC  |    -.07533229       .01759541    -4.281   .0000   -.48080659
 AGE     |     .00059378       .00025180     2.358   .0184    .53968780
 HHKIDS  |    -.00874507       .00608176    -1.438   .1505   -.05610378
---------+Marginal effects on Prob[Y = 1]
 Constant|    -.07581474       .01624087    -4.668   .0000
 EDUC    |    -.00021399       .00101558     -.211   .8331   -.07636415
 HHNINC  |    -.03569724       .01353007    -2.638   .0083   -.38652184
 AGE     |     .00052245       .00019922     2.622   .0087    .80558651
 HHKIDS  |     .00313091       .00463577      .675   .4994    .03407609
---------+Marginal effects on Prob[Y = 2]
 Constant|    -.09814200       .02502533    -3.922   .0001
 EDUC    |    -.00146816       .00158947     -.924   .3557   -.20405436
 HHNINC  |    -.04677448       .02027747    -2.307   .0211   -.19724874
 AGE     |     .00082844       .00031003     2.672   .0075    .49750446
 HHKIDS  |    -.00234229       .00728521     -.322   .7478   -.00992853
---------+Marginal effects on Prob[Y = 3]
 Constant|    -.13990259       .03064835    -4.565   .0000
 EDUC    |     .00429655       .00187257     2.294   .0218    .32276832
 HHNINC  |     .01275949       .02392200      .533   .5938    .02908292
 AGE     |     .00027978       .00039814      .703   .4822    .09081229
 HHKIDS  |    -.01264824       .00934649    -1.353   .1760   -.02897839
---------+Marginal effects on Prob[Y = 4]
 Constant|    -.10599103       .03277396    -3.234   .0012
 EDUC    |     .00415859       .00200931     2.070   .0385    .26381106
 HHNINC  |     .04913321       .02486677     1.976   .0482    .09457056
 AGE     |    -.00048333       .00042477    -1.138   .2552   -.13248126
 HHKIDS  |     .00451648       .00978660      .461   .6444    .00873817
---------+Marginal effects on Prob[Y = 5]
 Constant|     .45666308       .04483400    10.186   .0000
 EDUC    |    -.00262240       .00279117     -.940   .3475   -.05449699
 HHNINC  |     .09591130       .03450901     2.779   .0054    .06047510
 AGE     |    -.00174112       .00056626    -3.075   .0021   -.15633760
 HHKIDS  |     .01608821       .01313247     1.225   .2205    .01019657
 Marginal Effects Averaged Over Individuals
 --------+---------+---------+---------+---------+---------+---------+
 Variable|    Y=00 |    Y=01 |    Y=02 |    Y=03 |    Y=04 |    Y=05 |
 --------+---------+---------+---------+---------+---------+---------+
 ONE     |  -.0377 |  -.0772 |  -.0975 |  -.1380 |  -.1051 |   .4556 |
 EDUC    |  -.0044 |  -.0002 |  -.0014 |   .0043 |   .0042 |  -.0025 |
 HHNINC  |  -.0786 |  -.0361 |  -.0459 |   .0136 |   .0494 |   .0977 |
 AGE     |   .0006 |   .0005 |   .0008 |   .0003 |  -.0005 |  -.0018 |
```

```
HHKIDS   |  -.0092 |   .0033 |  -.0023 |  -.0125 |   .0045 |   .0162 |
--------+---------+---------+---------+---------+---------+---------+
Averages of Individual Elasticities of Probabilities
--------+---------+---------+---------+---------+---------+---------+
Variable|    Y=00 |    Y=01 |    Y=02 |    Y=03 |    Y=04 |    Y=05 |
--------+---------+---------+---------+---------+---------+---------+
ONE     |  -.7050 | -2.4807 | -1.2472 |  -.9593 |  -.6112 |   .8796 |
EDUC    |  -.8732 |  -.0764 |  -.2041 |   .3227 |   .2638 |  -.0545 |
HHNINC  |  -.4847 |  -.3904 |  -.2011 |   .0252 |   .0907 |   .0566 |
AGE     |   .5315 |   .7974 |   .4894 |   .0827 |  -.1406 |  -.1645 |
HHKIDS  |  -.0571 |   .0330 |  -.0110 |  -.0300 |   .0077 |   .0092 |
--------+---------+---------+---------+---------+---------+---------+
```

Matrix - B_LOGIT

[6, 5] Cell:

	1	2	3	4	5
1	0	0	0	0	0
2	-1.77566	0.0732571	0.285721	0.00565832	0.271876
3	-0.542169	0.0615164	0.859294	-0.000897661	0.13921
4	-0.254329	0.109956	1.54517	-0.00955207	0.081778
5	0.0937819	0.104535	1.74362	-0.0143038	0.195486
6	1.58459	0.0752677	1.6403	-0.0148114	0.199883

Matrix - ELAST_ML

[6, 5] Cell: -0.712012

	1	2	3	4	5
1	-0.712012	-0.873102	-0.480807	0.539688	-0.0561038
2	-2.48767	-0.0763641	-0.386522	0.805587	0.0340761
3	-1.25418	-0.204054	-0.197249	0.497504	-0.00992853
4	-0.966341	0.322768	0.0290829	0.0908123	-0.0289784
5	-0.61823	0.263811	0.0945706	-0.132481	0.00873817
6	0.872575	-0.054497	0.0604751	-0.156338	0.0101966

Matrix - PARTIALS

[6, 5] Cell: -0.0347176

	1	2	3	4	5
1	-0.0347176	-0.00354578	-0.065477	0.000527146	-0.00750032
2	-0.0512738	-0.000156155	-0.024176	0.000353446	0.00200267
3	-0.0843077	-0.00123466	-0.0402582	0.000704008	-0.00203835
4	-0.146084	0.00402383	0.00667303	0.00034874	-0.0125056
5	-0.123917	0.00415089	0.0436	-0.00039811	0.00424322
6	0.4403	-0.00323814	0.0796381	-0.00153523	0.0157984

Figure 4.1 Matrices Computed by MLOGIT

Marginal effects are computed by averaging the effects over individuals rather than computing them at the means. The difference between the two is likely to be quite small. Current practice favors the averaged individual effects, rather than the effects computed at the means. **MLOGIT** also reports elasticities with the marginal effects. An example appears below.

4.7 Computing Predicted Probabilities

Predicted probabilities can be computed automatically for the multinomial logit model. Since there are multiple outcomes, this must be handled a bit differently from other models. The procedure is as follows: Request the computation with

; Prob = name

as you would normally for a discrete choice model. However, for this model, *NLOGIT* does the following:

1. A namelist is created with name consisting of up to the first four letters of '*name*' and *prob* is appended to it. Thus, if you use **; Prob = pfit**, the namelist will be named *pfitprob*.

2. The set of variables, one for each outcome, are named with the same convention, with *prjj* instead of *prob*.

For example, in a five outcome model, the specification

; Prob = job

produces a namelist

jpbprob = *jobpr*00, *jobpr*01, *jobpr*02, *jobpr*03, *jobpr*04.

The variables will then contain the respective probabilities. You may also use

; Fill

with this procedure to compute probabilities for observations that were not in the sample. Observations which contain missing data are bypassed as usual.

You can also compute a vector of probabilities for a specific observation, for example the sample means, by using the matrix *b_logit*. The following suggests how this might be done using the group means

```
NAMELIST   ; x    = the Rhs variables $
MATRIX     ; xb = Mean(x) $
MATRIX     ; pvec = b_logit*xb
           ; pvec = Expn(pvec)
           ; pvec = <1'pvec> * pvec $
```

Chapter 5: Data Setup for *NLOGIT*

5.1 Introduction

In general, the data for the models described in Parts III and IV will be arranged in a format that is set up to work well with the specific *NLOGIT* estimators. In almost all cases, the data used for all models that you fit with *NLOGIT* will be set up as if they were a panel. That is, each individual 'observation' will have a set of observations, with one 'line' of data for each choice in the choice set. Thus, in the analogy to a panel, the 'group' is a person and the group size would be the number of choices. You will use this arrangement in nearly all cases. This chapter will explain the various aspects of setting up the data for the *NLOGIT* models.

5.2 Basic Data Setup for *NLOGIT*

In the base case, the data are arranged as follows, where we use a specific set of values for the problem to illustrate. Suppose you observe 25 individuals. Each individual in the sample faces three choices and there are two attributes, q and w. For each observation, we also observe which choice was made. Suppose further that in the first three observations, the choices made were two, three, and one, respectively. The data matrix would consist of 75 rows, with 25 blocks of three rows. Within each block, there would be the set of attributes and a variable y, which, at each row, takes the value one if the alternative is chosen and zero if not. Thus, within each block of J rows, y will be one once and only once. For the hypothetical case, then, we have:

	Y	Q	W
i=1	0	$q_{1,1}$	$w_{1,1}$
→	1	$q_{2,1}$	$w_{2,1}$
	0	$q_{3,1}$	$w_{3,1}$
i=2	0	$q_{1,2}$	$w_{1,2}$
	0	$q_{2,2}$	$w_{2,2}$
→	1	$q_{3,2}$	$w_{3,2}$
i=3 →	1	$q_{1,3}$	$w_{1,3}$
	0	$q_{2,3}$	$w_{2,3}$
	0	$q_{3,3}$	$w_{3,3}$

and so on, continuing to $i = 25$, where → marks the row of the respondent's actual choice.

When you read these data, the data set is not treated any differently from any other panel. *Nobs* would be the total number of rows in the data set, in the hypothetical case, 75, not 25. The separation of the data set into the above groupings would be done at the time your particular model is estimated.

NOTE: Missing values are handled automatically by estimation programs in *NLOGIT*. You should not reset the sample or use **SKIP** with the *NLOGIT* models. Observations that have missing values are bypassed as a group.

Thus far, it is assumed that the observed outcome is an indicator of which choice was made among a fixed set of up to 100 choices. Numerous variations on this are possible:

- Data on the observed outcome may be in the form of frequencies, market shares, or ranks.
- The number of choices may differ across observations.
- The choice set may be extremely large.

The preceding described the base case model for a fixed number of choices using individual level data. There are several alternative formulations that might apply to the data set you are using.

5.3 Fixed and Variable Numbers of Choices

When every individual in the sample chooses from the same choice set, and all alternatives are available to all individuals, then the data set will appear as in the first example above, and will consist of *n* sets of *J* 'observations.' You indicate this case with a command such as:

```
     NLOGIT      ; Lhs = y
or   CLOGIT      ; Choices = ... a list of J names for the choices
or   ...         ; ... the rest of the command  $
```

For convenience in what follows, we will use the generic model name **NLOGIT** in the command. The specific verbs, **CLOGIT**, **RLPOGIT**, etc. will be used in the specific chapters where the model itself is developed.) For example,

```
NLOGIT      ; Lhs = mode
            ; Choices = air,train,bus,car
            ; ... the rest of the command $
```

The list of choices is crucial, as it tells the program how many choices constitute an observation. (Otherwise, for example, there is no way to tell if 12 rows of data are three observations on a four choice setting or four observations on a three choice setting.)

We now consider the random utility model first in which the number of choices is not constant from one observation to the next. Two possible arrangements that might occur are as follows:

- There is a 'universal choice set,' from which individuals make their choices. But, not all choices are available to all individuals. Consider, for example, the choice of travel mode among (*air, train, bus, car*). If respondents are observed at many different locations, one or more of the choices, for example, *train*, might be unavailable to some of them, and those might vary from person to person.

- Individuals each choose among a set of J_i alternatives. However, there is no universal choice set defined as such. Consider, for example, the choice of which shopping center to shop at. If observations are taken in many different cities, we will observe numerous different choice sets, but there is no well defined universal choice set.

Either case can be accommodated. For both cases, you will provide a second **; Lhs** variable which gives the number of choices for each observation. The command is

```
NLOGIT      ; Lhs = y,nij
            ; ... specification of the utility functions
            ; ... the rest of the command $
```

Note that the **; Choices = list** is not defined in this command, since in this case (the second one above), there is no clearly defined choice set. Nothing else need be changed. *NLOGIT* does all of the accounting internally. In this case, it is simply assumed that each individual has his or her own choice set. For example, one such data set might appear as follows.

	Y	*Q*	*W*	*Nij*
i=1	0	$q_{1,1}$	$w_{1,1}$	3
→	1	$q_{2,1}$	$w_{2,1}$	3
	0	$q_{3,1}$	$w_{3,1}$	3
i=2	0	$q_{1,2}$	$w_{1,2}$	4
	0	$q_{2,2}$	$w_{2,2}$	4
	1	$q_{3,2}$	$w_{3,2}$	4
→	0	$q_{4,2}$	$w_{4,2}$	4
i=3	1	$q_{1,3}$	$w_{1,3}$	2
→	0	$q_{2,3}$	$w_{2,3}$	2

The model command might be

```
NLOGIT      ; Lhs = y,nij
            ; Rhs = q,w $
```

Notice, once again, that the command does not contain a definition of the choice set, such as **; Choices = list** specification.

For the case of a universal choice set, suppose that the data set were, instead:

	Y	*Q*	*W*	*Nij*	*Altij*
i=1	0	$q_{1,1}$	$w_{1,1}$	3	1 (*Air*)
→	1	$q_{2,1}$	$w_{2,1}$	3	2 (*Train*)
	0	$q_{3,1}$	$w_{3,1}$	3	4 (*Car*)
i=2	0	$q_{1,2}$	$w_{1,2}$	4	1 (*Air*)
	0	$q_{2,2}$	$w_{2,2}$	4	2 (*Train*)
	1	$q_{3,2}$	$w_{3,2}$	4	3 (*Bus*)
	0	$q_{4,2}$	$w_{4,2}$	4	4 (*Car*)
i=3 →	1	$q_{1,3}$	$w_{1,3}$	2	3 (*Bus*)
	0	$q_{2,3}$	$w_{2,3}$	2	4 (*Car*)

The specific choice identifier, when it is needed, is provided as a *third* Lhs variable. For this case, the choice set would have to be defined. For example,

```
NLOGIT      ; Lhs = y, nij, altij
            ; Choices = air,train,bus,car $
            ; Rhs = q,w $
```

Once again, in this setting, every individual is assumed to choose from a set of four alternatives, though the *altij* variable indicates that some of these choices are unavailable to some individuals.

Do note that if you are not defining a universal choice set, *NLOGIT* simply uses the largest number of choices for any individual in the sample to determine *J*. As such, an expanded set of choice specific constants is not likely to be meaningful. Also, in the absence of a universal choice set, the variable *altij* will not be meaningful.

The IIA test described later is carried out by fitting the model to a restricted choice set, then comparing the two sets of parameter estimates. You can restrict the choice set used in estimation, irrespective of the IIA test, by a slight change in the command. In the **; Choices = list of alternatives** specification, enclose any choices to be excluded in parentheses. For example, in our **CLOGIT** application, the specification

; Choices = air, (train), (bus), car

produces (in part – most of the results are omitted) the following display in the model output:

```
+-----------------------------------------------------+
|WARNING:   Bad observations were found in the sample. |
|Found  93 bad observations among     210 individuals. |
|You can use ;CheckData to get a list of these points. |
+-----------------------------------------------------+
Sample proportions are marginal, not conditional.
Choices marked with * are excluded for the IIA test.
+----------------+------+---+
|Choice   (prop.)|Weight|IIA|
+----------------+------+---+
|AIR       .49573| 1.000|   |
|TRAIN     .00000| 1.000| * |←
|BUS       .00000| 1.000| * |
|CAR       .50427| 1.000|   |
+----------------+------+---+
+-----------------------------------------------------------------+
| Model Specification:  Table entry is the attribute that         |
| multiplies the indicated parameter.                             |
+--------+------+-----------------------------------------------+
| Choice |******| Parameter                                     |
|        |Row  1| INVC     INVT     GC       TTME     A_AIR      |
|        |Row  2| A_TRAIN  A_BUS                                 |
+--------+------+-----------------------------------------------+
|AIR     |     1| INVC     INVT     GC       TTME     Constant   |
|        |     2| none     none                                  |
|TRAIN   |     1| INVC     INVT     GC       TTME     none       |
|        |     2| Constant none                                  |
|BUS     |     1| INVC     INVT     GC       TTME     none       |
|        |     2| none     Constant                              |
|CAR     |     1| INVC     INVT     GC       TTME     none       |
|        |     2| none     none                                  |
+-----------------------------------------------------------------+
Normal exit from iterations. Exit status=0.
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Maximum Likelihood Estimates                |
| Dependent variable               Choice     |
| Weighting variable                 None     |
| Number of observations              117     |
| Iterations completed                  6     |
| Log likelihood function       -52.79148     |
| Number of parameters                  5     |
```

```
| Info. Criterion: AIC =           .98789     |
|   Finite Sample: AIC =           .99251     |
| Info. Criterion: BIC =          1.10593     |
| Info. Criterion:HQIC =          1.03581     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only     -81.0939  .34901  .31995 |
| Chi-squared[ 4]          =     56.60494     |
| Prob [ chi squared > value ] =   .00000     |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped  93 bad obs. |
+---------------------------------------------+
| Restricted choice set. Excluded choices are |
| TRAIN    BUS                                |
+---------------------------------------------+
+---------------------------------------------+
| Notes No coefficients=> P(i,j)=1/J(i).      |
|       Constants only => P(i,j) uses ASCs    |
|         only. N(j)/N if fixed choice set.   |
|         N(j) = total sample frequency for j |
|         N    = total sample frequency.      |
|       These 2 models are simple MNL models. |
|       R-sqrd = 1 - LogL(model)/logL(other)  |
|       RsqAdj=1-[nJ/(nJ-nparm)]*(1-R-sqrd)   |
|         nJ   = sum over i, choice set sizes |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 INVC    |    -.04871233       .02756765    -1.767   .0772
 INVT    |    -.01195151       .00394602    -3.029   .0025
 GC      |     .08575924       .02654046     3.231   .0012
 TTME    |    -.08221552       .01854075    -4.434   .0000
 A_AIR   |    2.12899069      1.20530610     1.766   .0773
 A_TRAIN |       .000000    ......(Fixed Parameter).......
 A_BUS   |       .000000    ......(Fixed Parameter).......
```

Note that as in the IIA test, this procedure results in exclusion of some ‘bad’ observations, that is, the ones that selected the excluded choices. Because of the model specification, the ASCs for bus and train have been fixed at zero.

You may combine the choice based sampling estimator with the restricted choice set. All the necessary adjustments of the weights are made internally. Thus, the specification

; Choices = air,(train),(bus),car / .14,.13,.09,.64

produces the following listing:

```
+----------------+------+---+
|Choice   (prop.)|Weight|IIA|
+----------------+------+---+
|AIR       .49573|  .387|   |
|TRAIN     .00000|  .000| * |
|BUS       .00000|  .000| * |
|CAR       .50427| 1.739|   |
+----------------+------+---+
```

5.4 Types of Data on the Choice Variable

We allow several types of data on the choice variable, *y*. If you have *grouped data*, the values of *y* will be *proportions* or *frequencies*, instead of *individual choices*. In the first case, within each observation (*J* data points), the values of *y* will sum to one when summed *down* the *J* rows. (This will be the only difference in the grouped data treatment.) In the second case, *y* will simply be a set of nonnegative integers. An example of a setting in which such data might arise would be in marketing, where the proportions might be market shares of several brands of a commodity. Or, the data might be counts of responses to particular questions in a survey in which groups of people in different locations or at different times were surveyed. Finally, *y* might be a set of *ranks*, in which case, instead of zeros and ones, *y* would take values 1,2,...,*J* (not necessarily in that order) within, and reading down, each block.

More specifically, data on the dependent (Lhs) variable may come in these four forms:

- **Individual Data**: The Lhs variable consists of zeros and a single one which indicates the choice that the individual made. When data are individual, the observations on the Lhs variable will sum exactly to 1.0 for every person in the sample. A sum of 0.0 or some other value will only arise if a data error has occurred. Individual choice data may also be simulated.

- **Proportions Data**: The Lhs variable consists of a set of sample proportions. Values range from zero to one, and again, they sum to 1.0 over the set of choices in the choice set. Observed proportions may equal 1.0 or 0.0 for some individuals.

- **Frequency Data**: The Lhs variable consists of a set of frequency counts for the outcomes. Frequencies are nonnegative integers for the outcomes in the choice set and may be zero.

- **Ranks Data**: The Lhs variable consists of a complete set of ranks of the alternatives in the individual's choice set. Thus, if there are *J* alternatives available, the observation will consist of a full set of the integers 1,...,*J* not necessarily in that order, which indicate the individual's ranking of the alternatives. The number of choices may still differ by observation. Thus, we might have [(unranked),0,1,0,0,0] in the usual case, and [(ranked) 4,1,3,2,5] with ranks data. Note that the positions of the ones are the same for both sets, by definition. You may also have partial rankings. For example, suppose respondents are given 10 choices and asked to rank their top three. Then, the remaining six choices should be coded 4.0. A set of ranks might appear thusly: [1,4,2,4,3,4,4,4,4,4]. The ties must only appear at the lowest level. Ties in the data are detected automatically. No indication is needed. For later reference, we note the following for the model based on ranks data:

    - You may have observation weights, but no choice based sampling.
    - The IIA test described in Chapter 8 is not available.
    - The number of choices may be fixed or variable, as described above.
    - You may keep probabilities or inclusive values as described in Chapter 7.
    - Ranks data may only be used with the conditional logit model (**CLOGIT**) and the mixed logit (random parameters) model (**RPLOGIT**).

The first three data types are detected automatically by *NLOGIT*. You do not have to give any additional information about the data set, since the type of data being provided can usually be deduced from the values. (See below for one exception.) The ranks data are an exception for which you would use

NLOGIT ; ... as usual ...; Ranks $

If you are using frequency or proportions data, and your data contain zeros or ones, certain kinds of observations cannot be distinguished from erroneous individual data, and they may be flagged as such. For example, in a frequency data set, the observation [0,0,1,1,0,0] is a valid observation, but for individual data, it looks like a badly coded observation. In order to avoid this kind of ambiguity, if you have frequency data containing zeros, add

; Frequencies

to your **NLOGIT** command. (You may use this in any event to be sure that the data are always recognized correctly.) If you have proportions data, instead, you may use

; Shares

to be sure that the data are correctly marked. (Again, this will only be relevant if your data contain zeros and/or ones.)

Data are checked for validity and consistency. An unrecognizable mixture of the three types will cause an error. For example, a mixture of frequency and proportions data cannot be properly analyzed. For the ranks data, an error will occur if the set of ranks is miscoded or incomplete or if ties are detected at any ranks other than the lowest.

5.5 Data for the Applications

The documentation of the *NLOGIT* program in the chapters to follow includes numerous applications based on the data set CLOGIT.DAT, that is distributed with *NLOGIT*. These data are a survey of the transport mode chosen by a sample of 210 travelers between Sydney and Melbourne (about 500 miles) and other points in nonmetropolitan New South Wales. As discussed in Section 5.2, data for *NLOGIT* will generally consist of a record (row of data) for each alternative in the choice set, for each individual. Thus, the data file contains 210 observations, or 840 records. The variables in the data set are as follows:

Original Data

mode	=	0/1 for four alternatives: air, train, bus, car (this variable equals one for the choice made, labeled *choice* below),
ttme	=	terminal waiting time,
invc	=	invehicle cost for all stages,
invt	=	invehicle time for all stages,
gc	=	generalized cost measure = Invc + Invt × value of time,
chair	=	dummy variable for chosen mode is air,
hinc	=	household income in thousands,
psize	=	traveling party size.

Transformed variables

aasc = choice specific dummy for air (generated internally),
tasc = choice specific dummy for train,
basc = choice specific dummy for bus,
casc = choice specific dummy for car,
hinca = *hinc* × *aasc*,
psizea = *psize* × *aasc*.

The table below lists the first 10 observations in the data set. In the terms used here, each 'observation' is a block of four rows. The mode chosen in each block is boldfaced.

mode	*choice*	*ttme*	*invc*	*invt*	*gc*	*chair*	*hinc*	*psize*	*aasc*	*tasc*	*basc*	*casc*	*hinca*	*psizea*	*obs.*
Air	0	69	59	100	70	0	35	1	1	0	0	0	35	1	*i*=1
Train	0	34	31	372	71	0	35	1	0	1	0	0	0	0	
Bus	0	35	25	417	70	0	35	1	0	0	1	0	0	0	
Car	**1**	**0**	**10**	**180**	**30**	**0**	**35**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	
Air	0	64	58	68	68	0	30	2	1	0	0	0	30	2	*i*=2
Train	0	44	31	354	84	0	30	2	0	1	0	0	0	0	
Bus	0	53	25	399	85	0	30	2	0	0	1	0	0	0	
Car	**1**	**0**	**11**	**255**	**50**	**0**	**30**	**2**	**0**	**0**	**0**	**1**	**0**	**0**	
Air	0	69	115	125	129	0	40	1	1	0	0	0	40	1	*i*=3
Train	0	34	98	892	195	0	40	1	0	1	0	0	0	0	
Bus	0	35	53	882	149	0	40	1	0	0	1	0	0	0	
Car	**1**	**0**	**23**	**720**	**101**	**0**	**40**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	
Air	0	64	49	68	59	0	70	3	1	0	0	0	70	3	*i*=4
Train	0	44	26	354	79	0	70	3	0	1	0	0	0	0	
Bus	0	53	21	399	81	0	70	3	0	0	1	0	0	0	
Car	**1**	**0**	**5**	**180**	**32**	**0**	**0**	**3**	**0**	**0**	**0**	**1**	**0**	**0**	
Air	0	64	60	144	82	0	45	2	1	0	0	0	45	2	*i*=5
Train	0	44	32	404	93	0	45	2	0	1	0	0	0	0	
Bus	0	53	26	449	94	0	45	2	0	0	1	0	0	0	
Car	**1**	**0**	**8**	**600**	**99**	**0**	**45**	**2**	**0**	**0**	**0**	**1**	**0**	**0**	
Air	0	69	59	100	70	0	20	1	1	0	0	0	20	1	*i*=6
Train	**1**	**40**	**20**	**345**	**57**	**0**	**20**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	
Bus	0	35	13	417	58	0	20	1	0	0	1	0	0	0	
Car	0	0	12	284	43	0	20	1	0	0	0	1	0	0	
Air	**1**	**45**	**148**	**115**	**160**	**1**	**45**	**1**	**1**	**0**	**0**	**0**	**45**	**1**	*i*=7
Train	0	34	111	945	213	1	45	1	0	1	0	0	0	0	
Bus	0	35	66	935	167	1	45	1	0	0	1	0	0	0	
Car	0	0	36	821	125	1	45	1	0	0	0	1	0	0	
Air	0	69	121	152	137	0	12	1	1	0	0	0	12	1	*i*=8
Train	0	34	52	889	149	0	12	1	0	1	0	0	0	0	
Bus	0	35	50	879	146	0	12	1	0	0	1	0	0	0	
Car	**1**	**0**	**50**	**780**	**135**	**0**	**12**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	
Air	0	69	59	100	70	0	40	1	1	0	0	0	40	1	*i*=9
Train	0	34	31	372	71	0	40	1	0	1	0	0	0	0	
Bus	0	35	25	417	70	0	40	1	0	0	1	0	0	0	
Car	**1**	**0**	**17**	**210**	**40**	**0**	**40**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	
Air	0	69	58	68	65	0	70	2	1	0	0	0	70	2	*i*=10
Train	0	34	31	357	69	0	70	2	0	1	0	0	0	0	
Bus	0	35	25	402	68	0	70	2	0	0	1	0	0	0	
Car	**1**	**0**	**7**	**210**	**30**	**0**	**70**	**2**	**0**	**0**	**0**	**1**	**0**	**0**	

Chapter 6: *NLOGIT* Commands and Results

6.1 Introduction

This chapter will describe the common features of the *NLOGIT* models. The specification of models for *NLOGIT* follows the general pattern for model commands in *LIMDEP*. The different models, such as nested logit and multivariate probit, are requested by modifying the basic command.

NLOGIT is built around estimation of the parameters of the random utility model for discrete choice,

$$U(\text{choice } j \text{ for individual } i) = U_{ij} = \boldsymbol{\beta}_{ij}'\mathbf{x}_{ij} + \varepsilon_{ij}, j = 1,...,J_i,$$

in which individual i makes choice j if U_{ij} is the largest among the J_i utilities in the choice set. The parameters in the model are the weights in the utility functions and the deeper parameters of the distribution of the random terms. In some cases, the 'taste' parameters in the utility functions might vary across individuals and in most cases, they will vary across choices. The latter is simple to accommodate just by merging all parameters into one grand $\boldsymbol{\beta}$ and redefining $\mathbf{x}$ with some zeros in the appropriate places. But, for the former case, we will be interested in a lower level parameterization that involves what are sometimes labeled the 'hyperparameters.' Thus, it might be the extreme case (as in the random parameters logit model) that $\boldsymbol{\beta}_{ij} = \mathbf{f}(\mathbf{z}_i, \boldsymbol{\Delta}, \boldsymbol{\Gamma}, \boldsymbol{\beta}, \mathbf{v}_i)$ where $\boldsymbol{\Delta}, \boldsymbol{\Gamma}, \boldsymbol{\beta}$ are lower level parameters, $\mathbf{z}_i$ is observed data, and $\mathbf{v}_i$ is a set of latent unobserved variables. The parameters of the random terms will generally be few in number, usually consisting of a small number of scaling parameters as in the heteroscedastic logit model, but they might be quite numerous, again in the random parameters model. In all cases, the main function of the routines is estimation of the structural parameters, then use of the estimated model for analysis of individual and aggregate behavior.

6.2 *NLOGIT* Commands

The essential command for the set of discrete choice models in *NLOGIT* is the same for all, with the exception of the model name:

```
Model      ; Lhs = variable which indicates the choice made
           ; Choices = a set of J names for the set of choices
           ; Rhs = choice varying attributes in the utility functions
           ; Rh2 = choice invariant variables, including one for ASCs $
```

The various models are as follows, where either of the two forms given may be used:

Model	Command	Alternative Command Form
Conditional Logit	**CLOGIT**	**NLOGIT**
Nested Logit	**NLOGIT**	**NLOGIT ; Tree = ...**
Random Parameters Logit	**RPLOGIT**	**NLOGIT ; RPL**
Multinomial Probit	**MNPROBIT**	**NLOGIT ; MNP**

The description to follow in the rest of this chapter applies equally to all models. For convenience, we will use the generic **NLOGIT** command in most of the discussion, while you can use the specific model names in your estimation commands.

The commands builders for these models can be found in Model:Discrete Choice. There are several model options as shown in Figure 6.1

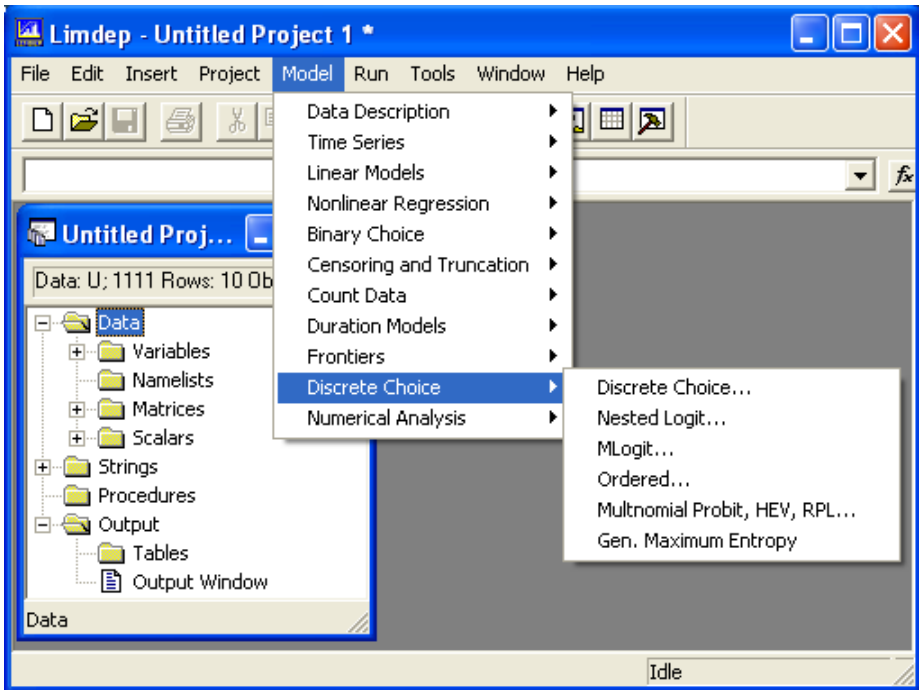


Figure 6.1 Command Builders for *NLOGIT* Models

The Main and Options pages of the command builder for the conditional logit model are shown in Figures 6.2a and 6.2b. (Some features of the models, and the ECM model, are not provided by the command builders. Most of the features of these models are much easier to specify in the editor or command mode of entry.) The model and the choice set are set up on the Main page. The Rhs variables (attributes) and Rh2 variables (characteristics) are defined on the Options page. Note in the two windows on the Options page, the Rhs variables of the model are defined in the left window and the Rh2 variables are specified in the right window.

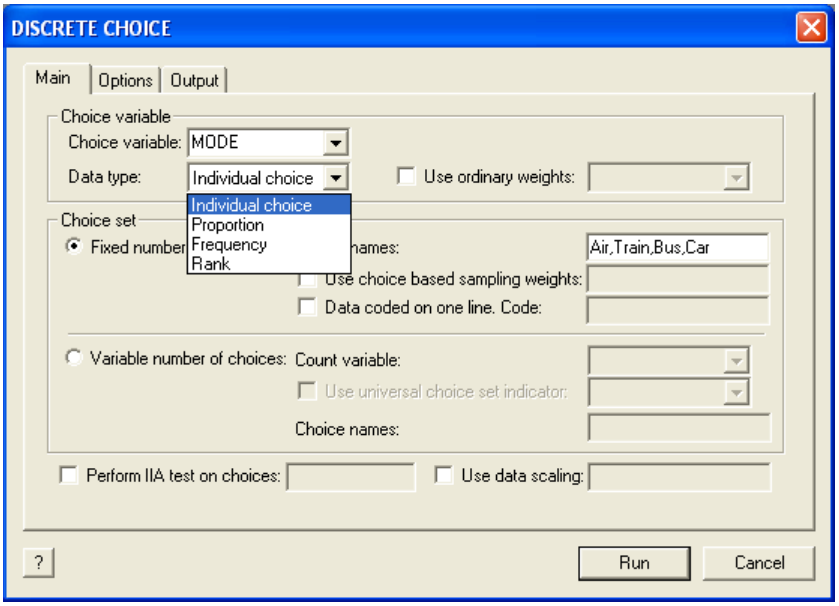


Figure 6.2a Main Page of Command Builder for Conditional Logit Model

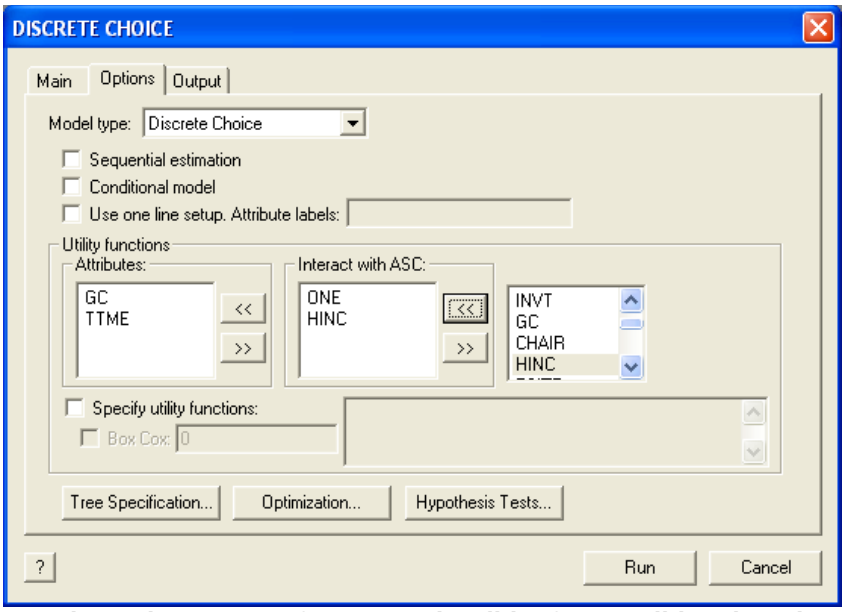


Figure 6.2b Options Page of Command Builder for Conditional Logit Model

A set of exactly *J* choice labels must be provided in the command. These are used to label the choices in the output. The number you provide is used to determine the number of choices there are in the model. Therefore, the set of the right number of labels is essential. Use any descriptor of eight or fewer characters desired – these do not have to be valid names, just a set of labels, separated in the list by commas.

The internal limit on J, the number of choices, is 100.

There are K attributes (Rhs variables) measured for the choices. The sections below will describe variations of this for different formulations and options. The total number of parameters in the utility functions will include K_1 for the Rhs variables and $(J\text{-}1)K_2$ for the Rh2 variables. The total number of utility function parameters is thus $K = K_1 + (J\text{-}1)K_2$.

The internal limit on K, the number of utility function parameters, is 100.

The random utility model specified by this setup is precisely of the form

$$U_{i,j} = \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_{K1} x_{i,K1} + \gamma_{1,j} z_{i,1} + \ldots + \gamma_{K2,j} z_{i,K2} + \varepsilon_{i,j},$$

where the x variables are given by the Rhs list and the z variables are in the Rh2 list. By this specification, the same attributes and the same characteristics appear in all equations, at the same position. The parameters, β_k appear in all equations, and so on. There are various ways to change this specification of the utility functions – i.e., the Rhs of the equations that underlie the model, and several different ways to specify the choice set. These will be discussed at several points below.

6.2.1 Other Optional Specifications on *NLOGIT* Commands

The **NLOGIT** command operates like other *LIMDEP* model commands. The following lists command features and options that may or may not be specified with the **NLOGIT** command. Features marked with '*' are unavailable with this command.

Controlling Output from Model Commands

	; Par	use with the random parameters logit model to save person specific parameter vectors.
	; Margin	displays marginal effects. (Use **; Effects: specification**.)
*	**; OLS**	displays least squares starting values. Not used here.
	; Table = name	saves model results to be combined later in output tables.

Robust Asymptotic Covariance Matrices

	; Printvc	displays estimated asymptotic covariance matrix (normally not shown).
*	**; Choice**	uses choice based sampling (sandwich with weighting) estimated matrix. (This is specified in the **; Choices = list** specification for **NLOGIT**.)
	; Cluster = name	cluster form of corrected covariance estimator.
*	**; Robust**	sandwich estimator or robust VC for TSCS and some discrete choice.

Optimization Controls for Nonlinear Optimization

; Start = list	gives starting values for a nonlinear model.
; Tlg[= value]	sets convergence value for gradient.
; Tlf[= value]	sets convergence value for function.
; Tlb[= value]	sets convergence value for parameters.
; Alg = name	algorithm. Newton's method is best. BFGS is occasionally needed.
; Maxit = n	maximum iterations.

	; Output = n	technical output.
*	**; Lpt = n**	Laguerre quadrature, number of points to use.
*	**; Hpt = n**	Hermite quadrature, number of points to use.
	; Set	keeps current setting of optimization parameters as permanent.

Predictions and Residuals

	; List	displays a list of estimated probabilities with model results.
*	**; Keep = name**	keeps fitted values as a new (or replacement) variable in data set. (Several other similar specifications are used with **NLOGIT**.)
*	**; Res = name**	keeps residuals as a new (or replacement) variable.
	; Prob = name	saves probabilities as a new (or replacement) variable.
*	**; Fill**	fills missing values (outside estimating sample) for fitted values.

Hypothesis Tests and Restrictions

	; Wald:spec	Wald test of linear restrictions in any model.
	; CML:spec	constrained maximum likelihood.
	; Test:spec	Wald test of linear restrictions - same as **Wald:spec**.
*	**; Rst = list**	specifies equality and fixed value restrictions.
	; Maxit = 0 ; Start = the restricted values	specifies Lagrange multiplier test.

6.2.2 Specifying the Choice Variable and the Choice Set

Every model fit by *NLOGIT* must include a specification for the choice variable and a definition of the choice set. The basic formulation would appear as

; Lhs = the dependent, or choice variable
; Choices = the names of the choices in the model

In general, your dependent variable is the name of a variable which indicates by a one or zero whether a particular alternative is selected, or it gives the proportion or frequency of individuals sampled that selected a particular alternative. When they are enumerated, the **; Choices** list gives names and possibly sampling weights for the set of alternatives.

All command builders begin with these two specifications. The discrete choice and nested logit models allow the full set of variants discussed earlier while the other command builders expect the simple form with a fixed choice set. The Main page of the conditional logit command builder shown in Figure 6.3 illustrates. (A similar Main page is used for the nested logit command builder.) The command builder allows you to specify the choice variable and type of choice set in the three sections of this dialog box.

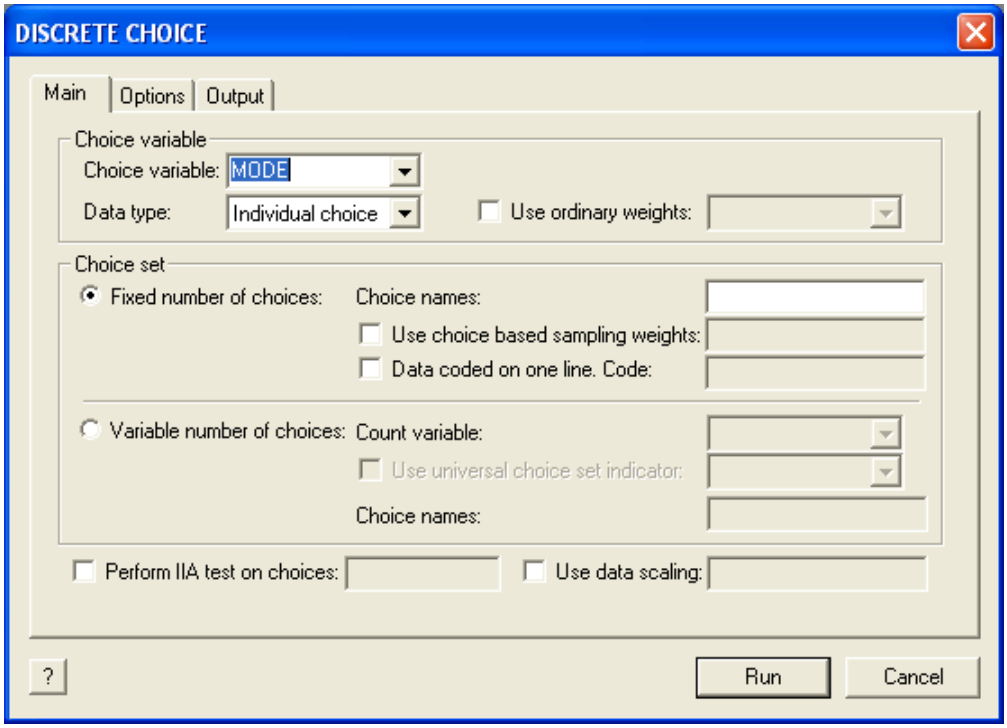


Figure 6.3 Main Page of Command Builder for Conditional Logit Model

NOTE: The command builder for the multinomial probit, HEV and RPL models requires you to provide a fixed sized choice set. This is a limitation of the command builder window, not the estimator. With the exception of the multinomial probit model, this is not a requirement of the models themselves. Only the multinomial probit model requires the number of choices to be fixed. For the HEV and RPL models, if you build your command in the text editor, rather than with the command builder, you may specify a variable choice set.

6.2.3 Restricting the Choice Set

The IIA test described in the Chapter 8 is carried out by fitting the model to a restricted choice set, then comparing the two sets of parameter estimates. You can restrict the choice set used in estimation, irrespective of IIA, by a slight change in the command. In the **; Choices = list of alternatives** specification, enclose any choices to be excluded in parentheses. For example, the specification

; Choices = air, (train), (bus), car

produces (in part – the results are omitted) the following display in the model output:

```
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped  93 bad obs. |
| Restricted choice set. Excluded choices are |
| TRAIN    BUS                                |
+---------------------------------------------+
```

Note that as in the IIA test, this procedure results in exclusion of some 'bad' observations, that is, the ones that selected the excluded choices. The model specified is fit to the data set that is based on the remaining choices. We note a caution at this point that applies equally here and in the IIA test. It is possible that there are attributes that do not vary within the retained choice set. If these remain in the model, it will not be possible to fit it. Consider, for example, a six choice model with choices *air, train, bus, car as driver, car as passenger, motorcycle*. Now, suppose that one of the attributes is *terminal time*. It will happen that the last three choices always have terminal time equal to zero. So, while it may be no trouble to fit a six choice model which includes terminal time, the same model specified with

; Choices = (air), (train), (bus), car_d, car_p, mc

will not be estimable, as terminal time will always be zero for all choices for all individuals.

6.2.4 Specifying the Utility Functions with Rhs and Rh2

There are several ways to specify the utility functions in your **NLOGIT** command, in the text editor and in the command builder. In order to provide a simple explanation that covers the cases, we will develop the application that will be used in the chapters to follow to illustrate the models. The application is based on the data summarized in Section 5.5. We will model travel mode choice for trips between Sydney and Melbourne with utility functions for the four choices as follows:

	gc	*ttme*	*one*	*hinc*	*one*	*hinc*	*one*	*hinc*	*one*	*hinc*
U(*air*) =	GC	TTME	A_AIR	AIR_HIN1	0	0	0	0	0	0
U(*train*) =	GC	TTME	0	0	A_TRAIN	TRA_HIN2	0	0	0	0
U(*bus*) =	GC	TTME	0	0	0	0	A_BUS	BUS_HIN3	0	0
U(*car*) =	GC	TTME	0	0	0	0	0	0	0	0

The columns are headed by the names of variables, generalized cost (*gc*), terminal time (*ttme*) and household income (*hinc*). The entries in the body of the table are the names given to coefficients that will multiply the variables. Note that the generic coefficients in the first two columns are given the names of the variables they multiply while the interactions with the constants are given compound names. It is important to note the last two columns. The last one in a set of choice specific constants or variables that are interacted with them must be dropped to avoid a problem of collinearity in the model. In what follows, for brevity, we will omit these two columns. Before proceeding, we note the format of a set of parameter estimates for a model set up in exactly this fashion:

```
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 GC      |    -.01092735       .00458775    -2.382   .0172
 TTME    |    -.09546055       .01047320    -9.115   .0000
 A_AIR   |    5.87481336       .80209034     7.324   .0000
 AIR_HIN1|    -.00537349       .01152940     -.466   .6412
 A_TRAIN |    5.54985728       .64042443     8.666   .0000
 TRA_HIN2|    -.05656186       .01397335    -4.048   .0001
 A_BUS   |    4.13028388       .67636278     6.107   .0000
 BUS_HIN3|    -.02858418       .01544418    -1.851   .0642
```

Note the construction of the compound names includes what might seem to be a redundant number at the end. This is necessary to avoid constructing identical names for different variables.

Utility Functions

A basic four choice model which contains *cost*, *time*, *one* and *income* will have utility functions

$$
\begin{aligned}
U_{i,air} &= \beta_{cost}\, cost_{i,air} + \beta_{time}\, time_{i,air} + \alpha_{air} + \gamma_{air}\, income_i + \varepsilon_{i,air},\\
U_{i,train} &= \beta_{cost}\, cost_{i,train} + \beta_{time}\, time_{i,train} + \alpha_{train} + \gamma_{train}\, income_i + \varepsilon_{i,train},\\
U_{i,bus} &= \beta_{cost}\, cost_{i,bus} + \beta_{time}\, time_{i,bus} + \alpha_{bus} + \gamma_{bus}\, income_i + \varepsilon_{i,bus},\\
U_{i,car} &= \beta_{cost}\, cost_{i,car} + \beta_{time}\, time_{i,car} + \varepsilon_{i,bus}.
\end{aligned}
$$

The simple device you will use to construct utility functions in this fashion is

; Rhs = list of attributes that vary across choices

and

; Rh2 = list of variables that do not vary across choices

The Rh2 variables are automatically expanded into a set of *J*-1 interactions with the choice specific constants, as they are in the matrix shown above. The implication is that, generally, you do not need to have these variables in your data set. They are automatically created by your command. (Note that our clogit.dat data set in Chapter 5 actually does contain the superfluous set of four choice specific constants, *aasc*, *tasc*, *basc* and *casc*.

NOTE: If you include *one* in your Rhs list, it is automatically expanded to become a set of alternative specific constants. That is, *one* is automatically moved to the Rh2 list if it is placed in the Rhs list.

The model specification for the four utility functions shown above would be

; Rhs = cost,time ; Rh2 = one,income

Note that the distinction between Rh2 and Rhs variables is that all variables in the first category are expanded by interacting with the choice specific binary variables. (The last term is dropped.)

Generic Coefficients

The simpler, but less flexible way to specify generic coefficients in a model is to use *NLOGIT*'s standard construction, by specifying a set of Rhs variables. The specification

; Rhs = gc,ttme

produces the utility functions in the first two columns in the table. Rhs variables are assumed to vary across the choices and will receive generic coefficients.

Alternative Specific Constants and Interactions with Constants

The logit model is homogeneous of degree zero in the attributes. Any attribute which does not vary across the choices, such as age, marital status, income etc., will simply fall out of the probability. Consider an example with a constant, one attribute and one characteristic,

$$\begin{aligned}
\text{Prob}(choice\ j) &= \frac{\exp(\alpha + \beta_1 cost_{ij} + \beta_2 income_i)}{\Sigma_{j=1}^{J} \exp(\alpha + \beta_1 cost_{ij} + \beta_2 income_i)} \\
&= \frac{\exp(\alpha + \beta_2 income_i)\exp(\beta_1 cost_{ij})}{\Sigma_{j=1}^{J} \exp(\alpha + \beta_2 income_i)\exp(\beta_1 cost_{ij})} \\
&= \frac{\exp(\alpha + \beta_2 income_i)\exp(\beta_1 cost_{ij})}{\exp(\alpha + \beta_2 income_i)\Sigma_{j=1}^{J} \exp(\beta_1 cost_{ij})} \\
&= \frac{\exp(\beta_1 cost_{ij})}{\Sigma_{j=1}^{J} \exp(\beta_1 cost_{ij})}.
\end{aligned}$$

With a generic coefficient, the choice invariant characteristic falls out of the model. This includes the constant term, *one*. A model which contains such a characteristic with a generic coefficient is not estimable. This carries over to all of the more elaborate models such as the HEV, nested logit and MNP models as well. The solution to this complication is to create choice specific constant terms and, if need be, interact the invariant characteristic with the constant term. This is what appears in the last eight columns in the example above. Here, it produces a hybrid model, which can have both types of variables in the utility functions.

$$\text{Prob}(choice = j) = \frac{\exp(\beta_1 cost_{i,j} + \alpha_j + \gamma_j Income_i)}{\sum_{j=1}^{J} \exp(\beta_1 cost_{i,j} + \alpha_j + \gamma_j Income_i)}.$$

There remains an indeterminacy in the model after it is expanded in this fashion. Suppose the same constant, say θ, is added to each γ_j. The resulting model is

$$\begin{aligned}
\text{Prob}(choice = j) &= \frac{\exp(\beta_1 cost_{i,j} + \alpha_j + (\gamma_j + \theta) Income_i)}{\sum_{j=1}^{J} \exp(\beta_1 cost_{i,j} + \alpha_j + (\gamma_j + \theta) Income_i)} \\
&= \frac{\exp(\beta_1 cost_{i,j} + \alpha_j + \gamma_j Income_i + \theta Income_i)}{\sum_{j=1}^{J} \exp(\beta_1 cost_{i,j} + \alpha_j + \gamma_j Income_i + \theta Income_i)} \\
&= \frac{\exp(\theta Income_i)\exp(\beta_1 cost_{i,j} + \alpha_j + \gamma_j Income_i)}{\exp(\theta Income_i)\sum_{j=1}^{J} \exp(\beta_1 cost_{i,j} + \alpha_j + \gamma_j Income_i)} \\
&= \frac{\exp(\beta_1 cost_{i,j} + \alpha_j + \gamma_j Income_i)}{\sum_{j=1}^{J} \exp(\beta_1 cost_{i,j} + \alpha_j + \gamma_j Income_i)}.
\end{aligned}$$

So, the identical model arises for any θ. This means that the model still cannot be estimated in this form. The solution to this remaining issue is to normalize the coefficients so that one of the choice varying parameters is equal to zero. *NLOGIT* sets the last one to zero. The same result applies to the choice specific constant terms that you create with *one*. This produces the data matrix shown earlier, with the last two columns (in the dashed box) normalized to zeros.

Finally, while it is necessary for choice invariant variables to appear in the Rh2 list, it is not necessary that all variables in the Rh2 list actually be choice invariant. Indeed, one could specify the preceding model with choice specific coefficients on the *cost* variable; it would appear

$$
\begin{aligned}
U_{i,air} &= \gamma_{cost,air}\, cost_{i,air} + \beta_{time}\, time_{i,air} + \alpha_{air} + \gamma_{air}\, income_i + \varepsilon_{i,air},\\
U_{i,train} &= \gamma_{cost,train}\, cost_{i,train} + \beta_{time}\, time_{i,train} + \alpha_{train} + \gamma_{train}\, income_i + \varepsilon_{i,train},\\
U_{i,bus} &= \gamma_{cost,bus}\, cost_{i,bus} + \beta_{time}\, time_{i,bus} + \alpha_{bus} + \gamma_{bus}\, income_i + \varepsilon_{i,bus},\\
U_{i,car} &= \gamma_{cost,car}\, cost_{i,car} + \beta_{time}\, time_{i,car} + \varepsilon_{i,bus}.
\end{aligned}
$$

Note also, that there is no need to drop one of the *cost* coefficients because the variable *cost* varies by choices. You *can* estimate a model with four separate coefficients for *cost*, one in each utility function. However, it is not possible to do it by including *cost* in the Rh2 list as described above, because this form will automatically drop the last term (the one in the *car* utility function). You could obtain this form, albeit a bit clumsily, by creating the four interaction terms yourself and including them on the right hand side. We already have the alternative specific constants, so the following would work

```
CREATE      ; cost_a = gc * aasc
            ; cost_t  = gc * tasc
            ; cost_b = gc * basc
            ; cost_c  = gc * casc $
NLOGIT      ; ...  ; Rhs = time,cost_a,cost_t,cost_b,cost_c
                   ; Rh2 = one,income $
```

Having to create the interaction variables is going to be inconvenient. The alternative method of specifying the model described in the next section will be much more convenient. This method also allows you much greater flexibility in specifying utility functions.

HINT: There are many different possible configurations of alternative specific constants (ASCs) and alternative specific variables. In estimating a model, it is not possible to determine a priori if a singularity will arise as a consequence of the specification. You will have to discern this from the estimation results for the particular model.

The constant term, *one* fits the hint above. Recognizing this, *NLOGIT* assumes that if your Rhs list includes *one*, you are requesting a set of alternative specific constants. As such, when the Rhs list includes *one*, *NLOGIT* will create a full set of *J*-1 choice specific constants. (One of them must be dropped to avoid what amounts to the dummy variable trap.)

HINT: You need not have choice specific dummy variables in your data set. The Rh2 setup described here allows you to produce these variables as part of the model specification.

The remaining columns of the utility functions in the example above are produced with

; Rh2 = one,hinc

You should note, in addition, how the variables are expanded, as a set, in constructing the utility functions.

Command Builders

You can specify utility functions in this format in any of the command builders, as shown in Figure 6.4. The two windows allow you to select variables from the list at the right and assemble the Rhs list at the left or the Rh2 list in the center.

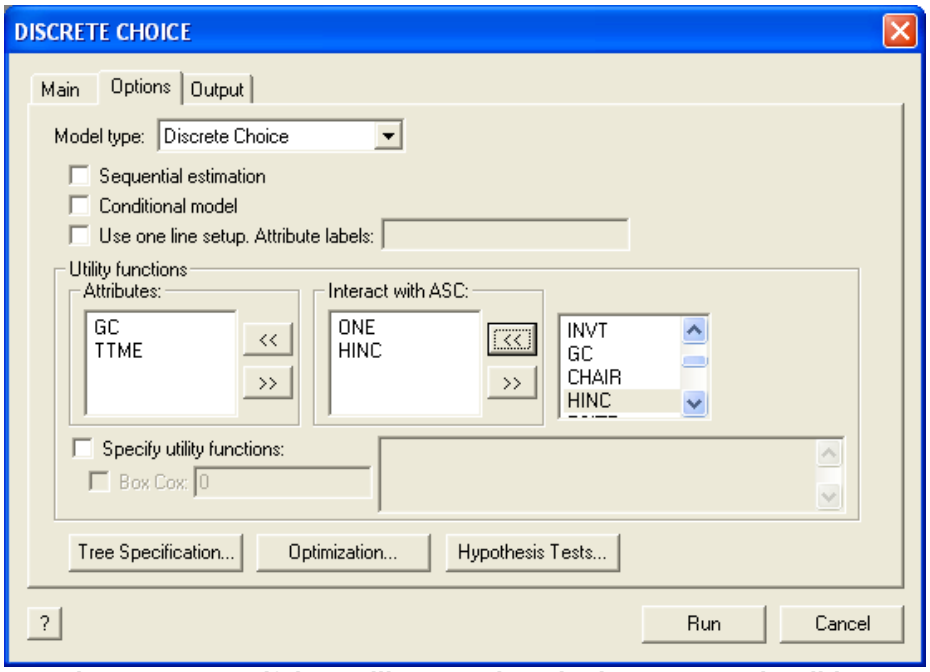


Figure 6.4 Specifying Utility Functions in the Command Builder

6.2.5 Building the Utility Functions

The model specification thus far builds the utility functions from the common Rhs and Rh2 specification. For example, in a four outcome model which contains *cost*, *time*, *one* and *income*, the data for the choice variable and the utility functions are contained in

$$\begin{array}{cc} & \begin{array}{ccccccccc} \text{choice} & \text{cost} & \text{time} & \text{constants} & & & & \text{income} & \end{array} \\ \mathbf{Z}_i = & \begin{bmatrix} y_{air} & c_a & t_a & 1 & 0 & 0 & income & 0 & 0 \\ y_{train} & c_t & t_t & 0 & 1 & 0 & 0 & income & 0 \\ y_{bus} & c_b & t_b & 0 & 0 & 1 & 0 & 0 & income \\ y_{car} & c_c & t_c & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

The utility functions are all the same;

$$U_{i,j} = \beta_1 cost_{i,j} + \beta_2 time_{i,j} + \alpha_j + \gamma_j income_i + \varepsilon_{i,j}.$$

One might want to have different attributes appear in the different utility functions, or impose other kinds of constraints on the parameters. This section will describe how to structure the utility functions individually, rather than generically with the Rhs and Rh2 lists.

The utility functions need not be the same for all choices. Different attributes may enter, and the coefficients may be constrained in different ways. The following more flexible format can be used instead of the **; Rhs = list** and **; Rh2 = list** parts of the command described above. This format also provides a way to provide starting values for parameters, so this can also replace the **; Start = list** specification. Finally, you will also be able to use this format to fix coefficients, so it will be an easy way to replace the **; Rst = list** specification.

We begin with the case of a fixed (and named) set of choices, then turn to the cases of variable numbers of choices. We replace the Rhs/Rh2 setup with explicit definitions of the utility functions for the alternatives. Utility functions are built up from the format

```
; Model:  U (choice 1) = linear equation  /
          U (choice 2) = linear equation  /
          ...
          U (choice J) = linear equation $
```

Though we have shown all *J* utility functions, for a given model specification, you could, in principle, not specify a utility function in the list. (The implied specification would be $U_{ij} = \varepsilon_{ij}$.) The **: U (list)** is mandatory. *NLOGIT* scans for the '*U*' and the parentheses. For example:

```
; Model:  U (air)  =  ba  +  bcost * gc
```

Note that the specification begins with '**; Model:**' – the colon ('**:**') is also mandatory. Parameters always come first, then variables. Constant terms need not multiply variables. Thus, *ba* in this model *could* be an '*air specific constant*.' (It depends on whether *ba* appears elsewhere in the model.) Notice that the utility function defines both the variables and the parameters. Usually, you would give an equation for each choice in the model. For example:

```
NLOGIT    ; Lhs = mode
          ; Choices = air,train,bus,car
          ; Model:  U(air)    = ba +  bcost * gc + btime * ttme /
                   U(car)    = bc +  bcost * gc   /
                   U(bus)    = bb +  bcost * gc   /
                   U(train) =         bcost * gc + btime * ttme $
```

Utility functions are separated by slashes. Note also that the alternative specific constants stand alone without multiplying a variable. Your utility definitions now provide the names for the parameters. The estimates produced by this model command are as follows:

```
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 BA      |    1.55491032       .37580063     4.138   .0000
 BCOST   |    -.02020918       .00434927    -4.647   .0000
 BTIME   |    -.08680295       .01122237    -7.735   .0000
 BC      |   -3.65316491       .46378035    -7.877   .0000
 BB      |   -3.91982604       .45611114    -8.594   .0000
```

One point that you might find useful to note. The order of the parameters in this list is determined by moving through the model definition from beginning to end. Each time a new parameter name is encountered, it is added to the list. Looking at the model command above, you can now see how the order in the displayed output arose.

The last example in the preceding subsection, which has four separate coefficients on a cost variable, *gc*, could be specified using

```
NLOGIT      ; Lhs = mode ; Choices = air,train,bus,car
            ; Model:  U(air)   = bc*invc+bt*invt+aa+cha*hinc +cga *gc /
                      U(train) = bc*invc+bt*invt+at +cht *hinc +cgt *gc /
                      U(bus)   = bc*invc+bt*invt+ab+chb*hinc+cgb *gc /
                      U(car)   = bc*invc+bt*invt                +cgc *gc $
```

The estimates are

```
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 BC      |    -.04386562       .01712959    -2.561   .0104
 BT      |    -.00815115       .00241976    -3.369   .0008
 AA      |   -1.37473591       .83837138    -1.640   .1011
 CHA     |     .00703267       .01078793      .652   .5145
 CGA     |     .03762100       .01676624     2.244   .0248
 AT      |    2.53156832       .60800716     4.164   .0000
 CHT     |    -.05096641       .01214303    -4.197   .0000
 CGT     |     .03348741       .01506250     2.223   .0262
 AB      |    1.17857565       .73948909     1.594   .1110
 CHB     |    -.03339204       .01299642    -2.569   .0102
 CGB     |     .03455919       .01516387     2.279   .0227
 CGC     |     .03808057       .01523791     2.499   .0125
```

Shorthand Notations for Sets of Utility Functions

There are several shorthands which will allow you to make the model specification much more compact. If the utility functions for several alternatives are the same, you can group them in one definition. Thus,

```
; Model: U(air)  = b0 + bcost * gc   /
         U(car)  = b0 + bcost * gc   $
```

could be specified with

```
; Model: U(air, car) = b0 + bcost * gc $
```

For the model we have been considering, i.e.,

; Choices = air,train,bus,car

all of the following are the same

```
; Model: U(air)   = b1 * ttme + bcost * gc  /
       U(train) = b1 * ttme + bcost * gc  /
       U(bus)   = b1 * ttme + bcost * gc  /
       U(car)   = b1 * ttme + bcost * gc  $
```

and **; Model: U(air,train,bus,car) = b1 * ttme + bcost * gc $**

and **; Model: U(*) = b1 * ttme + bcost * gc $**

and **; Rhs = ttme, gc**

The last will use the variable names instead of the supplied parameter names for the two parameters, but the models will be the same.

Alternative Specific Constants and Interactions

You can also specify alternative specific constants in this format, by using a special notation. When you have a **U(a1, a2, ..., aJ)** for *J* alternatives, then you may specify, instead of a single parameter, a list of parameters enclosed in pointed brackets, to signify interaction with choice specific constants. Thus, **<b1,b2,...,bL>** indicates *L* interactions with choice specific dummy variables. *L* may be any number up to the number of alternatives. Use a zero in any location in which the variable does not appear in the corresponding equation. For example,

```
; Model: U(air)   = ba +  bcost * gc  /
       U(car)   = bc +  bcost * gc  /
       U(bus)   =       bcost * gc  /
       U(train) = bt +  bcost * gc  $
```

could be specified as

; Model: U(air,car,bus,train) = <ba,bc,0,bt> + bcost * gc $

NOTE: Within a < ... > construction, the correspondence between positions in the list is with the **U (... list ...)** list, *not* with the original **; Choices** list.

Note the considerable savings in notation. The same device may also be used in interactions with attributes. For example:

```
; Model: U(air)   = ba    +      bcprv * gc  /
       U(car)   = bc + bcprv  * gc  /
       U(bus)   =      bcpub * gc  /
       U(train) = bt + bcpub * gc  $
```

There are two cost coefficients, but the variable *gc* is common. This entire model can be collapsed into the single specification

```
; Model: U(air,car,bus,train) = <ba,bc,0,bt> +
         <bcprv,bcprv,bcpub,bcpub> * gc $
```

Parameters inside the brackets need not all be different if you wish to impose equality constraints.

Equality Constraints

There is no requirement that parameters be unique across any specification. Equality constraints may be imposed anywhere in the model, just by using the same parameter name. For example, nothing precludes

```
; Model: U(air,car,bus,train) = <ba,bc,0,bt> +
         <ba,bc,bcpub,bcpub> * gc $
```

This forces two of the slope coefficients to equal the alternative specific constants. Expanded, this specification would be equivalent to

```
; Model: U(air)   = ba + ba      * gc  /
         U(car)   = bc + bc      * gc  /
         U(bus)   =      bcpub * gc  /
         U(train) = bt  + bcpub * gc  $
```

Logs and the Box-Cox Transformation

Variables may be specified in logarithms. This will be useful when you are using aggregate data and you wish to include, e.g., market size in a choice. To indicate that you wish to use logs, use **Log**(*variable name*) instead of just *variable name* in the utility definition. (The syntax **; Rhs = ... Log(x)** as described above is not available. This option may only be used when you are explicitly defining the utility functions.) Thus, the model above might have been

```
NLOGIT   ; Lhs = mode
         ; Choices = air,train,bus,car
         ; Model: U(air)   = ba + bcost * Log(gc)  /
                 U(car)   = bc + bcost * Log(gc)  /
                 U(bus)   = bb + bcost * Log(gc)  /
                 U(train) =      bcost * Log(gc)  $
```

When a variable appears in more than one utility function, you should take logs each time it appears. Although this is not mandatory, if you do not, your model will contain a mix of levels and logs, which is probably not what you want. Also, it will be necessary for you to be aware in your results when you have used this transformation. The model results will not contain any indication that logs have appeared in the equation. The preceding, for example, produces the following estimation results:

```
+---------+--------------+----------------+--------+---------+
|Variable | Coefficient  | Standard Error |b/St.Er.|P[|Z|>z] |
+---------+--------------+----------------+--------+---------+
 BA          -.5929844379       .21339917   -2.779   .0055
 BCOST       -2.630222154       .45170772   -5.823   .0000
 BC          -.9545367356       .24330526   -3.923   .0001
 BB          -.9785661344       .22951885   -4.264   .0000
```

You may also use the Box-Cox transformation to transform variables. Indicate this with Bcx(x) where x is the variable (which must be positive). The transformation is

$$\text{Bcx}(x) = (x^{\lambda} - 1) / \lambda,$$

which is Log(x) if λ equals 0 and is x-1 (not x) if λ equals 1. The Bcx(**.**) function may appear any number of times in the model specification. In general, if a variable is transformed with this function, it should be transformed every time it appears in the model. Not doing so is analogous to including both levels and logs of a variable, which while not invalid, is usually avoided. The default value of the transformation parameter, λ, is 1.0. The same value is used in all transformations. You may specify a different value by including the specification

; Lambda = value

in your **NLOGIT** command. *Lambda* is treated as a fixed value during estimation, not an estimated parameter. Thus, no standard error is computed for *lambda* (since you provide the fixed value) and the standard errors for the other estimates are not adjusted for the presence of *lambda*. I.e., by this construction, the Box-Cox transformation is treated like the log function – just a transformation. In this case, the model results will contain an indication that the transformation has appeared in the utility functions. For example, the preceding, with λ = 0.5, produces

```
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Dependent variable               Choice     |
| Number of observations              210     |
| Log likelihood function       -267.4253     |
| Log-L for Choice   model =   -267.42533     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -283.7588  .05756  .05154 |
| Chi-squared[ 1]          =     32.66687     |
| Prob [ chi squared > value ] =   .00000     |
| Response data are given as ind. choice.     |
| Box-Cox model. LAMBDA used is      .50000   |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
+---------+--------------+----------------+--------+---------+
|Variable | Coefficient  | Standard Error |b/St.Er.|P[|Z|>z] |
+---------+--------------+----------------+--------+---------+
 BA          -.6425631472       .21842858   -2.942   .0033
 BCOST       -.2433427492   .44558999E-01   -5.461   .0000
 BC          -.8456971936       .23246443   -3.638   .0003
 BB          -.9996728009       .22980046   -4.350   .0000
```

Do note, however, that the results can only indicate that a Box-Cox transformation using λ = 0.5 has appeared in the model. It is not possible to report where it appears.

Command Builders

The command builders provide space for you to build the utility functions in this fashion. See Figure 6.5. Since this is done by typing out the functions in the windows – there is no menu construction that would allow this – these will not save much effort.

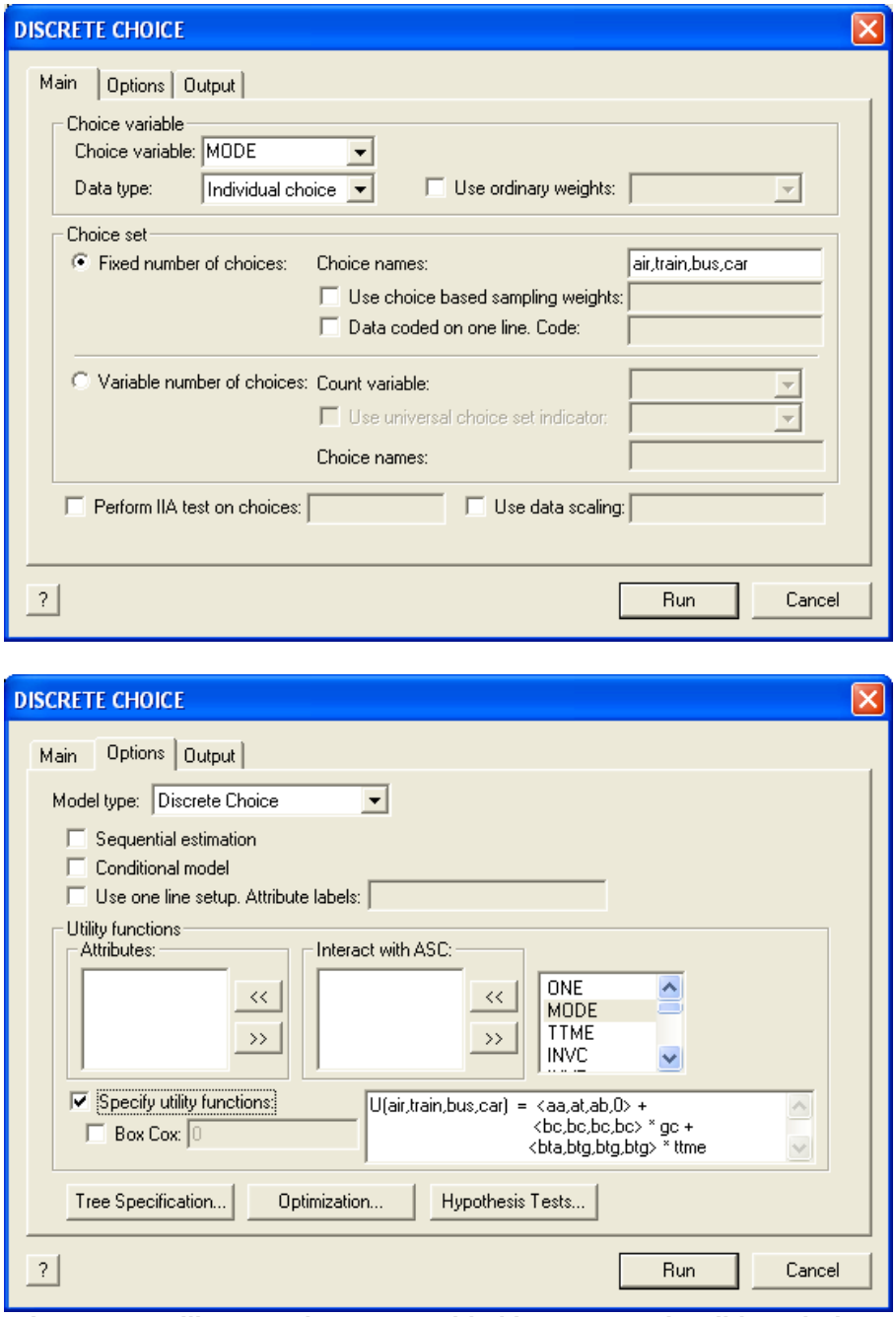


Figure 6.5 Utility Functions Assembled in Command Builder Window

Note that in the window, you must provide the entire specification for the utility functions, including the listing of which alternatives the definitions are to apply to. The model shown in the window in Figure 6.5 produces these results.

```
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Maximum Likelihood Estimates                |
| Dependent variable               Choice     |
| Weighting variable                 None     |
| Number of observations              210     |
| Iterations completed                  6     |
| Log likelihood function       -199.6825     |
| Number of parameters                  6     |
| Info. Criterion: AIC =          1.95888     |
|   Finite Sample: AIC =          1.96085     |
| Info. Criterion: BIC =          2.05451     |
| Info. Criterion:HQIC =          1.99754     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -283.7588  .29630  .28953 |
| Chi-squared[ 3]          =    168.15262     |
| Prob [ chi squared > value ] =   .00000     |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+

+---------------------------------------------+
| Notes No coefficients=> P(i,j)=1/J(i).      |
|       Constants only => P(i,j) uses ASCs    |
|         only. N(j)/N if fixed choice set.   |
|         N(j) = total sample frequency for j |
|         N    = total sample frequency.      |
|       These 2 models are simple MNL models. |
|       R-sqrd = 1 - LogL(model)/logL(other)  |
|       RsqAdj=1-[nJ/(nJ-nparm)]*(1-R-sqrd)   |
|         nJ   = sum over i, choice set sizes |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 AA      |    6.41353627      1.10452186     5.807   .0000
 AT      |    3.69564345       .52116476     7.091   .0000
 AB      |    2.96221779       .54485066     5.437   .0000
 BC      |    -.01702110       .00471351    -3.611   .0003
 BTA     |    -.10758045       .01791733    -6.004   .0000
 BTG     |    -.08939996       .01419339    -6.299   .0000
```

6.3 Standard Model Results

Estimation results for the model commands consist of the initial display of diagnostic followed by notes about the model, then the estimated coefficients. The preceding command, without the tree structure or the initial echo of the model specification,

```
NLOGIT      ; Lhs = mode ; Choices = air,train,bus,car
            ; Rhs = invc,invt,gc
            ; Rh2 = one,hinc $
```

produces the following results:

```
Normal exit from iterations. Exit status=0.
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Maximum Likelihood Estimates                |
| Dependent variable               Choice     |
| Weighting variable                 None     |
| Number of observations              210     |
| Iterations completed                  5     |
| Log likelihood function       -246.1098     |
| Number of parameters                  9     |
| Info. Criterion: AIC =          2.42962     |
|   Finite Sample: AIC =          2.43390     |
| Info. Criterion: BIC =          2.57306     |
| Info. Criterion:HQIC =          2.48761     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -283.7588  .13268  .12011 |
| Chi-squared[ 6]          =     75.29796     |
| Prob [ chi squared > value ] =   .00000     |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
+---------------------------------------------+
| Notes No coefficients=> P(i,j)=1/J(i).      |
|       Constants only => P(i,j) uses ASCs    |
|         only. N(j)/N if fixed choice set.   |
|         N(j) = total sample frequency for j |
|         N    = total sample frequency.      |
|       These 2 models are simple MNL models. |
|       R-sqrd = 1 - LogL(model)/logL(other)  |
|       RsqAdj=1-[nJ/(nJ-nparm)]*(1-R-sqrd)   |
|         nJ   = sum over i, choice set sizes |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 INVC    |    -.04612501       .01664864    -2.770   .0056
 INVT    |    -.00838543       .00214019    -3.918   .0001
 GC      |     .03633292       .01477727     2.459   .0139
 A_AIR   |   -1.31602481       .72323155    -1.820   .0688
 AIR_HIN1|     .00648950       .01079433      .601   .5477
 A_TRAIN |    2.10710471       .43179879     4.880   .0000
 TRA_HIN2|    -.05058498       .01206873    -4.191   .0000
 A_BUS   |     .86502331       .50318615     1.719   .0856
 BUS_HIN3|    -.03316081       .01299094    -2.553   .0107
```

NOTE: (This is one of our frequently asked questions.) The '*R*-squareds' shown in the output are R^2s in name only. They do not measure the fit of the model to the data. It has become common for researchers to report these with results as a measure of the improvement that the model gives over one that contains only a constant. But, users are cautioned not to interpret these measures as suggesting how well the model predicts the outcome variable. It is essentially unrelated to this.

To underscore the point, we will examine in detail the computations in the diagnostic measures shown in the box that precedes the coefficient estimates. Consider the example below, which was produced by fitting a model with five coefficients subject to two restrictions, or three free coefficients - *npfree* = 3. The effect is achieved by specifying

; Choices = air,(train),(bus),car

```
+-----------------------------------------------------+
|WARNING:   Bad observations were found in the sample. |
|Found  93 bad observations among     210 individuals. |
|You can use ;CheckData to get a list of these points. |
+-----------------------------------------------------+
Sample proportions are marginal, not conditional.
Choices marked with * are excluded for the IIA test.
+----------------+------+---
|Choice   (prop.)|Weight|IIA
+----------------+------+---
|AIR       .49573| 1.000|
|TRAIN     .00000| 1.000|*
|BUS       .00000| 1.000|*
|CAR       .50427| 1.000|
+----------------+------+---
+-----------------------------------------------------------------+
| Model Specification:  Table entry is the attribute that         |
| multiplies the indicated parameter.                             |
+--------+------+-----------------------------------------------+
| Choice |******| Parameter                                     |
|        |Row  1| GC       TTME     A_AIR    A_TRAIN  A_BUS      |
+--------+------+-----------------------------------------------+
|AIR     |     1| GC       TTME     Constant none     none       |
|TRAIN   |     1| GC       TTME     none     Constant none       |
|BUS     |     1| GC       TTME     none     none     Constant   |
|CAR     |     1| GC       TTME     none     none     none       |
+-----------------------------------------------------------------+
Normal exit from iterations. Exit status=0.
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Maximum Likelihood Estimates                |
| Dependent variable               Choice     |
| Weighting variable                 None     |
| Number of observations              117     |
| Iterations completed                  6     |
| Log likelihood function       -62.58418     |
| Number of parameters                  3     |
| Info. Criterion: AIC =          1.12110     |
|   Finite Sample: AIC =          1.12291     |
| Info. Criterion: BIC =          1.19192     |
| Info. Criterion:HQIC =          1.14985     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
```

```
| Constants only     -81.0939  .22825  .20794 |
| Chi-squared[ 2]          =     37.01953     |
| Prob [ chi squared > value ] =   .00000     |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped  93 bad obs. |
+---------------------------------------------+
| Restricted choice set. Excluded choices are |
| TRAIN    BUS                                |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 GC      |     .01320101       .00694790     1.900   .0574
 TTME    |    -.07141256       .01604643    -4.450   .0000
 A_AIR   |    3.96116758       .98004184     4.042   .0001
 A_TRAIN |       .000000    ......(Fixed Parameter)........
 A_BUS   |       .000000    ......(Fixed Parameter)........
```

There are 210 individuals in the data set, but this model was fit to a restricted choice set which reduced the data set to n = 210 - 93 = 117 useable observations. The original choice set had J_i = 4 choices, but two were excluded, leaving J_i = 2 in the sample. The log likelihood is -62.58418. The 'constants only' log likelihood is obtained by setting each choice probability to the sample share for each outcome in the choice set. For this application, those are 0.49573 for air and 0.50427 for car. (This computation cannot be done if the choice set varies by person or if weights or frequencies are used.) Thus, the log likelihood for the restricted model is

$$\text{Log } L_0 \;=\; 117\,(\,0.49573 \times \log 0.49573 + 0.50427 \times \log 0.50427\,) \;=\; -81.09395.$$

The 'R^2' is 1 - (-62.54818/-81.0939) = 0.22869 (including some rounding error). The adjustment factor is

$$K \;=\; (\Sigma_i\, J_i - n)\,/\,[(\Sigma_i\, J_i - n) - npfree] \;=\; (234 - 117)/(234 - 117 - 3) \;=\; 1.02632.$$

and the '*Adjusted* R^2' is 1 - K(log L /LogL_0);

$$\textit{Adjusted } R^2 \;=\; 1 \;-\; 1.02632\,(-62.54818/-81.0939) \;=\; 0.20794.$$

6.3.1 Retained Results

Results kept by this estimator are:

Matrices:	*b* and *varb*	=	coefficient vector and asymptotic covariance matrix
Scalars:	*logl*	=	log likelihood function
	nreg	=	N, the number of observational units
	kreg	=	the number of Rhs variables
Last Model:	*b_variable*	=	the labels kept for the **WALD** command

In the *Last Model*, groups of coefficients for variables that are interacted with constants get labels *choice_variable*, as in *trai_gco*. (Note that the names are truncated – up to four characters for the choice and three for the attribute.) The alternative specific constants are *a_choice*, with names truncated to no more than six characters. For example, the sum of the three estimated choice specific constants could be analyzed as follows:

WALD ; Fn1 = a_air + a_train + a_bus $

```
+-----------------------------------------------+
| WALD procedure. Estimates and standard errors |
| for nonlinear functions and joint test of     |
| nonlinear restrictions.                       |
| Wald Statistic             =     57.91928     |
| Prob. from Chi-squared[ 1] =       .00000     |
+-----------------------------------------------+
+---------+--------------+----------------+--------+---------+
|Variable | Coefficient  | Standard Error |b/St.Er.|P[|Z|>z] |
+---------+--------------+----------------+--------+---------+
 Fncn(1)      13.32858178       1.7513477    7.610   .0000
```

6.3.2 Robust Standard Errors

The ‘cluster’ estimator described elsewhere in this document is available in *NLOGIT*. However, this routine does not support hierarchical samples. There may be only one level of clustering. Also, the cluster specification is defined with respect to the *NLOGIT* groups of data, not the data set. *NLOGIT* sorts out how many clusters there are and how they are delineated. But, since the row count of the data set is used in constructing the estimator, you must treat a group of NALT observations as one. For example, our sample data used in this section contain 210 groups of four rows of data. Each group of four is an observation. Suppose that these data were grouped in clusters of three choice situations. The estimation command with the cluster estimator would appear

NLOGIT ; ... (the model) ; Cluster = 12 $

The relevant part of the output would appear as follows:

```
+----------------------------------------------------------------------+
| Covariance matrix for the model is adjusted for data clustering.     |
| Sample of    210 observations contained     70 clusters defined by   |
|      3 observations (fixed number) in each cluster.                  |
+----------------------------------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 GC      |    -.01578375       .00543575    -2.904   .0037
 TTME    |    -.09709052       .01366784    -7.104   .0000
 A_AIR   |    5.77635888       .74564933     7.747   .0000
 A_TRAIN |    3.92300124       .47890812     8.192   .0000
 A_BUS   |    3.21073471       .48991386     6.554   .0000
```

6.3.3 Descriptive Statistics for Alternatives

You may request a set of descriptive statistics for your model by adding

; Describe

to the model command. For each alternative, a table is given which lists the nonzero terms in the utility function and the means and standard deviations for the variables that appear in the utility function. Values are given for all observations and for the individuals that chose that alternative. For the example shown above, the following tables would be produced:

```
NLOGIT      ; Lhs = mode ; Choices = air,train,bus,car
            ; Rhs = invc,invt,gc ; Rh2 = one,hinc
            ; Show Model
            ; Describe $
```

```
+-----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative AIR                :
|     Utility Function          |                     |     58.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose AIR      |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| -------------------  -------- | -------------------+------------------- |
| INVC         -.0461  INVC     |   85.252     27.409|   97.569    31.733 |
| INVT         -.0084  INVT     |  133.710     48.521|  124.828    50.288 |
| GC            .0363  GC       |  102.648     30.575|  113.552    33.198 |
| A_AIR       -1.3160  ONE      |    1.000       .000|    1.000      .000 |
| AIR_HIN1      .0065  HINC     |   34.548     19.711|   41.724    19.115 |
+-----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative TRAIN              :
|     Utility Function          |                     |     63.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose TRAIN    |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| -------------------  -------- | -------------------+------------------- |
| INVC         -.0461  INVC     |   51.338     27.032|   37.460    20.676 |
| INVT         -.0084  INVT     |  608.286    251.797|  532.667   249.360 |
| GC            .0363  GC       |  130.200     58.235|  106.619    49.601 |
| A_TRAIN      2.1071  ONE      |    1.000       .000|    1.000      .000 |
| TRA_HIN2     -.0506  HINC     |   34.548     19.711|   23.063    17.287 |
+-----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative BUS                :
|     Utility Function          |                     |     30.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose BUS      |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| -------------------  -------- | -------------------+------------------- |
| INVC         -.0461  INVC     |   33.457     12.591|   33.733    11.023 |
| INVT         -.0084  INVT     |  629.462    235.408|  618.833   273.610 |
| GC            .0363  GC       |  115.257     44.934|  108.133    43.244 |
| A_BUS         .8650  ONE      |    1.000       .000|    1.000      .000 |
| BUS_HIN3     -.0332  HINC     |   34.548     19.711|   29.700    16.851 |
+-----------------------------------------------------------------------+
```

```
+----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative CAR                :
|     Utility Function          |                    |     59.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose CAR      |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| ------------------  -------- | -------------------+------------------- |
| INVC         -.0461  INVC     |   20.995     14.678|   15.644     9.629 |
| INVT         -.0084  INVT     |  573.205    274.855|  527.373   301.131 |
| GC            .0363  GC       |   95.414     46.827|   89.085    49.833 |
+----------------------------------------------------------------------+
```

You may also request a cross tabulation of the model predictions against the actual choices. (The predictions are obtained as the integer part of $\Sigma_t \hat{P}_{jt} y_{jt}$.) Add

; Crosstab

to your model command. For the same model, this would produce

```
+-------------------------------------------------------+
| Cross tabulation of actual vs. predicted choices.     |
| Row indicator is actual, column is predicted.         |
| Predicted total is F(k,j,i)=Sum(i=1,...,N) P(k,j,i).  |
| Column totals may be subject to rounding error.       |
+-------------------------------------------------------+
         AIR           TRAIN         BUS           CAR           Total
        +-----------------------------------------------------------------------
AIR     |   19.00000     13.00000      8.00000     18.00000     58.00000
TRAIN   |   12.00000     30.00000      9.00000     12.00000     63.00000
BUS     |   10.00000      8.00000      6.00000      6.00000     30.00000
CAR     |   17.00000     12.00000      7.00000     23.00000     59.00000
Total   |   58.00000     63.00000     30.00000     59.00000    210.00000
```

6.4 Marginal Effects and Elasticities

In the discrete choice model, the effect of a change in attribute '*k*' of alternative '*j*' on the probability that individual *i* would choose alternative '*m*' (where *m* may or may not equal *j*) is

$$\delta_{im}(k|j) \;=\; \partial \text{Prob}[y_i = m]/\partial x_i(k|j) \;=\; [\mathbf{1}(j = m) - P_{ij}]P_{im}\beta_k..$$

You can request a listing of the effects of a specific attribute on a specified set of outcomes with

; Effects: attribute [list of outcomes]

The outcomes listing defines the variables '*j*' in the definition above. The attribute is the '*k*th.' A calculated marginal effect is then listed for all alternatives (i.e., all '*m*') in the model. You can request additional tables by separating additional specifications with slashes. For example:

; Effects: gc [car, train] / ttme [bus,train]

HINT: It may generate quite a lot of output if your model is large, but you can request an analysis of 'all' alternatives by using the wildcard, **attribute [*]**.

The tables below are produced by

```
NLOGIT      ; Lhs  = mode ; Choices = air,train,bus,car
            ; Rhs  = invc,invt,gc
            ; Rh2 = one,hinc
            ; Effects: gc[*] $
```

```
+---------------------------------------------------+
| Derivative (times 100) averaged over observations.|
| Attribute is GC       in choice AIR               |
| Effects on probabilities of all choices in model: |
| * = Direct Derivative effect of the attribute.    |
|                                  Mean    St.Dev   |
| *     Choice=AIR                .6042     .2397   |
|       Choice=TRAIN             -.2007     .1132   |
|       Choice=BUS               -.1237     .0798   |
|       Choice=CAR               -.2798     .2044   |
+---------------------------------------------------+
| Derivative (times 100) averaged over observations.|
| Attribute is GC       in choice TRAIN             |
| Effects on probabilities of all choices in model: |
| * = Direct Derivative effect of the attribute.    |
|                                  Mean    St.Dev   |
|       Choice=AIR               -.2007     .1132   |
| *     Choice=TRAIN              .6180     .2612   |
|       Choice=BUS               -.1754     .1377   |
|       Choice=CAR               -.2420     .1305   |
+---------------------------------------------------+
| Derivative (times 100) averaged over observations.|
| Attribute is GC       in choice BUS               |
| Effects on probabilities of all choices in model: |
| * = Direct Derivative effect of the attribute.    |
|                                  Mean    St.Dev   |
|       Choice=AIR               -.1237     .0798   |
|       Choice=TRAIN             -.1754     .1377   |
| *     Choice=BUS                .4332     .1431   |
|       Choice=CAR               -.1342     .0648   |
+---------------------------------------------------+
| Derivative (times 100) averaged over observations.|
| Attribute is GC       in choice CAR               |
| Effects on probabilities of all choices in model: |
| * = Direct Derivative effect of the attribute.    |
|                                  Mean    St.Dev   |
|       Choice=AIR               -.2798     .2044   |
|       Choice=TRAIN             -.2420     .1305   |
|       Choice=BUS               -.1342     .0648   |
| *     Choice=CAR                .6559     .2159   |
+---------------------------------------------------+
```

These effects are always extremely small. They are multiplied by 100 in the output to make sure that some significant digits are shown in the tables. The effects are computed by averaging the individual specific results, so the report contains the average partial effects. Since the mean is computed over a sample of observations, we also report the standard deviation of the estimates.

NOTE: The standard deviations are not the asymptotic standard errors for the estimators of the marginal effects. In principle, that could be computed using the delta method. However, the estimates computed by *NLOGIT* are *average partial effects*. They are computed for each individual in the sample, then averaged. Computing an appropriate standard error for that statistic is difficult to impossible owing to its extreme nonlinearity and due to the fact that all observations in the average are correlated – they use the same estimated parameter vector. Nonetheless, it may be tempting to use the standard deviations for tests of hypotheses that the marginal effects are zero. We advise against this. There is no meaning that could be attached to an elasticity or marginal effect being zero – these are complicated functions of all parameters in the model. The hypothesis that a variable is not influential in the determination of the choices should be tested at the coefficient level.

As noted in the tables, the marginal effects are computed by averaging the individual sample observations. An alternative way to compute these is to use the sample means of the data, and compute the effects for this one hypothetical observation. Request this with

; Means

For the first table above, the results would be as follows:

```
+---------------------------------------------------+
| Derivative (times 100) Computed at sample means.  |
| Attribute is GC       in choice AIR               |
| Effects on probabilities of all choices in model: |
| * = Direct Derivative effect of the attribute.    |
|                                 Mean    St.Dev    |
| *     Choice=AIR                .7263     .0000   |
|       Choice=TRAIN             -.3010     .0000   |
|       Choice=BUS               -.1434     .0000   |
|       Choice=CAR               -.2819     .0000   |
+---------------------------------------------------+
```

Note that the changes are substantial. The literature is divided on this computation. Current practice seems to favor the first approach.

Rather than see the partial effects, you may want to see elasticities,

$$\eta_{im}(k|j) \;=\; \partial\text{logProb}[y_i = m]/\partial\text{log}x_i(k|j) \;=\; x_i(k|j)/P_{im} \times \delta_{im}(k|j)$$
$$= \;[\mathbf{1}(j = m) - P_{ij}]\; x_i(k|j)\beta_k.$$

Notice that this is not a function of P_{im}. The implication is that all the cross elasticities are identical. This will be obvious in the results below. This aspect of the model is specific to the basic multinomial logit model. As will emerge in the chapters to follow, the IIA property which produces this result is absent from every other model in *NLOGIT*.

You may request elasticities instead of partial effects simply by changing the square brackets above to parentheses, as in

; Effects: attribute (list of outcomes)

The first set of results above would become as shown in the following table:

```
+---------------------------------------------------+
| Elasticity             Averaged over observations.|
| Attribute is GC       in choice AIR               |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.    |
|                                  Mean    St.Dev   |
| *     Choice=AIR               2.6002     .8212   |
|       Choice=TRAIN            -1.1293     .9295   |
|       Choice=BUS              -1.1293     .9295   |
|       Choice=CAR              -1.1293     .9295   |
+---------------------------------------------------+
| Elasticity             Averaged over observations.|
| Attribute is GC       in choice TRAIN             |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.    |
|                                  Mean    St.Dev   |
|       Choice=AIR              -1.2046     .8221   |
| *     Choice=TRAIN             3.5259    2.1605   |
|       Choice=BUS              -1.2046     .8221   |
|       Choice=CAR              -1.2046     .8221   |
+---------------------------------------------------+
| Elasticity             Averaged over observations.|
| Attribute is GC       in choice BUS               |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.    |
|                                  Mean    St.Dev   |
|       Choice=AIR               -.5695     .2859   |
|       Choice=TRAIN             -.5695     .2859   |
| *     Choice=BUS               3.6181    1.4924   |
|       Choice=CAR               -.5695     .2859   |
+---------------------------------------------------+
| Elasticity             Averaged over observations.|
| Attribute is GC       in choice CAR               |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.    |
|                                  Mean    St.Dev   |
|       Choice=AIR               -.8688     .5119   |
|       Choice=TRAIN             -.8688     .5119   |
|       Choice=BUS               -.8688     .5119   |
| *     Choice=CAR               2.5979    1.5604   |
+---------------------------------------------------+
```

The force of the independence from irrelevant alternatives (IIA) assumption of the multinomial logit model can be seen in the identical elasticities in the tables above. The table also shows two aspects of the model. First, the meaning of the raw coefficients in a multinomial logit model, all of sign, magnitude and significance, are ambiguous. It is always necessary to do some kind of post estimation such as this to determine the implications of the estimates. Second, in light of this, we can see that the particular model we estimated seems to be misspecified. The estimates imply that as the generalized cost of each mode rises, it becomes more attractive. The *gc* coefficient has the ‘wrong’ sign.

6.5 Predicted Probabilities and Inclusive Values

There are some models that make use of the predicted probabilities from the discrete choice model.

6.5.1 In Sample Predicted Probabilities and Inclusive Values

You can compute a column of predicted probabilities for any estimated choice model. Each 'observation' consists of J_i rows of data, where the number of choices may be fixed or variable. Use the command

NLOGIT ; Lhs = ... ; ...
; Prob = name $

The variable *name* will contain the predicted probabilities. The probabilities will sum to 1.0 for each observation, that is, *down* each set of J_i choices. The **; Prob** option will put the probabilities in the right places in your data set regardless of the setting of the current sample. For example, if you happen to be estimating a model after having rejected some observations, the predictions will be placed with the outcomes for the observations actually used. Unused rows of the data matrix are left undefined.

If your model has 14 or fewer choices, you can also include

; List

in your command to request a listing of the predicted probabilities. These will be listed a full observation at a time, rowwise, with an indicator of the choice that was made by that individual. For example, the first 10 observations (individuals) in the sample for the model above are

```
PREDICTED PROBABILITIES (* marks actual, + marks prediction.)
Indiv    AIR       TRAIN     BUS       CAR
    1   .1481     .2376     .1101     .5042*+
    2   .1182     .3694 +   .1687     .3437*
    3   .5783 +   .0702     .0663     .2853*
    4   .2367     .0725     .0659     .6250*+
    5   .2203     .3176 +   .1884     .2736*
    6   .1048     .4958*+   .1589     .2405
    7   .6500*+   .0548     .0565     .2387
    8   .3241     .3868 +   .1472     .1419*
    9   .1824     .2199     .1112     .4866*+
   10   .2863     .0575     .0491     .6071*+
```

The '+' and '*' indicate the actual and predicted choices. Where these mark the same probability, the model has predicted the outcome correctly.

The *inclusive value*, or *log sum*, for the discrete choice model is

$$IV_i = \log \Sigma_j \exp(\boldsymbol{\beta}'\mathbf{x}_{i,j}).$$

Inclusive values are used for a number of purposes, including computing consumer surplus measures. You can keep the inclusive values for your model and data with the specification

; Ivb = name

The specification, Ivb stands for 'inclusive value for branch.' Inclusive values are stored the same way that predicted probabilities are stored. Since each observation has only one inclusive value, the same value will be stored for all rows (choices) for the observation (person). Figure 6.6 illustrates

Data Editor

17/900 Vars; 11111 Rows: 840 Obs Cell: 0

	LOGL_OBS	INCVLU	PMNL
1 »	0	-0.195808	0.148125
2 »	0	-0.195808	0.237597
3 »	0	-0.195808	0.110103
4 »	-0.684832	-0.195808	0.504175
5 »	0	0.239082	0.118216
6 »	0	0.239082	0.369399
7 »	0	0.239082	0.168723
8 »	-1.0681	0.239082	0.343662
9 »	0	-2.17441	0.578309
10 »	0	-2.17441	0.0701569
11 »	0	-2.17441	0.0662732
12 »	-1.25435	-2.17441	0.285261
13 »	0	-0.107306	0.236656
14 »	0	-0.107306	0.0725174
15 »	0	-0.107306	0.0658503
16 »	-0.470042	-0.107306	0.624976
17 »	0	-0.507153	0.220348
18 »	0	-0.507153	0.317646
19 »	0	-0.507153	0.188422

Figure 6.6 Saved Inclusive Values and Probabilities

6.5.2 Computing Out of Sample Model Probabilities

You can use an estimated model to compute (list and/or save) all probabilities, utilities, elasticities, and all descriptive statistics and crosstabulations for any specified set of observations, whether they were used in estimating the model or not. For example, this feature will allow you to compute predicted probabilities for a 'control' sample, to assess how well the model predicts outcomes for observations outside the estimation sample. For this feature, use the following steps:

Step 1. Set up the full model for estimation, and estimate the model parameters.

Step 2. Reset the sample to specify the observations for which you wish to simulate the model.

Step 3. Use the *identical* **NLOGIT** command, but add the specification **; Prlist** to the command.

The sample that you specify at Step 2 may contain as many observations as you wish; it may be just one individual or it may be an altogether different set of data.

NOTE: The observations in the new sample must be consistent with the specification of the model. The usual data checking is done to ensure this.

WARNING: You must not change the specification of the model between Steps 1 and 3. The coefficient vector produced by Step 1 is used for the simulation at Step 3. But it is not possible to check whether the coefficient vector used at Step 3 is actually the correct one for the model command used at Step 3. It will be if your model commands at Steps 1 and 3 are identical.

The following sequence fits the model in the preceding examples using the first 200 observations (800 data rows), then simulates the probabilities for the remaining 10 observations in the full sample:

```
SAMPLE    ; 1 - 800 $
NLOGIT    ; Lhs  = mode
          ; Choices = air,train,bus,car
          ; Rhs  = invc,invt,gc,ttme
          ; Rh2 = one $
SAMPLE    ; 801 - 840 $
NLOGIT    ; Lhs  = mode
          ; Choices = air,train,bus,car
          ; Rhs   = invc,invt,gc,ttme
          ; Rh2 = one
          ; Prlist $
```

```
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Dependent variable               Choice     |
| Number of observations              200     |
| Log likelihood function       -174.8393     |
| Number of parameters                  7     |
| Info. Criterion: AIC =          1.81839     |
|   Finite Sample: AIC =          1.82131     |
| Info. Criterion: BIC =          1.93383     |
| Info. Criterion:HQIC =          1.86511     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -267.3168  .34595  .33823 |
| Chi-squared[ 4]          =    184.95510     |
| Prob [ chi squared > value ] =   .00000     |
| Response data are given as ind. choice.     |
| Number of obs.=   200, skipped   0 bad obs. |
+---------------------------------------------+
+---------------------------------------------+
| Notes No coefficients=> P(i,j)=1/J(i).      |
|       Constants only => P(i,j) uses ASCs    |
|         only. N(j)/N if fixed choice set.   |
|         N(j) = total sample frequency for j |
|         N    = total sample frequency.      |
|       These 2 models are simple MNL models. |
|       R-sqrd = 1 - LogL(model)/logL(other)  |
|       RsqAdj=1-[nJ/(nJ-nparm)]*(1-R-sqrd)   |
|         nJ   = sum over i, choice set sizes |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 INVC    |    -.08826012       .01987417    -4.441   .0000
 INVT    |    -.01344131       .00256769    -5.235   .0000
 GC      |     .07053307       .01778244     3.966   .0001
 TTME    |    -.10176138       .01117372    -9.107   .0000
 A_AIR   |    5.33347705       .92158644     5.787   .0000
 A_TRAIN |    4.44686822       .52777949     8.426   .0000
 A_BUS   |    3.69334154       .52916432     6.980   .0000

+---------------------------------------------+
| Discrete Choice (One Level) Model           |
| Model Simulation Using Previous Estimates   |
| Number of observations               10     |
+---------------------------------------------+
PREDICTED PROBABILITIES (* marks actual, + marks prediction.)
Indiv    AIR       TRAIN     BUS       CAR
    1   .0543     .0445     .7540*+   .1472
    2   .2402     .2189     .2014     .3395*+
    3   .0137     .0885     .8571*+   .0406
    4   .0203     .0890     .8287*+   .0620
    5   .4058 +   .1092     .3745*    .1105
    6   .2766     .3248 +   .2785     .1201*
    7   .6129*+   .1446     .1240     .1185
    8   .0824     .5444 +   .0648*    .3084
    9   .1815     .3629 +   .1795     .2761*
   10   .1958     .1863     .0514     .5665*+
```

This arrangement of the model may also include

; Describe
; Show Model to display the model configuration
; Effects: desired elasticities or marginal effects
; Prob = name to save probabilities
; Ivb = name to save inclusive values

All of these computations are done for the current sample. This process is the same as the full model computations listed earlier. But, with **; Prlist** in place, the model estimated previously is used; it is not reestimated.

6.6 Testing Hypotheses

We consider two types of hypothesis tests. The first is a specification test of the IID extreme value specification. The model assumptions induce the most prominent shortcoming of the multinomial logit model, the *independence from irrelevant alternatives* (IIA) property. The fact that the ratio of any two probabilities in the model involves only the utilities for those two models produces a number of undesirable implications, including the striking pattern in the elasticities in the model shown earlier. We consider a test of the IIA assumption. The second part of this section considers more conventional hypothesis tests about the coefficients in the model.

6.6.1 Testing the Assumption of Independence from Irrelevant Alternatives (IIA)

Hausman and McFadden (1984) have proposed a specification test for this model to test the inherent assumption of the independence from irrelevant alternatives (IIA). (IIA is a consequence of the initial assumption that the stochastic terms in the utility functions are independent and extreme value distributed. Discussion may be found in standard texts on qualitative choice modeling, such as Hensher, Rose and Greene (2005) and Greene (2011).) The procedure is, first, to estimate the model with all choices. The alternative specification is the model with a smaller set of choices. Thus, the model is estimated with this restricted set of alternatives and the same model specification. The set of observations is reduced to those in which one of the smaller set of choices is made. The test statistic is

$$q = [\mathbf{b}_r - \mathbf{b}_u]'[\mathbf{V}_r - \mathbf{V}_u]^{-1}[\mathbf{b}_r - \mathbf{b}_u],$$

where '*u*' and '*r*' indicate unrestricted and restricted (smaller choice set) models and **V** is an estimated variance matrix for the estimates. To use *NLOGIT* to carry out this test, it is necessary to estimate both models. In the second, it is necessary to drop the outcomes indicated. This is done with the

; Ias = list

specification. The list gives the names of the outcomes to be dropped.

This procedure is automated as shown in the following example:

```
CLOGIT      ; Lhs = mode
            ; Choices = air,train,bus,car
            ; Rhs = invc,invt,gc,ttme $
CLOGIT      ; Lhs = mode
            ; Choices = air,train,bus,car
            ; Ias = car
            ; Rhs = invc,invt,gc,ttme $
```

```
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Dependent variable               Choice     |
| Number of observations              210     |
| Log likelihood function       -244.1342     |
| Number of parameters                  4     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -283.7588  .13964  .13414 |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 INVC    |    -.02242963       .01435409    -1.563   .1181
 INVT    |    -.00634473       .00184168    -3.445   .0006
 GC      |     .03182946       .01372856     2.318   .0204
 TTME    |    -.03480667       .00469397    -7.415   .0000
+-----------------------------------------------------+
|WARNING:   Bad observations were found in the sample. |
|Found  59 bad observations among     210 individuals. |
|You can use ;CheckData to get a list of these points. |
+-----------------------------------------------------+
Normal exit from iterations. Exit status=0.
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Dependent variable               Choice     |
| Number of observations              151     |
| Log likelihood function       -103.2012     |
| Number of parameters                  4     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -159.0502  .35114  .34243 |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped  59 bad obs. |
+---------------------------------------------+
| Hausman test for IIA. Excluded choices are  |
| CAR                                         |
| ChiSqrd[ 4] =  51.9631, Pr(C>c) =  .000000  |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 INVC    |    -.04641792       .02108920    -2.201   .0277
 INVT    |    -.00963276       .00271137    -3.553   .0004
 GC      |     .04116251       .01984102     2.075   .0380
 TTME    |    -.07938809       .00991501    -8.007   .0000
```

In order to compute the coefficients in the restricted model, it is necessary to drop those observations that choose the omitted choice(s). In the example above, 59 observations were skipped. They are marked as bad data because with *car* excluded, no choice is made for those observations. As a consequence, the log likelihood functions are not comparable. The Hausman statistic is used to carry out the test. In the preceding example, the large value suggests that the IIA restriction should be rejected. Note that you can carry out several tests with different subsets of the choices without refitting the benchmark model. Thus, in the example above, you could follow with a third model in which **; Ias = bus** instead of **car**.

There is a possibility that restricting the choice set can lead to a singularity. It is possible that when you drop one or more alternatives, some attribute will be constant among the remaining choices. Thus, you might induce the case in which there is a 'regressor' which is constant across the choices. In this case, *NLOGIT* will issue a diagnostic about a singular Hessian (it is). Hausman and McFadden (1984) suggest estimating the model with the smaller number of choice sets *and* a smaller number of regressors. There is no question of consistency, or omission of a relevant attribute, since if the attribute is always constant among the choices, variation in it is obviously not affecting the choice. After estimation, the subvector of the larger parameter vector in the first model can be measured against the parameter vector from the second model using the Hausman statistic given earlier. This possibility arises in the model with alternative specific constants, so it is going to be a common case. The examples below suggest one way you might proceed in such as case.

The first step is to fit the original model using the entire sample and retrieve the results.

```
NLOGIT      ; Lhs  = mode
            ; Choices = air,train,bus,car
            ; Rhs  = invc,invt,gc,ttme,one $
MATRIX      ; bu   = b(1:4) ; vu = Varb(1:4,1:4) $
```

The variable choice takes values 1,2,3,4,1,2,3,4... indicating the indexing scheme for the choices

```
CREATE      ; choice = Trn(-4,0)$
```

Chair is a dummy variable that equals one for all four rows when choice made is *air*. Now restrict the sample to the observations for choices *train*, *bus*, *car*.

```
REJECT      ; chair = 1 | choice = 1 $
```

Fit the model with the restricted sample (choice set) and one less constant term.

```
NLOGIT      ; Lhs  = mode ; Choices = train,bus,car
            ; Rhs  = invc,invt,gc,ttme,one $
```

Retrieve the restricted results and compute the Hausman statistic.

```
MATRIX      ; br = b(1:4) ; vr = Varb(1:4,1:4)
            ; db = br - bu ; vdb = Nvsm(vr,-vu) $
CALC        ; List ; q = Qfr(db,vdb) ; 1 - Chi(q,4) $
```

The results are:

```
Q       =  .33784450384775710D+02
Result  =  .82501941289780950D-06
```

NOTE: (We've been asked this one several times.) The difference matrix in this calculation, *vdb*, might be nonsingular (have an inverse), but not be positive definite. In such a case, the chi squared can be negative. If this happens, the right conclusion is probably that it should be zero.

6.6.2 Lagrange Multiplier, Wald, and Likelihood Ratio Tests

NLOGIT keeps the usual statistics for the classical, Neyman-Pearson hypothesis tests. After estimation, the matrices *b* and *varb* will be kept as usual, and can be further manipulated for any purposes, for example, in the **WALD** command. You can use

; Test: ... restrictions

as well within the **NLOGIT** command to set up Wald tests of linear restrictions on the parameters. Likelihood ratio tests can be carried out by using the scalar *logl*, which will be available after estimation. The value of the log likelihood function for a model which contains only *J*-1 alternative specific constants will be reported in the output as well (see the sample outputs above). If your model actually contains the ASCs, *NLOGIT* will also report the chi squared test statistic and its significance level for the hypothesis that the other coefficients in the model are all 0.0.

HINT: *NLOGIT* can detect that a model contains a set of ASCs if you have used *one* in an **; Rhs** specification. But, it cannot determine from a set of dummy variables that you, yourself, provide, if they are a set of ASCs, because it inspects the model, not the data, to make the determination. As such, there is an advantage, when possible, to letting *NLOGIT* set up the set of alternative specific constants for you.

Finally, an LM statistic for testing the hypothesis that the starting values are not significantly different from the MLEs (the standard LM test) is requested by adding

; Maxit = 0

to the **NLOGIT** command.

Chapter 7: Simulating Probabilities in Discrete Choice Models

7.1 Introduction

The simulation program described here allows you to fit a model, use it to predict the set of choices for your sample, then examine how those choices would change if the attributes of the choices changed. You can also examine scenarios that involve restricting the choice set from the original one. Finally, you can use your estimated model and this simulator to do these analyses with data sets that were not actually used to fit the model. The calculation proceeds as follows:

Step 1. Set the desired sample for the model estimation. Estimate the model using *NLOGIT*. This processor is supported for the following discrete choice models that are specific to *NLOGIT*:

Model	Command	Alternative Command
Conditional Logit	**CLOGIT**	**NLOGIT**
Nested Logit	**NLOGIT**	**NLOGIT ; Tree = ...**
Random Parameters Logit	**RPLOGIT**	**NLOGIT ; RPL**
Multinomial Probit	**MNPROBIT**	**NLOGIT ; MNP**

Step 2. The model is viewed as a random utility model in which the utility functions are functions of attributes $x_1,...,x_K$. The model is then fit to describe the choice among J alternatives, $C_1,...,C_J$. This may be a very simple model such as the basic multinomial logit model (MNL) of Chapter 8 or as complicated as a four level nested logit model as described in Chapter 9. In any event, the model is ultimately viewed in terms of these attributes and choices.

Step 3. (If desired) Reset the sample to any desired setup that is consistent with the model. This may be all or a subset of the data used to fit the model, or a set of individuals that were not used in fitting the model, or any mixture of the two.

Step 4. Specify which of the choices (possibly but not necessarily all) are to be used as the choice set for the simulation. The simulation is then produced to predict choice among this possibly reduced set of choices. (Probabilities for the full choice set are reallocated, but not necessarily proportionally. This would only occur in the MNL model which satisfies IIA.)

Step 5. Specify how the attributes that enter the utility functions will change – for example that a particular price is to rise by 25%.

Step 6. Simulate the model by computing the probabilities and predicting the outcomes for the specified sample and summarize the results, comparing them to the original, base case.

Steps 3-6 may be repeated as many times as desired once a model has been estimated. The model is not reestimated; the existing model is used to compute the simulation results. The simulation produces an output table that compares absolute frequencies and shares for each alternative in the full or a restricted choice set to the base case in which the predicted shares are the means of the sample predictions from the model absent the changes specified in the scenario.

In addition, this feature provides a capability for implementing simulation/scenario analysis when one is using mixtures of data (for example stated preference and revealed preference). This option allows you to combine the two types of data in a simulation. An example is shown in the case study below.

7.2 Essential Subcommands

NLOGIT's models are all built around the specification which indicates the choice set being modeled:

; Choices = the full list of alternatives in the model

This simulation program is used to compute simulated probabilities assuming that the individuals in the sample being simulated are choosing among some or all of these alternatives. The first subcommand for the simulation is

; Simulation = a list of names of alternatives

The list of names must be some or all of the names in the **; Choices** list. If they are to be all of them, then you may use

; Simulation = * (or, just **; Simulation**)

NOTE: Simulation on a subset of alternatives in the full choice set is done by analyzing the full set of data while, in process, pretending (simulating) that alternatives not in the simulation list are not available to these individuals even if they are physically in the data set and actually available. (Note, this is just for the purposes of the simulation.) You must not change the sample settings in any way to produce this effect yourself. It is handled completely internally by this program simply by using a set of switches ('on' for included, 'off' for excluded) for the choice set while numerical results are computed.

The second specification you will provide is the name of the attribute that is being set or changed and the names of the alternatives in which this attribute is changing. This is the 'scenario.' The base case, for a single changing attribute is

; Scenario: attribute name (list of alternatives whose attribute levels will change)
= [action] magnitude of action

If you wish to include in the scenario, all the alternatives that are defined in the simulation, simply use the wildcard character, * as the list. Note that this 'all items in list' refers back to your **; Simulation** list, not to the **; Choices** list. The actions in the scenario specification are as follows:

	=	specific value to force the attribute to take this value in all cases,
or	=	[*] value to multiply observed values by the value,
or	=	[+] value to add 'value' to the observed values,
or	=	[/] value to divide the attribute by the specified value,
or	=	[-] value to subtract 'value' from the observed values.

The following example:

; Choices = air,train,bus,car
; Simulation = air,car
; Scenario: gc(car) = [*] 1.5

specifies a simulation over two choices in a four choice model. The scenario is enacted by changing the *gc* attribute for *car* only by multiplying whatever value is found in the original sample by 1.5.

7.3 Multiple Attribute Specifications and Multiple Scenarios

The simulation may specify that more than one attribute is to change. The multiple settings may provide for changes in different alternatives. The specification is

; Scenario: attribute name 1 (list of alternatives) = [action] magnitude of action /
attribute name 2 (list of alternatives) = [action] magnitude of action /
... repeated up to a maximum of 20 attributes specifications

The different change specifications are separated by slashes. To continue the earlier example, we might specify

; Choices = air,train,bus,car
; Simulation = air,train, car
; Scenario: gc(car) = [*] 1.5 /
ttme (air,train) = [*] 1.25

You may also provide more than one full scenario for the simulation. In this case, each scenario is compared to the base case, then the scenarios are compared to each other. You may compare up to five scenarios in one run with this tool. Use

; Scenario: attribute name 1 (list of alternatives) = [action] magnitude of action ...
&
attribute name 2 (list of alternatives) = [action] magnitude of action ...

Use ampersands (&) to separate the scenarios. Within each scenario, you may have up to 20 attribute specifications separated by slashes.

7.4 Simulation Commands

The simulation instruction does not produce new model estimates. However all other **NLOGIT** options can be invoked with the command, such as descriptive statistics and computing and retaining predicted probabilities.

7.4.1 Observations Used for the Simulations

The data set used in the simulation can be the original data set used to estimate the model or a new data set. The base model is fit with an 'estimation' data set. After this operation (Steps 1 and 2 in the introduction), if desired, you may respecify the sample to direct the simulator to do the calculations with a completely different set of observations. This would precede Step 4 above. If you do not change the sample setting, the same data are used for the simulation. (The simulation must follow the estimation. In any case, it will require a second command, which will generally be identical to the first save for the specification of the simulation.)

7.4.2 Variables Used for the Simulations

If a new data set is used, the attributes must have the exact same names and measurement units and the alternatives must also have the same names as the full or a restricted set of those used in model estimation. A natural application that would obey this convention would be to use one half of a sample to estimate the model, then repeat the simulation using the other half of the same sample.

7.4.3 Choices Simulated

One can undertake simulation either on the full choice set used in estimation or a restricted set. This latter option is very useful for modelers using mixtures of data (e.g., combined stated and revealed preference data), where some alternatives are only included in estimation but not in application. An extensive example is shown below in the case study.

7.4.4 Other *NLOGIT* Options

The routine that does simulation also allows you to compute the various elasticities and/or derivatives (**; Effects: ...**) and descriptive statistics (**; Describe** and **; Crosstab**) as described in Chapter 6, and will produce the standard results for these. You might already have done this at the estimation step, but if you change the sample, you can use this simulation program to recompute those values.

7.4.5 Observations Used for the Simulations

This program also allows you to compute, display, and save fitted probabilities, utilities and inclusive values for specific observations, using the standard setup for these as described in the *LIMDEP* documentation. Once again, this is likely to be useful when your estimation and simulation steps are based on different sets of observations.

7.5 Applications

We compute the shares for a particular sample using the following:

$$S(\text{alternative } j) \;=\; N \times \sum_{i=1}^{N} \hat{P}_{ij}\,.$$

Thus, save for the rounding error which is distributed, the model predicts the number of individuals in the sample who will choose each alternative. The crosstabulation described in Section 6.3 summarizes this calculation. For example, using the clogit.dat data, the following results from estimation of a simple multinomial logit model:

```
+-----------------------------------------------------+
| Cross tabulation of actual vs. predicted choices.    |
| Row indicator is actual, column is predicted.        |
| Predicted total is F(k,j,i)=Sum(i=1,...,N) P(k,j,i). |
| Column totals may be subject to rounding error.      |
+-----------------------------------------------------+
         AIR           TRAIN         BUS           CAR           Total
        +-------------------------------------------------------------------
AIR     |    34.0000        8.0000        4.0000       13.0000       58.0000
TRAIN   |     8.0000       39.0000        4.0000       12.0000       63.0000
BUS     |     5.0000        4.0000       17.0000        4.0000       30.0000
CAR     |    11.0000       13.0000        5.0000       30.0000       59.0000
Total   |    58.0000       63.0000       30.0000       59.0000      210.0000
```

The feature described here is used to examine how these predictions change when the value of an attribute changes. For example, how do the predictions change when the generalized cost of air travel changes. The simulator is used as follows:

Step 1. Fit the model.

Step 2. Use the identical model specification, but add to the command:

; Simulation [= a subset of the choices, if desired – see below]
; Scenario = what changes and how

We take the base case first, in which all alternatives are considered in the simulation. A scenario is defined using

; Scenario: attribute (choices in which it appears) = the change

as shown in the preceding section. The results of the computation will show the market shares before and after the change.

For example, we will refit the transport mode model examined at various points in Chapters 7 and 8, then examine the effect of increasing by 25% the terminal time spent waiting for air transport.

```
SAMPLE    ; 1 - 840 $
NLOGIT    ; Lhs = mode ; Rhs = one,gc,ttme ; Choices = air,train,bus,car $
NLOGIT    ; Lhs = mode ; Rhs = one,gc,ttme ; Choices = air,train,bus,car
          ; Simulation
          ; Scenario: ttme (air) = [*]1.25 $
```

The estimated model appears first, followed by the simulation.

```
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Maximum Likelihood Estimates                |
| Dependent variable               Choice     |
| Number of observations              210     |
| Log likelihood function       -199.9766     |
| Log-L for Choice   model =   -199.97662     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -283.7588  .29526  .28962 |
| Chi-squared[ 2]          =    167.56429     |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
+---------+--------------+----------------+--------+---------+
|Variable | Coefficient  | Standard Error |b/St.Er.|P[|Z|>z] |
+---------+--------------+----------------+--------+---------+
 GC       -.1578374521E-01  .43827919E-02   -3.601   .0003
 TTME     -.9709052295E-01  .10435090E-01   -9.304   .0000
 A_AIR        5.776358875       .65591872    8.807   .0000
 A_TRAIN      3.923001236       .44199360    8.876   .0000
 A_BUS        3.210734711       .44965283    7.140   .0000
```

These are the predictions of the model in the base case and after enacting the scenario.

```
+---------------------------------------------+
| Discrete Choice (One Level) Model           |
| Model Simulation Using Previous Estimates   |
| Number of observations              210     |
+---------------------------------------------+
+-------------------------------------------------------+
|Simulations of Probability Model                       |
|Model: Discrete Choice (One Level) Model               |
|Simulated choice set may be a subset of the choices.   |
|Number of individuals is the probability times the     |
|number of observations in the simulated sample.        |
|Column totals may be affected by rounding error.       |
|The model used was simulated with    210 observations.|
+-------------------------------------------------------+
-----------------------------------------------------------------------------
Specification of scenario 1 is:
Attribute  Alternatives affected            Change type             Value
---------  ------------------------------   ------------------- ---------
TTME       AIR                              Scale base by value     1.250
-----------------------------------------------------------------------------
The simulator located    209 observations for this scenario.
Simulated Probabilities (shares) for this scenario:
+----------+--------------+--------------+------------------+
|Choice    |     Base     |   Scenario   | Scenario - Base  |
|          |%Share Number |%Share Number |ChgShare ChgNumber|
+----------+--------------+--------------+------------------+
|AIR       | 27.619    58 | 15.118    32 |-12.501%      -26 |
|TRAIN     | 30.000    63 | 33.694    71 |  3.694%        8 |
|BUS       | 14.286    30 | 16.126    34 |  1.841%        4 |
|CAR       | 28.095    59 | 35.061    74 |  6.966%       15 |
|Total     |100.000   210 |100.000   211 |   .000%        1 |
+----------+--------------+--------------+------------------+
```

The model predicts the base case using the actual data, shown in the left side and what would become of this case if the scenario is assumed. In this case, each person's *ttme* for *air* travel is increased by 25%, and the probabilities are recomputed. In this case, a fairly strong effect is predicted. 26 of 58 people who chose *air* are now expected to take other modes, eight changing to *train*, four to *bus*, and 15 to *car*. (The one stray person at the end is the result of rounding error in the allocation of the probabilities.)

You may combine up to five scenarios in each simulation. This allows you to have simultaneous changes in attributes. Use

```
; Scenario :    attribute (choices in which it appears) = the change   /
                attribute (choices in which it appears) = the change   /
                ...
```

For example, suppose terminal time for both *air* and *train* both increased by 25%. We would extend our previous setup as follows:

```
SAMPLE      ; 1 - 840 $
NLOGIT      ; Lhs = mode ; Rhs = one,gc,ttme ; Choices = air,train,bus,car $
NLOGIT      ; Lhs = mode ; Rhs = one,gc,ttme ; Choices = air,train,bus,car
            ; Simulation
            ; Scenario:  ttme (air)    = [*] 1.25 /
                         ttme (train) = [*] 1.25  $
```

```
+---------------------------------------------+
| Discrete Choice (One Level) Model           |
| Model Simulation Using Previous Estimates   |
| Number of observations              210     |
+---------------------------------------------+
+-------------------------------------------------------+
|Simulations of Probability Model                       |
|Model: Discrete Choice (One Level) Model               |
|Simulated choice set may be a subset of the choices.   |
|Number of individuals is the probability times the     |
|number of observations in the simulated sample.        |
|Column totals may be affected by rounding error.       |
|The model used was simulated with    210 observations.|
+-------------------------------------------------------+
-----------------------------------------------------------------------------
Specification of scenario 1 is:
Attribute  Alternatives affected            Change type             Value
---------  ------------------------------   ------------------- ---------
TTME       AIR                              Scale base by value     1.250
TTME       TRAIN                            Scale base by value     1.250
-----------------------------------------------------------------------------
The simulator located    209 observations for this scenario.
Simulated Probabilities (shares) for this scenario:
```

```
+----------+--------------+--------------+------------------+
|Choice    |     Base     |   Scenario   | Scenario - Base  |
|          |%Share Number |%Share Number |ChgShare ChgNumber|
+----------+--------------+--------------+------------------+
|AIR       | 27.619    58 | 16.417    34 |-11.202%      -24 |
|TRAIN     | 30.000    63 | 23.178    49 | -6.822%      -14 |
|BUS       | 14.286    30 | 18.796    39 |  4.510%        9 |
|CAR       | 28.095    59 | 41.609    87 | 13.514%       28 |
|Total     |100.000   210 |100.000   209 |   .000%       -1 |
+----------+--------------+--------------+------------------+
```

You may also compare the effects of different scenarios. For example, rather than assume that *ttme* for both *air* and *train* changed, you might compare the two scenarios. To do a pairwise comparison of scenarios, separate them with '&' in the command. For example,

```
NLOGIT      ; Lhs = mode ; Rhs = one,gc,ttme
            ; Choices = air,train,bus,car
            ; Simulation
            ; Scenario:  ttme (air)    = [*] 1.25
                      & ttme (train) = [*] 1.25  $
```

produces the separate results, then the pairwise comparison:

```
+-------------------------------------------------------+
|Simulations of Probability Model                       |
|Model: Discrete Choice (One Level) Model               |
|Simulated choice set may be a subset of the choices.   |
|Number of individuals is the probability times the     |
|number of observations in the simulated sample.        |
|Column totals may be affected by rounding error.       |
|The model used was simulated with    210 observations.|
+-------------------------------------------------------+
-------------------------------------------------------------------------
Specification of scenario 1 is:
Attribute  Alternatives affected            Change type             Value
---------  ------------------------------  ------------------- ---------
TTME       AIR                              Scale base by value     1.250
-------------------------------------------------------------------------
The simulator located    209 observations for this scenario.
Simulated Probabilities (shares) for this scenario: (Note rounding error)
+----------+--------------+--------------+------------------+
|Choice    |     Base     |   Scenario   | Scenario - Base  |
|          |%Share Number |%Share Number |ChgShare ChgNumber|
+----------+--------------+--------------+------------------+
|AIR       | 27.619    58 | 15.118    32 |-12.501%      -26 |
|TRAIN     | 30.000    63 | 33.694    71 |  3.694%        8 |
|BUS       | 14.286    30 | 16.126    34 |  1.841%        4 |
|CAR       | 28.095    59 | 35.061    74 |  6.966%       15 |
|Total     |100.000   210 |100.000   211 |   .000%        1 |
+----------+--------------+--------------+------------------+
```

```
-----------------------------------------------------------------------------
Specification of scenario 2 is:
Attribute  Alternatives affected            Change type             Value
---------  ------------------------------  ------------------- ---------
TTME       TRAIN                            Scale base by value     1.250
-----------------------------------------------------------------------------
The simulator located    209 observations for this scenario.
Simulated Probabilities (shares) for this scenario:
+----------+--------------+--------------+------------------+
|Choice    |     Base     |   Scenario   | Scenario - Base  |
|          |%Share Number |%Share Number |ChgShare ChgNumber|
+----------+--------------+--------------+------------------+
|AIR       | 27.619    58 | 30.168    63 |  2.548%        5 |
|TRAIN     | 30.000    63 | 20.787    44 | -9.213%      -19 |
|BUS       | 14.286    30 | 16.383    34 |  2.097%        4 |
|CAR       | 28.095    59 | 32.662    69 |  4.567%       10 |
|Total     |100.000   210 |100.000   210 |   .000%        0 |
+----------+--------------+--------------+------------------+
The simulator located    209 observations for this scenario.
Pairwise Comparisons of Specified Scenarios
Base     for this comparison is scenario 1.
Scenario for this comparison is scenario 2.
+----------+--------------+--------------+------------------+
|Choice    |     Base     |   Scenario   | Scenario - Base  |
|          |%Share Number |%Share Number |ChgShare ChgNumber|
+----------+--------------+--------------+------------------+
|AIR       | 15.118    32 | 30.168    63 | 15.049%       31 |
|TRAIN     | 33.694    71 | 20.787    44 |-12.907%      -27 |
|BUS       | 16.126    34 | 16.383    34 |   .257%        0 |
|CAR       | 35.061    74 | 32.662    69 | -2.399%       -5 |
|Total     |100.000   211 |100.000   210 |   .000%       -1 |
+----------+--------------+--------------+------------------+
```

Finally, you can use the simulator to restrict the choice set. The computed probabilities are computed assuming only the specified alternatives are available. To do this, use

; Scenario = the subset of alternatives

To continue the example, we simulate the model assuming that people could not drive, and examine what the effect of increasing terminal time in airports would do to the market shares for the remaining three alternatives.

```
SAMPLE     ; 1 - 840 $
NLOGIT     ; Lhs = mode ; Rhs = one,gc,ttme ; Choices = air,train,bus,car $
NLOGIT     ; Lhs = mode ; Rhs = one,gc,ttme ; Choices = air,train,bus,car
           ; Simulation = air,train,bus
           ; Scenario: ttme (air)  = [*] 1.25  $
```

```
+---------------------------------------------+
| Discrete Choice (One Level) Model           |
| Model Simulation Using Previous Estimates   |
| Number of observations              210     |
+---------------------------------------------+
+-------------------------------------------------------+
|Simulations of Probability Model                       |
|Model: Discrete Choice (One Level) Model               |
|Simulated choice set may be a subset of the choices.   |
|Number of individuals is the probability times the     |
|number of observations in the simulated sample.        |
|Column totals may be affected by rounding error.       |
|The model used was simulated with    210 observations.|
+-------------------------------------------------------+
-----------------------------------------------------------------------------
Specification of scenario 1 is:
Attribute  Alternatives affected            Change type             Value
---------  ------------------------------   ------------------- ---------
TTME       AIR                              Scale base by value     1.250
-----------------------------------------------------------------------------
The simulator located    209 observations for this scenario.
Simulated Probabilities (shares) for this scenario:
+----------+--------------+--------------+------------------+
|Choice    |     Base     |   Scenario   | Scenario - Base  |
|          |%Share Number |%Share Number |ChgShare ChgNumber|
+----------+--------------+--------------+------------------+
|AIR       | 39.353    83 | 22.933    48 |-16.420%      -35 |
|TRAIN     | 40.985    86 | 52.281   110 | 11.297%       24 |
|BUS       | 19.662    41 | 24.786    52 |  5.123%       11 |
|Total     |100.000   210 |100.000   210 |   .000%        0 |
+----------+--------------+--------------+------------------
```

Chapter 8: The Multinomial Logit Model

8.1 Introduction

In the multinomial logit model, there is a single vector of characteristics, which describes the individual, and a set of J parameter vectors. In the 'discrete choice' setting of this section, these are essentially reversed. The J alternatives are each characterized by a set of K 'attributes,' $\mathbf{x}_{ij}$. Respondent 'i' chooses among the J alternatives. There is a single parameter vector, $\boldsymbol{\beta}$. The model underlying the observed data is assumed to be the following random utility specification:

$$U(\text{choice } j \text{ for individual } i) \;=\; U_{ij} \;=\; \boldsymbol{\beta}'\mathbf{x}_{ij} \;+\; \varepsilon_{ij},\; j = 1,\ldots,J_i.$$

The random, individual specific terms, $(\varepsilon_{i1},\varepsilon_{i2},\ldots,\varepsilon_{iJ})$ are assumed to be independently distributed, each with an extreme value distribution. Under these assumptions, the probability that individual i chooses alternative j is

$$\text{Prob}(U_{ij} > U_{iq}) \text{ for all } q \neq j.$$

It has been shown that for independent extreme value distributions, as above, this probability is

$$\text{Prob}(y_i = j) \;=\; \frac{\exp\left(\boldsymbol{\beta}'\mathbf{x}_{ij}\right)}{\sum_{m=1}^{J_i}\exp\left(\boldsymbol{\beta}'\mathbf{x}_{im}\right)}$$

where y_i is the index of the choice made. Regardless of the number of choices, there is a single vector of K parameters to be estimated. This model does not suffer from the proliferation of parameters that appears in the logit model described in Chapter 4. It does, however, make the very strong 'Independence from Irrelevant Alternatives' assumption which will be discussed below.

NOTE: The distinction made here between 'discrete choice' and 'multinomial logit' is not hard and fast. It is made purely for convenience in the discussion. By interacting the characteristics with the alternative specific constants, the discrete choice model of this chapter becomes the multinomial logit model. From this point, in the remainder of this reference guide for *NLOGIT*, we will refer to the model described in this chapter, with mathematical formulation as given above, as the 'multinomial logit model,' or MNL model as is common in the literature.

The basic setup for this model consists of observations on n individuals, each of whom makes a single choice among J_i choices, or alternatives. There is a subscript on J_i because we do not restrict the choice sets to have the same number of choices for every individual. The data will typically consist of the choices and observations on K 'attributes' for each choice. The attributes that describe each choice, i.e., the arguments that enter the utility functions, may be the same for all choices, or may be defined differently for each utility function. The estimator described in this chapter allows a large number of variations of this basic model. In the discrete choice framework, the observed 'dependent variable' usually consists of an indicator of which among J_i alternatives was *most* preferred by the respondent. All that is known about the others is that they were judged inferior to the one chosen. But, there are cases in which information is more complete and consists of a subjective ranking of all J_i alternatives by the individual. *NLOGIT* allows specification of the model for estimation with 'ranks data.' In addition, in some settings, the sample data might consist of aggregates for the choices, such as proportions (market shares) or frequency counts. *NLOGIT* will accommodate these cases as well.

8.2 Command for the Multinomial Logit Model

The simplest form of the command for the discrete choice models is

```
CLOGIT      ; Lhs  = variable which indicates the choice made
            ; Choices = a set of J names for the set of choices
            ; Rhs  = choice varying attributes in the utility functions
            ; Rh2 = choice invariant characteristics $
```

(With no qualifiers to indicate a different model, such as RPL or MNP, **CLOGIT** and **NLOGIT** are the same.) There are various ways to specify the utility functions – i.e., the right hand sides of the equations that underlie the model, and several different ways to specify the choice set. The **; Rhs** specification may be replaced with an explicit definition of the utility functions, using **; Model ...**

A set of exactly J choice labels must be provided in the command. These are used to label the choices in the output. The number you provide is used to determine the number of choices there are in the model. Therefore, the set of the right number of labels is essential. Use any descriptor of eight or fewer characters desired – these do not have to be valid names, just a set of labels, separated in the list by commas.

The command builder for this model is found in Model:Discrete Choice/Discrete Choice. The Main and Options pages are both used to set up the model. The model and the choice set are defined in the Main page; the attributes are defined in the Options page. See Figure 8.1.

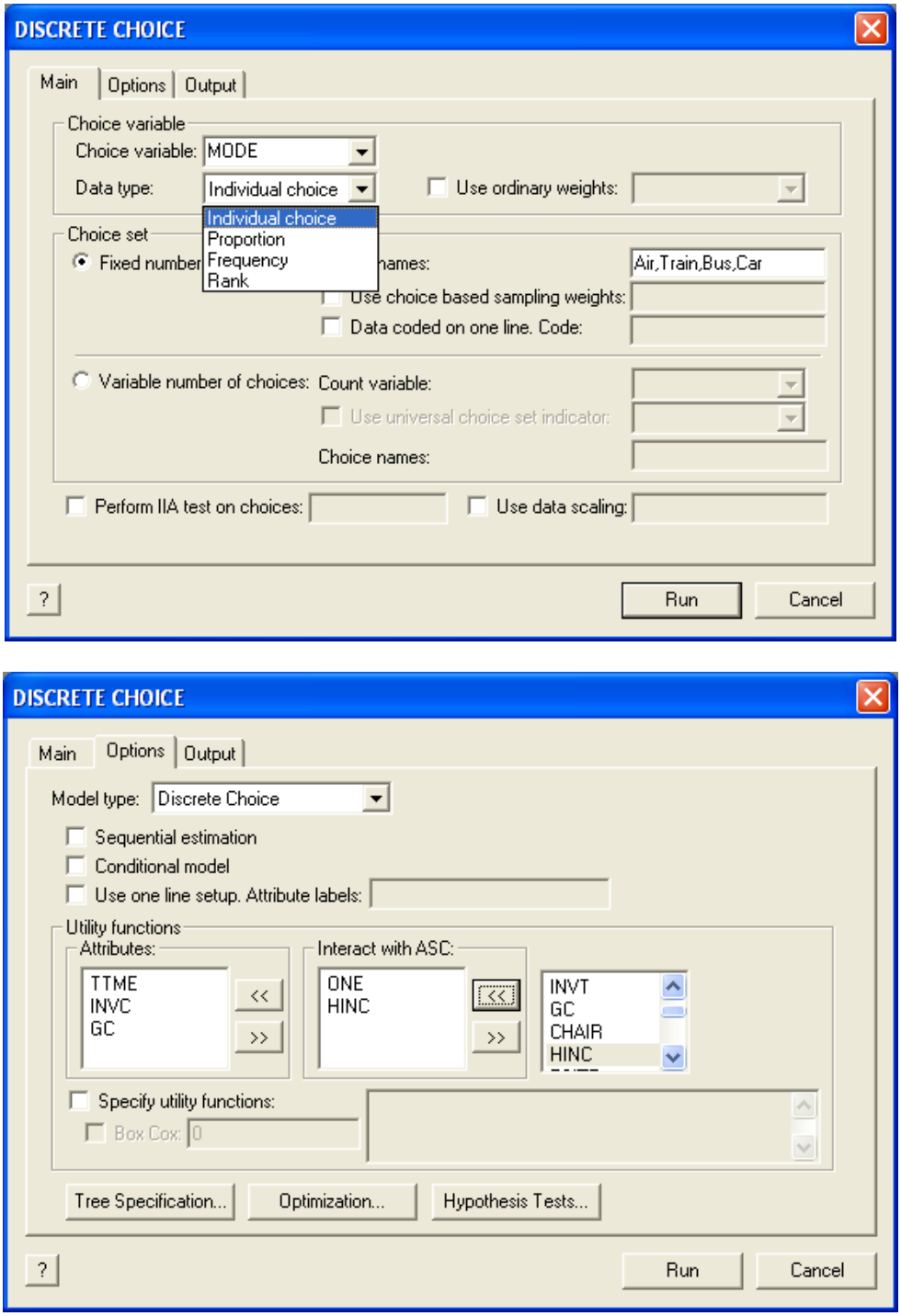


Figure 8.1 Command Builder for Multinomial Logit Model

8.3 Results for the Multinomial Logit Model

Results for the multinomial logit model will consist of the standard model results and any additional descriptive output you have requested. The application below will display the full set of available results. Results kept by this estimator are:

Matrices:	*b* and *varb*	=	coefficient vector and asymptotic covariance matrix
Scalars:	*logl*	=	log likelihood function
	nreg	=	N, the number of observational units
	kreg	=	the number of Rhs variables
Last Model:	*b_variable*	=	the labels kept for the **WALD** command.

In the *Last Model*, groups of coefficients for variables that are integrated with constants get labels *choice_variable*, as in *trai_gco*. (Note that the names are truncated – up to four characters for the choice and three for the attribute.) The alternative specific constants are *a_choice*, with names truncated to no more than six characters. For example, the sum of the three estimated choice specific constants could be analyzed as follows:

WALD ; Fn1 = a_air + a_train + a_bus $

```
+-----------------------------------------------+
| WALD procedure. Estimates and standard errors |
| for nonlinear functions and joint test of     |
| nonlinear restrictions.                       |
| Wald Statistic             =     57.91928     |
| Prob. from Chi-squared[ 1] =       .00000     |
+-----------------------------------------------+
+---------+--------------+----------------+--------+---------+
|Variable | Coefficient  | Standard Error |b/St.Er.|P[|Z|>z] |
+---------+--------------+----------------+--------+---------+
 Fncn(1)      13.32858178       1.7513477    7.610   .0000
```

8.4 Application

The MNL model based on the CLOGIT data is estimated with the command

```
CLOGIT      ; Lhs  = mode
            ; Choices = air,train,bus,car
            ; Rhs  = gc,ttme
            ; Rh2 = one,hinc
            ; Show Model
            ; Describe
            ; Crosstab
            ; Effects: gc(*)
            ; Ivb    = incvlu
            ; Prob = pmnl
            ; List $
```

This requests all the optional output from the model. The **; Describe** specification detailed in Chapter 6 requests a set of descriptive statistics for the variables in the model, by choice. The leftmost set of results gives the coefficient estimates. Note that in this model, they are the same for the two generic coefficients, on *gc* and *ttme*, but they vary by choice for the alternative specific constant and its interaction with income. Also, since there is no ASC for *car* (it was dropped to avoid the dummy variable trap), there are no coefficients for the car grouping. The second set of values in the center section gives the mean and standard deviation for that attribute in that outcome for all observations in the sample. The third set of results gives the mean and variance for the particular attribute for the individuals that made that choice. The full set of results from the model is as follows.

```
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
+---------------------------------------------+
Sample proportions are marginal, not conditional.
Choices marked with * are excluded for the IIA test.
+----------------+------+---
|Choice   (prop.)|Weight|IIA
+----------------+------+---
|AIR       .27619| 1.000|
|TRAIN     .30000| 1.000|
|BUS       .14286| 1.000|
|CAR       .28095| 1.000|
+----------------+------+---
+-----------------------------------------------------------------+
| Model Specification:  Table entry is the attribute that         |
| multiplies the indicated parameter.                             |
+--------+------+----------------------------------------------+
| Choice |******| Parameter                                    |
|        |Row  1| GC       TTME     A_AIR    AIR_HIN1 A_TRAIN   |
|        |Row  2| TRA_HIN2 A_BUS    BUS_HIN3                    |
+--------+------+----------------------------------------------+
|AIR     |     1| GC       TTME     Constant HINC     none      |
|        |     2| none     none     none                        |
|TRAIN   |     1| GC       TTME     none     none     Constant  |
|        |     2| HINC     none     none                        |
|BUS     |     1| GC       TTME     none     none     none      |
|        |     2| none     Constant HINC                        |
|CAR     |     1| GC       TTME     none     none     none      |
|        |     2| none     none     none                        |
+-----------------------------------------------------------------+
Normal exit from iterations. Exit status=0.
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Dependent variable               Choice     |
| Number of observations              210     |
| Log likelihood function       -189.5252     |
| Number of parameters                  8     |
| Info. Criterion: AIC =          1.88119     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -283.7588  .33209  .32350 |
| Chi-squared[ 5]          =    188.46723     |
| Prob [ chi squared > value ] =   .00000     |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
```

```
+---------------------------------------------+
| Notes No coefficients=> P(i,j)=1/J(i).      |
|       Constants only => P(i,j) uses ASCs    |
|         only. N(j)/N if fixed choice set.   |
|         N(j) = total sample frequency for j |
|         N    = total sample frequency.      |
|       These 2 models are simple MNL models. |
|       R-sqrd = 1 - LogL(model)/logL(other)  |
|       RsqAdj=1-[nJ/(nJ-nparm)]*(1-R-sqrd)   |
|         nJ   = sum over i, choice set sizes |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 GC      |    -.01092735       .00458775    -2.382   .0172
 TTME    |    -.09546055       .01047320    -9.115   .0000
 A_AIR   |    5.87481336       .80209034     7.324   .0000
 AIR_HIN1|    -.00537349       .01152940     -.466   .6412
 A_TRAIN |    5.54985728       .64042443     8.666   .0000
 TRA_HIN2|    -.05656186       .01397335    -4.048   .0001
 A_BUS   |    4.13028388       .67636278     6.107   .0000
 BUS_HIN3|    -.02858418       .01544418    -1.851   .0642

PREDICTED PROBABILITIES (* marks actual, + marks prediction.)
Indiv    AIR       TRAIN     BUS       CAR
    1   .0984     .3311     .1959     .3746*+
    2   .2566     .2262     .0530     .4641*+
    3   .1401     .1795     .1997     .4808*+
    4   .2732     .0297     .0211     .6759*+
    5   .3421     .1478     .0527     .4575*+
    6   .0831     .3962*+   .2673     .2534
    7   .6066*+   .0701     .0898     .2335
    8   .0626     .6059 +   .1925     .1390*
    9   .1125     .2932     .1995     .3947*+
   10   .1482     .0804     .1267     .6447*+
```

(Rows 11 – 210 are omitted.)

```
+-------------------------------------------------------+
| Cross tabulation of actual vs. predicted choices.     |
| Row indicator is actual, column is predicted.         |
| Predicted total is F(k,j,i)=Sum(i=1,...,N) P(k,j,i). |
| Column totals may be subject to rounding error.       |
+-------------------------------------------------------+

Matrix Crosstab has  5 rows and  5 columns.
         AIR           TRAIN         BUS           CAR           Total
        +-----------------------------------------------------------------
AIR     |   33.00000      7.00000      4.00000     14.00000     58.00000
TRAIN   |    7.00000     39.00000      5.00000     12.00000     63.00000
BUS     |    3.00000      6.00000     15.00000      6.00000     30.00000
CAR     |   15.00000     11.00000      6.00000     27.00000     59.00000
Total   |   58.00000     63.00000     30.00000     59.00000    210.00000
```

```
+-----------------------------------------------------+
| Elasticity             averaged over observations.|
| Attribute is GC       in choice AIR                 |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.    |
|                                  Mean    St.Dev   |
| *     Choice=AIR               -.8019     .3834   |
|       Choice=TRAIN              .3198     .3370   |
|       Choice=BUS                .3198     .3370   |
|       Choice=CAR                .3198     .3370   |
+-----------------------------------------------------+
+-----------------------------------------------------+
| Elasticity             averaged over observations.|
| Attribute is GC       in choice TRAIN               |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.    |
|                                  Mean    St.Dev   |
|       Choice=AIR                .3534     .3511   |
| *     Choice=TRAIN            -1.0693     .7134   |
|       Choice=BUS                .3534     .3511   |
|       Choice=CAR                .3534     .3511   |
+-----------------------------------------------------+
+-----------------------------------------------------+
| Elasticity             averaged over observations.|
| Attribute is GC       in choice BUS                 |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.    |
|                                  Mean    St.Dev   |
|       Choice=AIR                .1679     .2308   |
|       Choice=TRAIN              .1679     .2308   |
| *     Choice=BUS              -1.0916     .5183   |
|       Choice=CAR                .1679     .2308   |
+-----------------------------------------------------+
+-----------------------------------------------------+
| Elasticity             averaged over observations.|
| Attribute is GC       in choice CAR                 |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.    |
|                                  Mean    St.Dev   |
|       Choice=AIR                .2934     .2674   |
|       Choice=TRAIN              .2934     .2674   |
|       Choice=BUS                .2934     .2674   |
| *     Choice=CAR               -.7492     .4430   |
+-----------------------------------------------------+

+-----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative AIR                 :
|     Utility Function          |                    |     58.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose AIR      |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| -------------------  -------- | -------------------+------------------- |
| GC           -.0109  GC       |  102.648     30.575|  113.552    33.198 |
| TTME         -.0955  TTME     |   61.010     15.719|   46.534    24.389 |
| A_AIR        5.8748  ONE      |    1.000       .000|    1.000      .000 |
| AIRxHIN1     -.0054  HINC     |   34.548     19.711|   41.724    19.115 |
+-----------------------------------------------------------------------+
```

```
+----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative TRAIN              :
|     Utility Function          |                     |     63.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose TRAIN    |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| -------------------  -------- | -------------------+------------------- |
| GC           -.0109  GC       |  130.200     58.235|  106.619    49.601 |
| TTME         -.0955  TTME     |   35.690     12.279|   28.524    19.354 |
| A_TRAIN      5.5499  ONE      |    1.000       .000|    1.000      .000 |
| TRAxHIN2     -.0566  HINC     |   34.548     19.711|   23.063    17.287 |
+----------------------------------------------------------------------+
+----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative BUS                :
|     Utility Function          |                     |     30.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose BUS      |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| -------------------  -------- | -------------------+------------------- |
| GC           -.0109  GC       |  115.257     44.934|  108.133    43.244 |
| TTME         -.0955  TTME     |   41.657     12.077|   25.200    14.919 |
| A_BUS        4.1303  ONE      |    1.000       .000|    1.000      .000 |
| BUSxHIN3     -.0286  HINC     |   34.548     19.711|   29.700    16.851 |
+----------------------------------------------------------------------+
+----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative CAR                :
|     Utility Function          |                     |     59.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose CAR      |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| -------------------  -------- | -------------------+------------------- |
| GC           -.0109  GC       |   95.414     46.827|   89.085    49.833 |
| TTME         -.0955  TTME     |     .000       .000|     .000      .000 |
+----------------------------------------------------------------------+
```

8.5 Marginal Effects

We define the marginal effects in the multinomial logit model as the derivatives of the probability of choice *j* with respect to attribute *k* in alternative *m*. This is

$$\frac{\partial P_j}{\partial x_{km}} = \left[\mathbf{1}(j = m) - P_m\right] P_j \beta_k ,$$

where the function $\mathbf{1}(j = m)$ equals one if *j* equals *m* and zero otherwise. These are naturally scaled since the probability is bounded. They are usually very small, so *NLOGIT* reports 100 times the value obtained, as in the example below, which is produced by

; Effects: gc[air]

```
+---------------------------------------------------+
| Derivative (times 100) averaged over observations.|
| Attribute is GC       in choice AIR               |
| Effects on probabilities of all choices in model: |
| * = Direct Derivative effect of the attribute.    |
|                                  Mean    St.Dev   |
| *     Choice=AIR               -.1339     .0880   |
|       Choice=TRAIN              .0362     .0309   |
|       Choice=BUS                .0204     .0204   |
|       Choice=CAR                .0773     .0763   |
+---------------------------------------------------+
```

Derivatives and elasticities are obtained by averaging the observation specific values, rather than by computing them at the sample means. The listing reports the sample mean (average partial effect) and the sample standard deviation.

It is common to report elasticities rather than the derivatives. These are

$$\frac{\partial \log P_j}{\partial \log x_{km}} = \left[\mathbf{1}(j = m) - P_m\right] x_{km} \beta_k \,.$$

The example below shows the counterpart to the preceding results produced by

; Effects: gc(air)

which requests a table of elasticities for the effect of changing *gc* in the *air* alternative.

```
+---------------------------------------------------+
| Elasticity             averaged over observations.|
| Attribute is GC       in choice AIR               |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.    |
|                                  Mean    St.Dev   |
| *     Choice=AIR               -.8019     .3834   |
|       Choice=TRAIN              .3198     .3370   |
|       Choice=BUS                .3198     .3370   |
|       Choice=CAR                .3198     .3370   |
+---------------------------------------------------+
```

The difference between the two commands is the use of '[*air*]' for derivatives and '(*air*)' for elasticities. The full set of tables, one for each alternative, is requested with

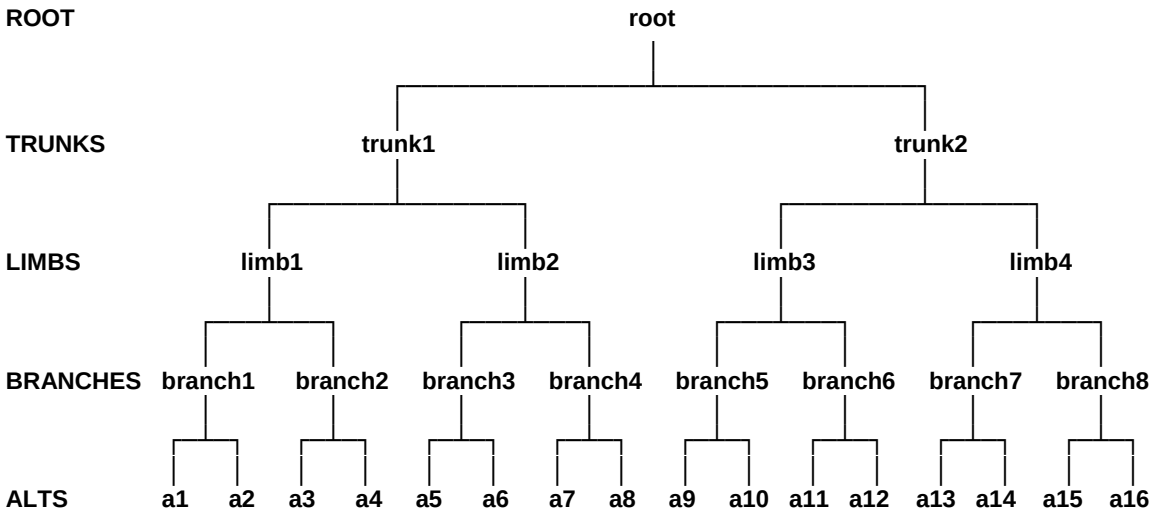
alternative[*] or **alternative(*)**.

Note that for this model, the elasticities take only two values, the 'own' value when *j* equals *m* and the 'cross' elasticity when *j* is not equal to *m*. The fact that the cross elasticities are all the same is one of the undesirable consequences of the IIA property of this model.

Chapter 9: The Nested Logit Model

9.1 Introduction

The nested logit model is an extension of the multinomial model presented in Chapter 8. The models described here are based on variations of a four level tree structure such as the following:



Individuals are assumed to make a choice among NALT = *J* alternatives (alts) in a choice set. The 'twigs' in the tree are the elemental alternatives in the choice set. There may be up to 100 alternatives in the model, a total of 25 branches throughout the tree, 10 limbs, and five trunks. The model may contain one or more limbs. Each limb may contain one or more branches, and each branch may contain one or more twigs (choices). If there is only one trunk and one limb, the model is, by implication, a two level model. As for single level models, choice sets may vary by individual. However, in order to construct a tree for such a setting, a universal choice set, as described in Chapter 5, is necessary. The variable sized choice set is then indicated by setting up the full tree structure, and indicating that certain choices are unavailable for the particular individual.

The command for fitting nested logit models is the same as described in Chapter 3 for one level models, save for the addition of the tree definition in the command and, optionally, the specification of additional utility functions for choices made at higher levels in the tree. The nested logit model is limited to four level models for full information maximum likelihood (FIML) estimation. It also allows estimation of two and higher level models by sequential, or two step estimation.

Utility functions can be specified for trunks the same as for limbs and branches (though it is unlikely that there will be very many attributes at this level in a tree). All options are available, including logs, Box-Cox transformation, fixed values, starting values, trunk specific constants, interaction terms, and so on. Utility functions for the trunks may include up to 10 variables including the set of constant terms if used. Since the command structure and options for the nested logit model are the same as those for the one level model, we will present in this chapter only the parts of the command setup that are specific to nested models. All users of this program should read Chapters 2-6 before proceeding.

9.2 Mathematical Specification of the Model

Individuals are assumed to choose one of the alternatives at the lowest level of the tree. Thus, they also choose a branch, a limb and a trunk. We denote by $j|b,l,r$ the choice of alternative j in branch b in limb l in trunk r. The number of alternatives in the branch/limb/trunk, $N_{b|l,r}$, can vary in every branch, limb, and trunk, and the number of branches in the l,rth limb/trunk, $N_{l|r}$ is likely to vary across limbs and trunks as well. No assumption of equal choice set sizes is made at any point in the following. (Note that for ease of presentation, we have dropped the observation subscript.)

The choice probability defined in Chapter 8 is now redefined to be the conditional probability of alternative j in branch b, limb l, and trunk r, $j|b,l,r$:

$$P(j|b,l,r) = \frac{\exp(\boldsymbol{\beta}'\mathbf{x}_{j|b,l,r})}{\sum_{q|b,l,r}\exp(\boldsymbol{\beta}'\mathbf{x}_{q|b,l,r})} = \frac{\exp(\boldsymbol{\beta}'\mathbf{x}_{j|b,l,r})}{\exp(J_{b|l,r})},$$

where $J_{b|l,r}$ is the *inclusive value* for branch b in limb l, trunk r, $J_{b|l,r} = \log \Sigma_{q|b,l,r} \exp(\boldsymbol{\beta}'\mathbf{x}_{q|b,l,r})$. At the next level up the tree, we define the conditional probability of choosing a particular branch in limb l, trunk r,

$$P(b|l,r) = \frac{\exp(\boldsymbol{\alpha}'\mathbf{y}_{b|l,r} + \tau_{b|l,r}J_{b|l,r})}{\sum_{s|l,r}\exp(\boldsymbol{\alpha}'\mathbf{y}_{s|l,r} + \tau_{s|l,r}J_{s|l,r})} = \frac{\exp(\boldsymbol{\alpha}'\mathbf{y}_{b|l,r} + \tau_{b|l,r}J_{b|l,r})}{\exp(I_{l|r})},$$

where $I_{l|r}$ is the inclusive value for limb l in trunk r, $I_{l|r} = \log \Sigma_{s|l,r} \exp(\boldsymbol{\alpha}'\mathbf{y}_{s|l,r} + \tau_{s|l,r}J_{s|l,r})$. The probability of choosing limb l in trunk r is

$$P(l|r) = \frac{\exp(\boldsymbol{\delta}'\mathbf{z}_{l|r} + \sigma_{l|r}I_{l|r})}{\sum_{s|r}\exp(\boldsymbol{\delta}'\mathbf{z}_{s|r} + \sigma_{s|r}I_{s|r})} = \frac{\exp(\boldsymbol{\delta}'\mathbf{z}_{l|r} + \sigma_{l|r}I_{l|r})}{\exp(H_r)},$$

where H_r is the inclusive value for trunk r, $H_r = \log \Sigma_{s|r} \exp(\boldsymbol{\delta}'\mathbf{z}_{s|r} + \sigma_{s|r} I_{s|r})$. Finally, the probability of choosing a particular limb, r, is

$$P(r) = \frac{\exp(\boldsymbol{\theta}'\mathbf{h}_r + \phi_r H_r)}{\sum_s \exp(\boldsymbol{\theta}'\mathbf{h}_s + \phi_s H_s)}.$$

By the laws of probability, the unconditional probability of the observed choice made by an individual is

$$P(j,b,l,r) = P(j|b,l,r) \times P(b|l,r) \times P(l|r) \times P(r).$$

This is the contribution of an individual observation to the likelihood function for the sample.

The 'nested logit' aspect of the model arises when any of the $\tau_{j|i,l}$ or $\sigma_{i|l}$ or ϕ_l differ from 1.0. If all of these deep parameters are set equal to 1.0, the unconditional probability specializes to

$$P(j,bj,l,r) = \frac{\exp(\boldsymbol{\beta}'\mathbf{x}_{j|b,l,r} + \boldsymbol{\alpha}'\mathbf{y}_{b|l,r} + \boldsymbol{\delta}'\mathbf{z}_{l|r} + \boldsymbol{\theta}'\mathbf{h}_r)}{\sum_r \sum_l \sum_b \sum_j \exp(\boldsymbol{\beta}'\mathbf{x}_{jmb\ nl,r} + \boldsymbol{\alpha}'\mathbf{y}_{b,l,r} + \boldsymbol{\delta}'\mathbf{z}_{l,r} + \boldsymbol{\theta}'\mathbf{h}_r)},$$

which is the probability for a one level model. The model is written in a very general form. The parameters of the model are, in exactly this order:

$$\beta_1,\beta_2,...,\beta_{nx},\alpha_1,\alpha_2,...,\alpha_{ny},\delta_1,\delta_2,...\delta_{nz},\theta_1,\theta_2,...,\theta_{nh},\tau_1...\tau_B,\sigma_1...,\sigma_L,\phi_1,...,\phi_R$$

where B is the total number of branches in the model, L is the number of limbs, and R is the number of trunks in the model. The **x**, **y**, **z**, and **h** vectors in the formulation above include all basic variables as well as all variables that interact with choice, branch, or limb specific dummy variables, etc. Once again, in this form, there may be different utility functions for each choice and, as described below, different utility functions defined for branches and limbs.

There is a vector of 'shallow' parameters, $[\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{\delta},\boldsymbol{\theta}]$ at each level, which multiplies the attributes (at the lowest level), or, e.g., demographics, at a higher level. There are also three vectors of 'deep' parameters, which multiply the inclusive values at the middle and high levels. In principle, there is one free inclusive value parameter for each branch in the model ($J_{b|l,r}$), one for each limb ($\sigma_{l|r}$), and one for each trunk (ϕ_r). But, some may have to be restricted to equal 1.0 for identification purposes. There are some degenerate cases:

- If the model has one trunk, then the one ϕ equals 1.0.
- If the model has one limb in a trunk, the one σ also equals 1.0.
- If a limb contains a single branch, the τ for that branch equals 1.0.

9.3 Commands for FIML Estimation

This section will describe how to set up a nested logit model. The default estimation technique is full information maximum likelihood (FIML). That is, the entire model is estimated in a single pass. .

9.3.1 Data Setup

The arrangement of the data set for estimation of the nested logit model is exactly the same as shown in Chapter 5. There is no requirement that the choice sets be the same across individuals, but the nested logit model will require a definition of a universal choice set, so the command must contain the

; Choices = list of labels ...

specification. The nested model structure does mandate one special consideration if you are going to define utility functions for branches (*y*s), or limbs (*z*s). Since you have one line of data for each alternative, you will have more than one line of data for the variables in any branch or limb. In these cases, the values of *y* and *z* must be repeated for each alternative in the branch or limb.

The following model and setup illustrate this for a three level model: (all in trunk 1)

			*x*1	*x*2	*y*1	*y*2	*z*1	*z*2
limb 1	branch 1|1	twig 1|1,1	.6	1	3	.02	104	.9
		twig 2|1,1	.1	2	3	.02	104	.9
	branch 2|1	twig 1|2,1	.8	2	7	.15	104	.9
		twig 2|2,1	.2	3	7	.15	104	.9
limb 2	branch 1|2	twig 1|1,2	.9	6	11	.08	96	.4
		twig 2|1,2	.3	1	11	.08	96	.4
		twig 3|1,2	.4	0	11	.08	96	.4

9.3.2 Tree Definition

The model command for estimating nested logit models is exactly as described in Chapter 8 for single level models, where the model name is now the generic **NLOGIT**;

NLOGIT ; Lhs = ... ; Choices = ... definition of choice set
; ... definition of utility functions for alternatives

All of the options described earlier are available. The nested logit model is requested by adding

; Tree = ... definition of the tree structure

to the command. In order to specify the tree, use these conventions:

{ } specifies a trunk,
[ ] specifies a limb within a trunk,
( ) specifies a branch within a limb in a trunk.

Entries in a list are separated by commas. Names for trunks, limbs and branches are optional before the opening '{' or '[' or '('. If you elect not to provide names, the defaults chosen will be Trunk{*l*}, Lmb[*i*|*l*] and Br(*j*|*i*,*l*) respectively, where the numbering is developed reading from left to right in your tree definition. Alternative names appear inside the parentheses. Some examples are as follows:

One limb:

; Tree = travel [fly(air), ground(train,bus,car)]

One limb: (With one limb, the [ ] is optional.)

; Tree = fly(air), ground(train,bus,car)

One limb: (Branch names are optional. These would be Limb[1], Br(1|1) and Br(2|1).)

; Tree = (air), (train,bus,car)

One limb, one branch, no nesting: (This would be unnecessary and could be omitted.)

; Tree = (air,train,bus,car)

Nested logit model – two limbs, one with one branch:

```
; Tree =  private [fly(air), ground(car_pas, car_drv)],
          public  [(train,bus)]
```

The fully nested 2×2×2×2 model shown in Section 9.1 could be specified with

```
; Choices = a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16
; Tree =  Trunk1 { limb1 [branch1 (a1, a2),     branch2 (a3, a4)    ],
                   limb2 [branch3 (a5, a6),     branch4 (a7, a8)    ] },
        Trunk2 { limb3 [branch5 (a9, a10),   branch6 (a11, a12)],
                   limb4 [branch7 (a13, a14), branch8 (a15, a16)] }
```

9.3.3 Utility Functions

You may define the utility functions exactly as described in Chapter 3 for one level models. You may also define utility functions for branches and limbs and trunks, but note that in order to do so, you must use the explicit form. These are specified exactly the same as those for elemental alternatives. For example, in a two level model, you might put demographic characteristics, such as income or family size, at the top level. A complete model might appear as follows:

```
NLOGIT      ; Lhs = mode ; Choices = air,train,bus,car
            ; Tree  = travel [public(bus,train), private(air,car)]
            ; Model:  U(air)      = ba +  bcost * gc + btime * ttme   /
                      U(train)    = bt  +  bcost * gc + btime * ttme   /
                      U(car)      = bc +  bcost * gc + btime * ttme   /
                      U(bus)      =       bcost * gc + btime * ttme   /
                      U(public)   = ap +  apub * hinc /
                      U(private)  =       aprv  * hinc $
```

This model can be considerably collapsed;

```
; Model:  U(air,train,bus,car) = <ba,bc,0,bt> +
                                   bcost * gc + btime * ttme /
          U(public,private)    = <ap,0> +
                                 <apub, aprv> * income $
```

Note that the same function specification U(...) is used for all three kinds of equations, for alternatives, branches, and limbs.

Finally, as noted earlier, you may impose equality constraints at any points in the model, just by using the same parameter name where you want the equality imposed. For example, if, for some reason, you desired to force the parameters *apub* and *bcost* to be equal, you could just change *apub* to *bcost* in the utility equation for *public*. That is, you can, if you wish, force equality of parameters at different levels of a model, once again, just by using the same parameter name in the model specification. (Given the impact of the scale parameters, this is probably inadvisable, but the program will allow you to do it nonetheless.)

The interaction of alternative specific constants, and branch and limb specific constants is complex, and it is difficult to draw generalities. As a general rule, models will usually become overdetermined, resulting in a singular Hessian, when there are more than NALT-1 constants, of all three types, in the entire model. Likewise, interactions of attributes and choice specific dummy variables can produce this effect as well. Users who encounter problems in which *NLOGIT* claims either that it is impossible to maximize the log likelihood function, or there is a singular Hessian, should examine the model for this pitfall.

9.3.4 Setting and Constraining Inclusive Value Parameters

There is an inclusive value parameter for each limb, branch, and trunk in the model. For example, in the tree

; Choices = air,train,bus,car
; Tree = travel [public(bus,train), private(air,car)]

with the other parameters, we estimate $\tau_{public|travel}$, $\tau_{private|travel}$, σ_{travel}. Since there is only one limb, *travel*, σ_{travel} = 1.0. The other two parameters are free and unrestricted. You can modify the specification of these parameters in two ways:

- You may specify that they are equal to each other.
- You may specify that they are fixed values instead of free parameters to estimate.

To use these features, add the specification

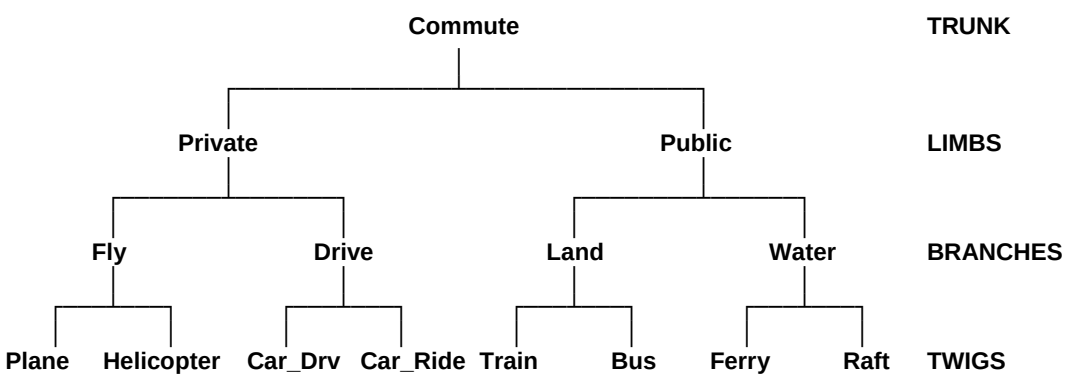
; Ivset: ... specification

Note, once again, the presence of a colon in this specification. For purposes of this specification, τs, σs, and ϕs are treated the same. To force parameters to be equal, put the names of the branches and/or limbs together in parentheses in the **; Ivset:** specification.

For the example given above, to force the two τs to be equal in the estimated model, use

; Ivset: (public,private)

For a second example, consider this larger tree:



We would define this with

; Tree = private [fly(plane,helicptr), drive(car_ride,car_drv)],
public [land(train,bus), water(ferry,raft)]

There are six IV parameters, $\tau_{i|l}$ for each of *fly*, *drive*, *land*, and *water*, and σ_l for *private* and *public*. If it were desired to force $\sigma_{private} = \sigma_{public}$, $\tau_{fly|private} = \tau_{land|public}$, and $\tau_{water|public}$ (for some reason) to equal σ_{public}, you could use

; Ivset: (private,public,water) / (fly,land)

Note, once again, separate specifications are separated by slashes. Also, there is no problem using this device to force IV parameters at one level to equal those at another. Thus, '(**private,public,water**)' forces σ_{public} to equal $\tau_{water|public}$ and $\sigma_{private}$.

In addition to the preceding, you may fix inclusive value parameters. The setup is the same as above with the additional specification of the value in square brackets. I.e.,

; Ivset: (...) = [the value]

The list in parentheses may contain a single name, so as to fix a particular coefficient at a given value. You might have

; Ivset: (private,public) / (fly,ground) = [.75] / (land) = [.95] $

You will see a diagnostic message if you attempt to modify an inclusive value parameter that is fixed at 1.0 for identification purposes. For example, this specification of a two level model:

; Tree = travel [public(bus,train), private(air,car)]
; Ivset: (travel) = [.75]

generates an error message, since $\sigma_{travel} = 1.0$ (one limb). Note, also, that fixed IV parameters are off limits to equality constraints, as well. Thus, for this example, the specification

; Ivset: (travel,public)

also generates an error.

```
Error:  1093: You have given a spec for an IV parm that is fixed at 1.
```

You may not change the specification of ϕ_{travel}.

In the output of the estimation procedure, inclusive value parameters are denoted by the name of the branch or limb to which they are attached (or the default names given earlier).

9.3.5 Command Builder

The command builders can be used to specify the nested logit models. Select Model:Discrete Choice/Nested Logit to access the command builder. The choice variable is defined on the Main page and the rest of the model may be specified on the Options page. See Figure 9.1.

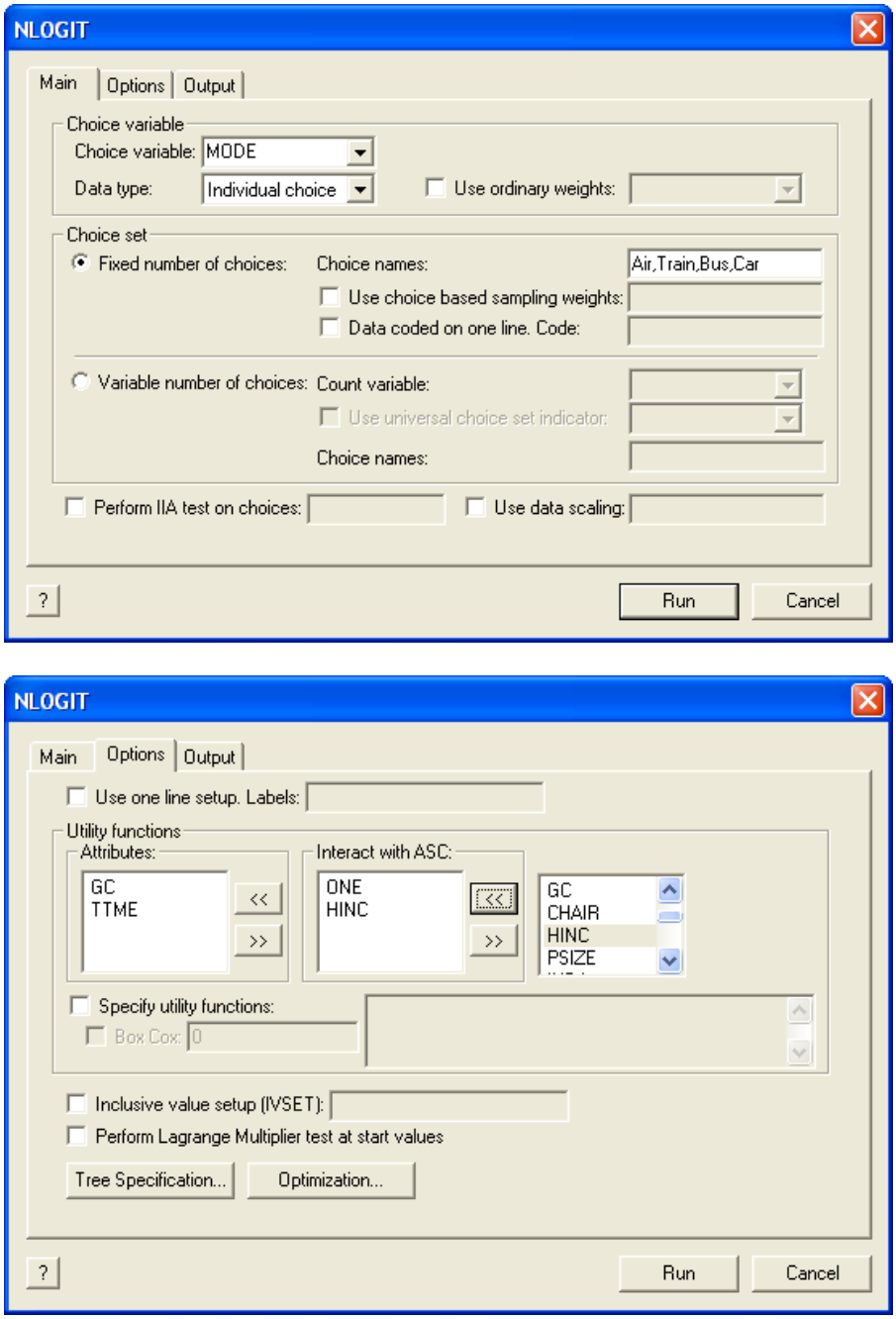


Figure 9.1 Command Builder for Nested Logit Models

The tree is specified in a subsidiary dialog box by selecting Tree Specification at the bottom of the Options page. The dialog box, shown in Figure 9.2, allows you to define the tree graphically. Note in the dialog shown, *public* and *private* are siblings while *bus* is a child node of *public*.

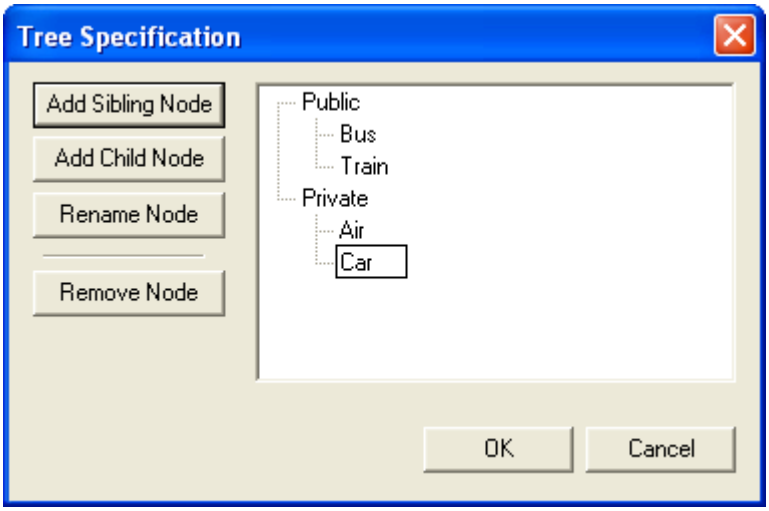


Figure 9.2 Tree Specification Dialog Box for Defining the Tree Structure

The remaining options for output and results to be saved are defined in the Output page as shown in Figure 9.3.

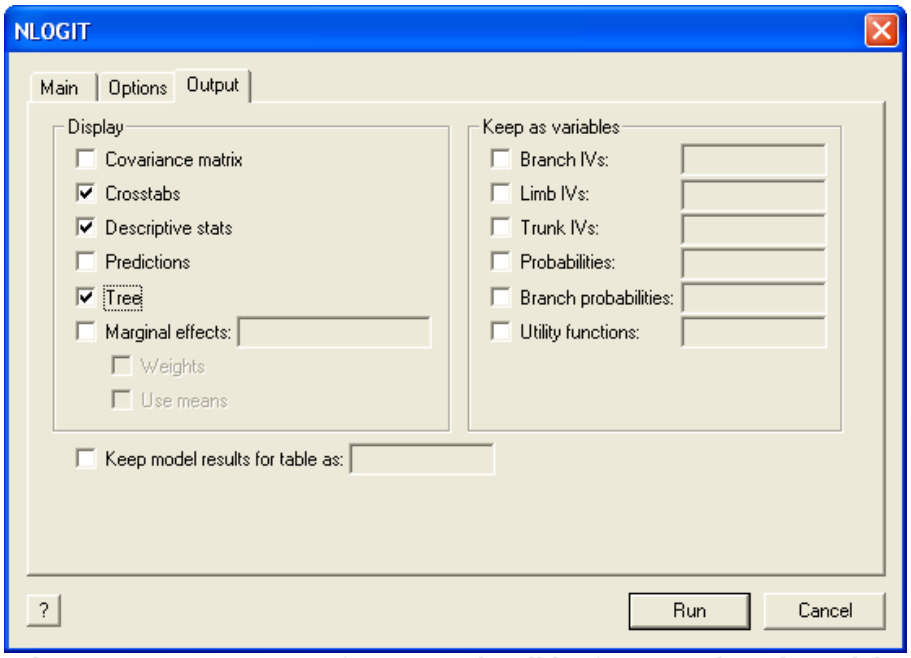


Figure 9.3 Output Page of Command Builder for Nested Logit Models

9.4 Marginal Effects and Elasticities

In the nested logit model with $P(j,b,l,r) = P(j|b,l,r) \times P(b|l,r) \times P(l|r) \times P(r)$, the marginal effect of a change in attribute k in the utility function for alternative J in branch B of limb L of trunk R on the probability of choice j in branch b of limb l of trunk r is computed using the following result: Lower case letters indicate the twig, branch, limb and trunk of the outcome upon which the effect is being exerted. Upper case letters indicate the twig, branch, limb and trunk which contain the outcome whose attribute is being changed:

$$\frac{\partial \log P(alt = j, limb = l, branch = b, trunk = r)}{\partial x(k) \mid alt = J, limb = L, branch = B, trunk = R)} = D(k \mid J, B, L, R) = \Delta(k) \times F \ ,$$

where $\Delta(k)$ = coefficient on $x(k)$ in $U(J|B,L,R)$

and

$$\begin{aligned}
F \ = \ & \mathbf{1}(r{=}R) \times \mathbf{1}(l{=}L) \times \mathbf{1}(b{=}B) \times [\mathbf{1}(j{=}J) - P(J|BLR)] && \text{(trunk effect)},\\
& \mathbf{1}(r{=}R) \times \mathbf{1}(l{=}L) \times [\mathbf{1}(b{=}B) - P(B|LR)] \times P(J|BLR) \times \tau_{B|LR} && \text{(limb effect)},\\
& \mathbf{1}(r{=}R) \times [\mathbf{1}(l{=}L) - P(L|R)] \times P(B|LR) \times P(J|BLR) \times \tau_{B|LR} \times \sigma_{L|R} && \text{(branch effect)},\\
& [\mathbf{1}(r{=}R) - P(R)] \times P(L|R) \times P(B|LR) \times P(J|BLR) \times \tau_{B|LR} \times \sigma_{L|R} \times \phi_R && \text{(twig effect)}.
\end{aligned}$$

(Note, in this expression, J, B, L and R are being used generically to indicate a particular choice, branch, limb and trunk, not the total numbers of twigs, branches, limbs and trunks.) The marginal effect is

$$\partial\, P(j,b,l,r)/\partial x(k)|J,B,L,R \ = \ P(j,b,l,r)\ \Delta(k)\ F.$$

A marginal effect has four components, an effect on the probability of the particular trunk, one on the probability for the limb, one for the branch, and one for the probability for the twig. (Note that with one trunk, $P(l) = P(1) = 1$, and likewise for limbs and branches.) For continuous variables, such as cost, you might be interested, instead, in the

$$Elasticity = x(k)|J,B,L,R \times \Delta(k|J,B,L,R) \times F.$$

NLOGIT will provide either. As in the case of nonnested models, marginal effects are requested with

; Effects: attribute [list of outcomes] / ...

or

; Effects: attribute (list) / ... for elasticities

This generates a table of results for each of the outcomes listed. For example,

```
NLOGIT      ; Lhs = mode
            ; Choices = air,train,bus,car
            ; Tree = travel [public(bus,train), private(air,car)]
            ; Model:  U(air)   = ba +  bcost * gc + btime * ttme /
                      U(train) = bc +  bcost * gc + btime * ttme /
                      U(bus)   =       bcost * gc + btime * ttme /
                      U(car)   = bc +  bcost * gc
            ; Effects: gc(car) $
```

This lists the effects on all four probabilities of changes in attribute generalized cost (*gc*) of choice *car*.

```
+-----------------------------------------------------------+
| Partial effects = average over observations               |
|                                                           |
| dlnP[alt=j,br=b,lmb=l,tr=r]                               |
| ---------------------------- = D(k:J,B,L,R) = delta(k)*F  |
| dx(k):alt=J,br=B,lmb=L,tr=R]                              |
| delta(k) = coefficient on x(k) in U(J|B,L,R)              |
| F = (r=R)  (l=L) (b=B) [(j=J)-P(J|BLR)]                   |
|  +  (r=R)  (l=L) [(b=B) -P(B|LR)]P(J|BLR)t(B|LR)           |
|  +  (r=R) [(l=L)-P(L|R)] P(B|LR) P(J|BLR)t(B|LR)s(L|R)     |
|  + [(r=R) -P(R)] P(L|R)  P(B|IR) P(J|BIR)t(B|LR)s(L|R)f(R) |
|                                                           |
| P(J|BLR)=Prob[choice=J |branch=B,limb=L,trunk=R]          |
| P(B|LR), P(L|R), P(R) defined likewise.                   |
| (n=N) = 1 if n=N, 0 else, for n=j,b,l,r and N=J,B,L,R.    |
| Elasticity = x(k) * D(j|B,L,R)                            |
| Marginal effect = P(JBLR)*D = P(J|BLR)P(B|LR)P(L|R)P(R)D  |
| F is decomposed into the 4 parts in the tables.           |
+-----------------------------------------------------------+
+-----------------------------------------------------------------------+
| Elasticity            averaged over observations.                     |
| Attribute is GC       in choice CAR                                   |
| Effects on probabilities of all choices in the model:                 |
| * indicates direct Elasticity effect of the attribute.                |
|                        Decomposition of Effect if Nest    Total Effect|
|                        Trunk   Limb   Branch   Choice     Mean  St.Dev|
| Trunk=Trunk{1}                                                        |
| Limb=TRAVEL                                                           |
|    Branch=PUBLIC                                                      |
|       Choice=BUS        .000   .000    .857    .000       .857   .532 |
|       Choice=TRAIN      .000   .000    .857    .000       .857   .532 |
|    Branch=PRIVATE                                                     |
|       Choice=AIR        .000   .000  -1.015    .571      -.444   .746 |
| *     Choice=CAR        .000   .000  -1.015   -.338     -1.353  1.059 |
+-----------------------------------------------------------------------+
```

Note that across a row, the effects sum to the total effect given. The default method of computing the elasticities is to average the observation specific results. The results show the mean and the sample standard deviations. If you use the **; Means** specification, then the elasticities are computed once, and the results reflect the change, as shown below. (The differences are noticeably large.)

```
+-----------------------------------------------------------------------+
| Elasticity            computed at sample means.                       |
| Attribute is GC       in choice CAR                                   |
| Effects on probabilities of all choices in the model:                 |
| * indicates direct Elasticity effect of the attribute.                |
|                        Decomposition of Effect if Nest    Total Effect|
|                        Trunk   Limb   Branch   Choice     Mean  St.Dev|
| Trunk=Trunk{1}                                                        |
| Limb=TRAVEL                                                           |
|    Branch=PUBLIC                                                      |
|       Choice=BUS        .000   .000    .584    .000       .584   .000 |
|       Choice=TRAIN      .000   .000    .584    .000       .584   .000 |
|    Branch=PRIVATE                                                     |
|       Choice=AIR        .000   .000   -.411    .303      -.107   .000 |
| *     Choice=CAR        .000   .000   -.411   -.605     -1.016   .000 |
+-----------------------------------------------------------------------+
```

9.5 Inclusive Values, Utilities, and Probabilities

You can request a listing of the actual outcomes and predicted probabilities with

; List

For large nested logit models, the listing would be extremely cumbersome, so a list can only be produced for models with seven or fewer elemental alternatives. You can also keep as variables the fitted probabilities and the branch, limb, and trunk inclusive values. The predicted probabilities are $P(j,b,l,r)$. The inclusive values for the branches are repeated for each choice (row of data) within the branches. The inclusive values for the limbs are, likewise, repeated for every alternative in the limb and similarly for trunks. An example appears below. The command specifications are:

; Prob = name to retain predicted probabilities as a variable
; Ivb = name to retain the branch level inclusive values as a variable
; Ivl = name to retain the limb level inclusive values as a variable
; Ivt = name to retain the trunk level inclusive values as a variable

Normally, in this setting, the unconditional probability, $P(j,b,l,r)$, is the one of interest. However, for some purpose, you might want, instead, the conditional probabilities at the twig level, $P(j,b,l,r)$. You can request to have this retained as a variable with

; Cprob = name to retain estimated conditional probabilities

Lastly, the utility values at the twig level of the tree are

$$U(j|b,l,r) = \boldsymbol{\beta}'\mathbf{x}_{j|b,l,r}\,.$$

These are the values that you define in your **; Model: ...** specification. You may request to retain these for later use with

; Utility = name of the variable

If you have not defined a utility function for an alternative, the value returned for that row of data is 0.0, not missing (-999). Utility values may be further processed like any other variable. You may find them useful, for example, for computing inclusive values in another model.

An example of the use of these features is shown in the next section.

9.6 Application of a Nested Logit Model

The following estimates a two level model. The tree has a 'degenerate' branch; the *air* branch has only a single alternative, *fly*. It also uses most of the optional features mentioned above.

```
NLOGIT      ; Lhs = mode
            ; Start = logit
            ; Choices = air,train,bus,car
            ; Tree = travel[fly(air), ground(train,bus,car)]
            ; Model: U(air,train,bus,car)  = bt*tasc+bb*basc+bg*gc+at*ttme /
                     U(fly,ground)         = aa*aasc+ah*hinca
            ; Describe
            ; Effects: gc[car] ; Pwt
            ; List
            ; Ivb = branchiv
            ; Ivl  = limbiv
            ; Utility = u_choice
            ; Prob = pkji
            ; Cprob = pk_ji $
```

Starting values for the iterations are obtained by a one level multinomial logit model. The MNL also reports results of estimation of the branch choice model. These are the (inconsistent) estimates of α in the branch choice model.

```
+---------------------------------------------+
| Discrete choice and multinomial logit models|
+---------------------------------------------+
+---------------------------------------------+
| Start values obtained using MNL model       |
| Maximum Likelihood Estimates                |
| Dependent variable               Choice     |
| Weighting variable                 None     |
| Number of observations              210     |
| Iterations completed                  5     |
| Log likelihood function       -378.5920     |
| Number of parameters                  6     |
| Info. Criterion: AIC =          3.66278     |
|   Finite Sample: AIC =          3.66475     |
| Info. Criterion: BIC =          3.75841     |
| Info. Criterion:HQIC =          3.70144     |
| Log-L for Choice   model =    -260.1975     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -283.7588  .08303  .07124 |
| Log-L for Branch   model =    -118.3945     |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
```

```
+---------------------------------------------+
| Notes No coefficients=> P(i,j)=1/J(i).      |
|       Constants only => P(i,j) uses ASCs    |
|         only. N(j)/N if fixed choice set.   |
|         N(j) = total sample frequency for j |
|         N    = total sample frequency.      |
|       These 2 models are simple MNL models. |
|       R-sqrd = 1 - LogL(model)/logL(other)  |
|       RsqAdj=1-[nJ/(nJ-nparm)]*(1-R-sqrd)   |
|         nJ   = sum over i, choice set sizes |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
---------+Model for Choice Among Alternatives
 BT      |     .77778700       .20792992     3.741   .0002
 BB      |    -.13076048       .22872416     -.572   .5675
 BG      |    -.01773795       .00405470    -4.375   .0000
 AT      |    -.01340138       .00317904    -4.216   .0000
---------+Model for Choice Among Branches
 AA      |   -1.92254215       .35420335    -5.428   .0000
 AH      |     .02612091       .00817431     3.195   .0014
```

The MNL estimates are followed by the nested logit estimates.

```
Normal exit from iterations. Exit status=0.
+---------------------------------------------+
| FIML Nested Multinomial Logit Model         |
| Dependent variable                 MODE     |
| Number of observations              210     |
| Log likelihood function       -193.6561     |
| Number of parameters                  8     |
| Info. Criterion: AIC =          1.92053     |
|   Finite Sample: AIC =          1.92395     |
| Info. Criterion: BIC =          2.04804     |
| Info. Criterion:HQIC =          1.97208     |
| Restricted log likelihood     -312.5500     |
| McFadden Pseudo R-squared      .3803994     |
| Chi squared                    237.7877     |
| Degrees of freedom                    8     |
| Prob[ChiSqd > value] =         .0000000     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| No coefficients   -312.5500  .38040  .37243 |
| Constants only    -283.7588  .31753  .30875 |
| At start values   -287.6816  .32684  .31818 |
| Response data are given as ind. choice.     |
+---------------------------------------------+
+---------------------------------------------+
| Notes No coefficients=> P(i,j)=1/J(i).      |
|       Constants only => P(i,j) uses ASCs    |
|         only. N(j)/N if fixed choice set.   |
|         N(j) = total sample frequency for j |
|         N    = total sample frequency.      |
|       These 2 models are simple MNL models. |
|       R-sqrd = 1 - LogL(model)/logL(other)  |
|       RsqAdj=1-[nJ/(nJ-nparm)]*(1-R-sqrd)   |
|         nJ   = sum over i, choice set sizes |
+---------------------------------------------+
```

```
+---------------------------------------------+
| FIML Nested Multinomial Logit Model         |
| The model has 2 levels.                     |
| Nested Logit form:IV parms = taub|l,r,sl|r  |
| and fr. No normalizations imposed a priori. |
| p(alt=j|b=B,l=L,r=R)=exp[bX_j|BLR]/Sum      |
| p(b=B|l=L,r=R)=exp[aY_B|LR+tauB|LRIVB|LR)]/ |
| Sum. p(l=L|r=R)=exp[cZ_L|R+sL|RIVL|R)]/Sum  |
| p(r=R)=exp[qH_R+fRIVR]/Sum...               |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
---------+Attributes in the Utility Functions (beta)
 BT      |    5.06460277       .66202159     7.650   .0000
 BB      |    4.09631480       .61515554     6.659   .0000
 BG      |    -.03158748       .00815636    -3.873   .0001
 AT      |    -.11261749       .01412912    -7.971   .0000
---------+Attributes of Branch Choice Equations (alpha)
 AA      |    3.54086522      1.20812715     2.931   .0034
 AH      |     .01533132       .00938134     1.634   .1022
---------+IV parameters, tau(j|i,l),sigma(i|l),phi(l)
 FLY     |     .58600939       .14062118     4.167   .0000
 GROUND  |     .38896192       .12366583     3.145   .0017

+-----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative AIR                :
|     Utility Function          |                     |     58.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose AIR      |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| -------------------  -------- | -------------------+------------------- |
| BT           5.0646  TASC     |     .000       .000|     .000      .000 |
| BB           4.0963  BASC     |     .000       .000|     .000      .000 |
| BG           -.0316  GC       |  102.648     30.575|  113.552    33.198 |
| AT           -.1126  TTME     |   61.010     15.719|   46.534    24.389 |
+-----------------------------------------------------------------------+
+-----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative TRAIN              :
|     Utility Function          |                     |     63.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose TRAIN    |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| -------------------  -------- | -------------------+------------------- |
| BT           5.0646  TASC     |    1.000       .000|    1.000      .000 |
| BB           4.0963  BASC     |     .000       .000|     .000      .000 |
| BG           -.0316  GC       |  130.200     58.235|  106.619    49.601 |
| AT           -.1126  TTME     |   35.690     12.279|   28.524    19.354 |
+-----------------------------------------------------------------------+
+-----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative BUS                :
|     Utility Function          |                     |     30.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose BUS      |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| -------------------  -------- | -------------------+------------------- |
| BT           5.0646  TASC     |     .000       .000|     .000      .000 |
| BB           4.0963  BASC     |    1.000       .000|    1.000      .000 |
| BG           -.0316  GC       |  115.257     44.934|  108.133    43.244 |
| AT           -.1126  TTME     |   41.657     12.077|   25.200    14.919 |
+-----------------------------------------------------------------------+
```

```
+----------------------------------------------------------------------+
|             Descriptive Statistics for Alternative CAR                :
|     Utility Function          |                    |     59.0 observs.  |
|     Coefficient               | All      210.0 obs.|that chose CAR      |
| Name          Value  Variable | Mean      Std. Dev.|Mean      Std. Dev. |
| -------------------  -------- | -------------------+------------------- |
| BT           5.0646  TASC     |     .000       .000|     .000      .000 |
| BB           4.0963  BASC     |     .000       .000|     .000      .000 |
| BG           -.0316  GC       |   95.414     46.827|   89.085    49.833 |
| AT           -.1126  TTME     |     .000       .000|     .000      .000 |
+----------------------------------------------------------------------+
PREDICTED PROBABILITIES (* marks actual, + marks prediction.)
Indiv    AIR       TRAIN     BUS       CAR
    1   .1515     .3518     .1232     .3734*+
    2   .2676     .1949     .0260     .5114*+
    3   .1563     .1040     .1509     .5888*+
    4   .3998     .1180     .0153     .4669*+
    5   .3418     .3510 +   .0469     .2603*
    6   .1323     .3423*+   .2212     .3043
    7   .4186*+   .0815     .1182     .3817
    8   .0955     .4956 +   .1848     .2241*
    9   .1685     .3915 +   .1371     .3030*
   10   .2484     .3203 +   .1122     .3191*
```

(Observations 11 - 210 are omitted.)

```
+-------------------------------------------------------------+
| Partial effects = average over observations                 |
|                                                             |
| dlnP[alt=j,br=b,lmb=l,tr=r]                                 |
| ---------------------------- = D(k:J,B,L,R) = delta(k)*F    |
| dx(k):alt=J,br=B,lmb=L,tr=R]                                |
|                                                             |
| delta(k) = coefficient on x(k) in U(J|B,L,R)                |
| F = (r=R)  (l=L) (b=B) [(j=J)-P(J|BLR)]                     |
|  +  (r=R)  (l=L) [(b=B) -P(B|LR)]P(J|BLR)t(B|LR)            |
|  +  (r=R) [(l=L)-P(L|R)] P(B|LR) P(J|BLR)t(B|LR)s(L|R)      |
|  + [(r=R) -P(R)] P(L|R)  P(B|IR) P(J|BIR)t(B|LR)s(L|R)f(R)  |
|                                                             |
| P(J|BLR)=Prob[choice=J |branch=B,limb=L,trunk=R]            |
| P(B|LR), P(L|R), P(R) defined likewise.                     |
| (n=N) = 1 if n=N, 0 else, for n=j,b,l,r and N=J,B,L,R.      |
| Elasticity = x(k) * D(j|B,L,R)                              |
| Marginal effect = P(JBLR)*D = P(J|BLR)P(B|LR)P(L|R)P(R)D    |
| F is decomposed into the 4 parts in the tables.             |
+-------------------------------------------------------------+
```

(Note, the within branch cross elasticities are not equal, as would be imposed by the IID assumptions, because we used **; Pwt** to weight the observations.)

```
+----------------------------------------------------------------------+
| Derivative (times 100) averaged over observations.                   |
| Attribute is GC       in choice CAR                                  |
| Effects on probabilities of all choices in the model:                |
| * indicates direct Derivative effect of the attribute.               |
|                        Decomposition of Effect if Nest    Total Effect|
|                        Trunk   Limb   Branch   Choice     Mean  St.Dev|
| Limb=TRAVEL                                                          |
|    Branch=FLY                                                        |
|       Choice=AIR        .000   .000    .119    .000       .119   .083 |
|    Branch=GROUND                                                     |
|       Choice=TRAIN      .000   .000   -.017    .235       .218   .147 |
|       Choice=BUS        .000   .000   -.014    .223       .209   .130 |
| *     Choice=CAR        .000   .000   -.105   -.399      -.504   .140 |
+----------------------------------------------------------------------+
```

Chapter 10: The Random Parameters Logit Model

10.1 Introduction

The random parameters logit (RPL) model, also referred to as the mixed logit model, is the most general model form in *NLOGIT* in terms of the variety of model specifications it can accommodate and in terms of the range of behavior that it can model. This chapter will develop the numerous different specifications of the model that can be accommodated.

NLOGIT offers an extensive set of specifications within the mixed logit structure. This model is gaining great popularity in applications. Capabilities provided by the estimator include (i) choosing from among a large number of analytical distributions for each random parameter, (ii) accounting for the non-independence between observations associated with the same respondent (a theme of importance in stated choice studies), (iii) decomposing the mean and standard deviation of one or more random parameters to reveal sources of systematic taste heterogeneity, (iv) accounting for correlation of random parameters, (v) imposing priors based on known choices in model estimation, (vi) imposing constraints on distributions (e.g. constraining the triangular or normal to ensure that it does not change sign over its range), (vii) selecting subsets of pre-specified variables to interact with the mean and standard deviation of random parameterized attributes, and (viii) deriving willingness to pay estimates when both the numerator and denominator are random parameter estimates.

10.2 Random Parameters (Mixed) Logit Models

This model is somewhat similar to the random coefficients model for linear regressions. The model formulation is a one level multinomial logit model, for individuals $i = 1,...,N$ in choice setting t. Neglecting for the moment the error components aspect of the model, we begin with the basic form of the multinomial logit model, with (optional) alternative specific constants α_{ji} and attributes $\mathbf{x}_{ji}$,

$$\text{Prob}(y_{it} = j) \;=\; \frac{\exp\left(\alpha_{ji} + \boldsymbol{\beta}_i'\mathbf{x}_{ji}\right)}{\sum_{q=1}^{J_i}\exp\left(\alpha_{qi} + \boldsymbol{\beta}_i'\mathbf{x}_{qi}\right)}.$$

The RPL model emerges as the form of the individual specific parameter vector, $\boldsymbol{\beta}_i$ is developed. The most familiar, simplest version of the model specifies

$$\beta_{ki} \;=\; \beta_k \;+\; \sigma_k v_{ik},$$

and

$$\alpha_{ji} \;=\; \alpha_j \;+\; \sigma_j v_{ji},$$

where β_k is the population mean, v_{ik} is the individual specific heterogeneity, with mean zero and standard deviation one, and σ_k is the standard deviation of the distribution of β_{ik}s around β_k. The term 'mixed logit' is often used in the literature for this model. The choice specific constants, α_{ji} and the elements of $\boldsymbol{\beta}_i$ are distributed randomly across individuals with fixed means. A refinement of the

model is to allow the means of the parameter distributions to be heterogeneous with observed data, $\mathbf{z}_i$, (which does not include *one*). This would be a set of choice invariant characteristics that produce individual heterogeneity in the means of the randomly distributed coefficients so that

$$\beta_{ki} = \beta_k + \boldsymbol{\delta}_k'\mathbf{z}_i + \sigma_k v_{ki},$$

and likewise for the constants. The model is not limited to the normal distribution. We consider several alternatives below. One important variation is the lognormal model,

$$\beta_{ki} = \exp(\rho_k + \boldsymbol{\delta}_k'\mathbf{z}_i + \sigma_k v_{ki}).$$

The v_{jki}s are individual and choice specific, unobserved random disturbances – the source of the heterogeneity. Thus, as stated above, in the population, if the random terms are normally distributed,

$$\alpha_{ji} \text{ or } \beta_{ki} \sim \text{Normal or Lognormal } [\rho_{j\ or\ k} + \boldsymbol{\delta}_{j\ or\ k}'\mathbf{z}_i, \sigma_{j\ or\ k}^2].$$

(Other distributions may be specified.) For the full vector of K random coefficients in the model, we may write the full set of random parameters as

$$\boldsymbol{\rho}_i = \boldsymbol{\rho} + \boldsymbol{\Delta}\mathbf{z}_i + \boldsymbol{\Gamma}\mathbf{v}_i.$$

where $\boldsymbol{\Gamma}$ is a diagonal matrix which contains σ_k on its diagonal. For convenience at this point, we will simply gather the parameters, choice specific or not, under the subscript 'k.' (The notation is a bit more cumbersome for the lognormally distributed parameters. We will return to that in the technical details.)

We can go a step further and allow the random parameters to be correlated. All that is needed to obtain this additional generality is to allow $\boldsymbol{\Gamma}$ to be a triangular matrix with nonzero elements below the main diagonal. Then, the full covariance matrix of the random coefficients is $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Gamma}'$. The standard case of uncorrelated coefficients has $\boldsymbol{\Gamma} = \text{diag}(\sigma_1,\sigma_2,\ldots,\sigma_k)$. If the coefficients are freely correlated, $\boldsymbol{\Gamma}$ is a full, unrestricted, lower triangular matrix and $\boldsymbol{\Sigma}$ will have nonzero off diagonal elements. (It will be convenient to aggregate this one step further. We may gather the entire parameter vector for the model in this formulation simply by specifying that for the nonrandom parameters in the model, the corresponding rows in $\boldsymbol{\Delta}$ and $\boldsymbol{\Gamma}$ are zero.) We will also define the data and parameter vector so that any choice specific aspects are handled by appropriate placements of zeros in the applicable parameter vector.

An additional extension of the model allows the distribution of the random parameters to be heteroscedastic. As stated above, the variance of v_{ik} is taken to be a constant. The model is made heteroscedastic by assuming, instead, that

$$\text{Var}[v_{ik}] = \sigma_{jk}^2\,[\exp(\boldsymbol{\omega}_k'\mathbf{hr}_i)]^2.$$

A convenient way to parameterize this is to write the full model as

$$\boldsymbol{\rho}_i = \boldsymbol{\rho} + \boldsymbol{\Delta}\mathbf{z}_i + \boldsymbol{\Gamma}\boldsymbol{\Omega}_i\mathbf{v}_i$$

where $\boldsymbol{\Omega}_i$ is the diagonal matrix of individual specific variance terms; $\omega_{ik} = \exp(\boldsymbol{\omega}_k'\mathbf{hr}_i)$.

The list of variations above produces an extremely flexible, general model. Typically, you would use only some of them, though in principle, all could appear in the model at once. We will develop them in parts in the sections to follow. A convenient form of the full random parameters logit model to begin with is

$$\text{Prob}(y_{it} = j) \;=\; \frac{\exp(\alpha_{ji} + \boldsymbol{\beta}_i'\mathbf{x}_{jit})}{\sum_{q=1}^{J_{it}} \exp(\alpha_{qi} + \boldsymbol{\beta}_i'\mathbf{x}_{qit})},$$

Finally, an additional layer of individual heterogeneity may be added to the model in the form of the error components detailed below. The full model with all components is

$$\text{Prob}(y_{it} = j) \;=\; \frac{\exp\left[\alpha_{ji} + \boldsymbol{\beta}_i'\mathbf{x}_{jit} + \boldsymbol{\Sigma}_{m=1}^{M} d_{jm}\theta_m \exp(\boldsymbol{\gamma}_m'\mathbf{he}_i)E_{im}\right]}{\sum_{q=1}^{J_i} \exp\left[\alpha_{qi} + \boldsymbol{\beta}_i'\mathbf{x}_{qit} + \boldsymbol{\Sigma}_{m=1}^{M} d_{qm}\theta_m \exp(\boldsymbol{\gamma}_m'\mathbf{he}_i)E_{im}\right]},$$

where the components of the model are as follows:

Random Alternative Specific Constants and Taste Parameters:

$(\alpha_{ji}, \boldsymbol{\beta}_i) = (\alpha_j, \boldsymbol{\beta}) + \boldsymbol{\Delta}\mathbf{z}_i + \boldsymbol{\Gamma}\boldsymbol{\Omega}_i\mathbf{v}_i$, $\boldsymbol{\Omega}_i =$ diag(ω_{i1}, ω_{i2}, ...) or $\boldsymbol{\Omega}_i$ = diag(σ_1,...,σ_k)

$\boldsymbol{\beta}, \alpha_{ji}$ = constant terms in the distributions of the random taste parameters

Uncorrelated Parameters with Homogeneous Means and Variances

β_{ik} = $\beta_k + \sigma_k v_{ik}$ when $\boldsymbol{\Delta} = \mathbf{0}$, $\boldsymbol{\Gamma} = \mathbf{I}$, $\boldsymbol{\Omega}_i$ = diag(σ_1,...,σ_k)

$\mathbf{x}_{jit}$ = all observed choice attributes and individual characteristics

$\mathbf{v}_i$ = random unobserved taste variation, with mean vector $\mathbf{0}$ and covariance matrix $\mathbf{I}$

Uncorrelated Parameters with Heterogeneous Means and Variances

β_{ik} = $\beta_k + \boldsymbol{\delta}_k'\mathbf{z}_i + \sigma_k \exp(\boldsymbol{\omega}_k'\mathbf{hr}_i)v_{ik}$ when $\boldsymbol{\Gamma} = \mathbf{I}$, $\boldsymbol{\Omega}_i$ = diag(ω_{i1}, ω_{i2}, ...)

$\boldsymbol{\Delta}$ = parameters that enter the heterogeneous means of the distributions of the random parameters; $\boldsymbol{\beta} + \boldsymbol{\Delta}\mathbf{z}_i$ = the heterogeneous means

ω_{ik} = $\exp(\boldsymbol{\omega}_k'\mathbf{hr}_i)$ = heterogeneity in the variances of the distributions of the random parameters

$\boldsymbol{\omega}_k$ = parameters in the variance heterogeneity of the random parameters

σ_{ik} = $\sigma_k\omega_{ik}$ = heterogeneous standard deviations in the distributions of the random parameters; $\sigma_{ik} = \sigma_k$ in a homoscedastic model

$\mathbf{z}_i$ = observed variables that measure the heterogeneity in the means of the random parameters

$\mathbf{hr}_i$ = observed variables that measure the heterogeneity in the variances of the random parameters

Correlated Parameters with Heterogeneous Means

β_{ik} = $\beta_k + \boldsymbol{\delta}_k'\mathbf{z}_i + \boldsymbol{\Sigma}_{s=1}^{k} \Gamma_{ks}\, v_{is}$ when $\boldsymbol{\Gamma} \neq \mathbf{I}$, and $\boldsymbol{\Omega}_i = \text{diag}(\sigma_1,...,\sigma_k)$

$\boldsymbol{\Gamma}$ = lower triangular matrix with ones on the diagonal that allows correlation across random parameters when $\boldsymbol{\Gamma} \neq \mathbf{I}$

Individual Error Components

E_{im} = the individual specific underlying random error components, $m = 1,...,M$, $E_{im} \sim N[0,1]$

d_{jm} = 1 if E_{im} appears in utility for alternative j and 0 otherwise

θ_m = scale factor for error component m

γ_{im} = $\exp(\boldsymbol{\gamma}_m'\mathbf{he}_i)$ = heterogeneity in the variances of the error components

λ_{im} = $\theta_m\gamma_{im}$ = standard deviations of random error components

$\boldsymbol{\gamma}_m$ = parameters in the heteroscedastic variances of the error components

$\mathbf{he}_i$ = individual choice invariant characteristics that produce heterogeneity in the variances of the error components

The model specification will dictate which parameters are random and which are not, how the heteroscedasticity, if any, is parameterized, the distributions of the random terms, and how the error components enter the model.

The probabilities defined above are conditioned on the random terms, $\mathbf{v}_i$ and the error components, $\mathbf{E}_i$. The unconditional probabilities are obtained by integrating v_{ik} and E_{im} out of the conditional probabilities: $P_j = E_{\mathbf{v},\mathbf{E}}[P(j|\mathbf{v}_i,\mathbf{E}_i)]$. This is a multiple integral which does not exist in closed form. The integral is approximated by sampling *nrep* draws from the assumed populations and averaging. (See Bhat (1996) and Revelt and Train (1998) and Greene (2003) for discussion.) Parameters are estimated by maximizing the simulated log likelihood,

$$\log L_s = \sum_{i=1}^{N} \log \frac{1}{R} \sum_{r=1}^{R} \prod_{t=1}^{T_i} \frac{\exp\left[\alpha_{ji} + \boldsymbol{\beta}_{ir}'\mathbf{x}_{jit} + \boldsymbol{\Sigma}_{m=1}^{M} d_{jm}\theta_m \exp(\boldsymbol{\gamma}_m'\mathbf{he}_i)E_{im,r}\right]}{\sum_{q=1}^{J_i} \exp\left[\alpha_{qi} + \boldsymbol{\beta}_{ir}'\mathbf{x}_{qit} + \boldsymbol{\Sigma}_{m=1}^{M} d_{qm}\theta_m \exp(\boldsymbol{\gamma}_m'\mathbf{he}_i)E_{im,r}\right]},$$

with respect to $(\boldsymbol{\beta}, \boldsymbol{\Delta}, \boldsymbol{\Gamma}, \boldsymbol{\Omega}, \boldsymbol{\theta}, \boldsymbol{\gamma})$, where

R = the number of replications,

$\boldsymbol{\beta}_{ir}$ = $\boldsymbol{\beta} + \boldsymbol{\Delta}\mathbf{z}_i + \boldsymbol{\Gamma}\boldsymbol{\Omega}_i\mathbf{v}_{ir}$ = the rth draw on $\boldsymbol{\beta}_i$,

$\mathbf{v}_{ir}$ = the rth multivariate draw for individual i,

$E_{im,r}$ = the rth univariate normal draw on the underlying effect for individual i.

(Note that the multivariate draw, $\mathbf{v}_{ir}$ is actually K independent draws. The heteroscedasticity is induced first by multiplying by $\mathbf{\Omega}_i$, then the correlation is induced by multiplying $\mathbf{\Omega}_i\mathbf{v}_{ir}$ by $\mathbf{\Gamma}$.)

The model components may be restricted and varied in several ways.

- A variety of distributions may be chosen for the random parameters, and they need not be the same for all parameters.
- The observed heterogeneity, $\mathbf{\Delta z}_i$, is optional. You may specify that a coefficient is randomly distributed around a fixed mean. Thus, $\boldsymbol{\delta}_k$ may be set to a zero vector for some or all random coefficients.
- σ_k may be set equal to zero for some coefficients. This may change the way a coefficient enters the model. If $\sigma_k = 0$ and $\boldsymbol{\delta}_k = \mathbf{0}$, then the coefficient is a nonrandom fixed parameter. But, including it in $\boldsymbol{\beta}$ allows you to force a coefficient to be positive. This device also allows you to form a hierarchical model with nonrandom coefficients.
- Any coefficient in the model may be fixed at a specific value.
- The heteroscedasticity may apply to some or all (or none) of the random parameters.
- Different variables may be placed in the heterogeneous means ($\mathbf{\Delta z}_i$) or the heteroscedastic variances ($\mathbf{\Omega}_i$) of any of the random parameters.
- The variables that enter the heteroscedasticity of the error components may be different.
- The model with both heteroscedasticity and cross parameter correlation is not estimable. (There is no way to make the covariance heterogeneous.)

A number of additional features are listed in the sections to follow.

10.3 Command for the Random Parameters Logit Models

The command for the mixed logit model is as follows:

```
RPLOGIT     ; Lhs = ... as usual
            ; Choices = ...
            ; ... Utility function specification using
                ; Rhs = ... ; Rh2 = ...  or
                ; Model: U(...) = ... to specify utilities
            ; Fcn = specification of random parameters $
```

(The model command **NLOGIT ; RPL** is equivalent.) The last specification is used to define the random parameters. There are many variants. We begin with the simplest, and add features as we proceed. The **; Fcn** specification takes the basic form

```
; Fcn = parameter label (type)
```

where 'parameter label' is defined either by a variable name that you use in your **; Rhs** specification or by the name you give in your **; Model: ...** definitions and the 'type' is one of the distributions defined in the next section. Alternative specific constants are a special case. You will generally not want to specify the parameters that multiply Rh2 variables as random. These two cases are considered specifically below. For example, the following specifies two normally distributed random parameters:

```
RPLOGIT     ; Lhs  = mode ; Choices = air,train,bus,car
            ; Rhs  = gc,ttme,invc ; Rh2 = hinc
            ; Fcn  = gc(n),ttme(n) $
```

(The 'type' in the example is 'n' indicating normally distributed parameters. Several other specifications would probably be added.) Alternatively, you might use the following to specify a model with two random parameters:

```
RPLOGIT     ; Lhs  = mode ; Choices = air,train,bus,car
            ; Model:U(air) = a_air + bgc*gc + btt*ttme + binvc*invc + ghinc*hinc /
                    U(train,bus,car) = a_ground + bgc*gc
            ; Fcn  = a_ground(n),btt(n) $
```

Note that the specifications of the random parameters are separated by commas, not semicolons. The next several subsections will describe the various parts of the specifications of the random parameters. The last part of this section describes the command builder for this model. Because so much of this model is custom made for the particular application, the command builder is somewhat limited compared to the command form indicated above.

10.3.1 Distributions of Random Parameters in the Model

There are many distributions that can be used for the random parameters. The most common will be the normal, which is used in the example above. Many alternatives are supported, however. A few of these are listed below. The basic distributions are specified with the following:

```
            ; Fcn = parameter name (type), ...
```

The types are

n	normal	$\beta_i = \beta + \sigma v_i,\ v_i \sim N[0,1]$
l	lognormal	$\beta_i = \exp(\beta + \sigma v_i),\ v_i \sim N[0.1]$
u	uniform	$\beta_i = \beta + \sigma v_i,\ v_i \sim U[-1.1,]$
t	triangular	$\beta_i = \beta + \sigma v_i,\ v_i \sim \text{triangle}[-1.1]$
d	dome	$\beta_i = \beta + \sigma v_i,\ v_i \sim 2\times\text{beta}(2,2) - 1$
e	Erlang	$\beta_i = \beta + \sigma v_i,\ v_i \sim \text{gamma}(1,4) - 4$
w	Weibull	$\beta_i = \beta + \sigma\, v_i,\ v_i = 2(-\log u_i)^{\sqrt{.5}},\ u_i \sim U[0,1]$
p	exponential	$\beta_i = \beta + \sigma v_i,\ v_i \sim \text{exponential} - 1$
c	nonstochastic	$\beta_i = \beta$

In the list above, we have denoted the constant in the distribution as 'β.' However, the parameter definition may involve heterogeneity in the mean – so, what appears there may be of the form $\theta_i = \beta + \boldsymbol{\delta}'\mathbf{z}_i$. We have also written the scaling parameter in each form as 'σ,' however, you may also specify heterogeneity in the variances – so what appears there may be of the form $\sigma_i = \sigma \exp(\boldsymbol{\omega}'\mathbf{h}_i)$. The list above suggests the variety of different distributions that may be used.

Any distribution may be used for any parameter. The normal distribution will be the usual choice. However, you may wish to restrict a particular coefficient in the model to be positive. The lognormal distribution is the obvious choice, though there are several other possibilities. The normal, lognormal, exponential, Erlang and Weibull distributions all have infinite ranges. If you wish to restrict the range of variation of a parameter, then the triangular, dome or uniform can be used. The lognormal distribution has an infinite tail in the positive direction and is anchored at zero while the Erlang and Weibull models as specified have infinite range from $\beta - \sigma E[v_i]$ to $+\infty$.

It is important to note that the means and variances of the distributions are not always simple functions when the parameters are not linear functions of the underlying random variables. For all but the Weibull distributions shown above, the mean of v_i is zero, which centers the distributions at β. For the lognormal and Weibull models, the mean depends on the parameters. This is also true of the modified distributions shown below. This means that one must be careful in interpreting the estimated coefficients, even in simple cases in which there is no heterogeneity in the means or variances. It is possible to learn about these empirically however, it is often not possible to state a priori what the population means are for most of the distributions. The problem becomes yet more complicated as additional features such as heterogeneity in the means and heteroscedasticity are added to the model.

Some practical aspects of the specifications are as follows:

- If you will be mixing distributions, the specification of correlated parameters, while allowable, produces ambiguous results. The nature of the correlation is difficult to define. However, the program will have no unusual difficulty estimating a model in which correlated parameters have different distributions. One particular case worth noting is a mixture of normal and lognormal parameters. In such a model, the reported correlation will be between the normally distributed parameter and the log of the lognormally distributed parameter. This is probably not a useful result.

- Researchers often find that the long, thick tail of the lognormal distribution produces an implausible distribution of parameters. .

- Type '*c*' is the same as not including the parameter in the Fcn list, which is how this usually should be done. But sometimes, for convenience, this might be preferred. Variable name (*c*) specifies a free mean and zero variance of the parameter.

Model results for these distributions will display the structural parameters, not necessarily the means and variances of the parameter distributions. Note, for example, that the means of the lognormal and the Weibull distributions are not equal to β; for the lognormal it is $\exp(\beta+\sigma^2/2)$ while for the Weibull it is $\beta+2\sigma\Gamma(1+1/\sqrt{2})$. Consider an example. The following estimates a model with two random parameters. We will use the normal, Weibull and exponentiated Weibull (our 'Rayleigh') distributions. Since the exponentiated Weibull estimator forces the coefficient to be positive, and the coefficients on the two variables would naturally be negative, we reverse the signs on the data before estimation.

The commands are:

```
CREATE    ; mgc = -gc ; mttme = -ttme $
RPLOGIT   ; Lhs  = mode
          ; Choices = air,train,bus,car
          ; Rhs  = mgc,mttme ; Rh2 = one
          ; Fcn  = mgc(n),mttme(n) ? Normally distributed parameters
          ; Maxit = 50 ; Pts = 25 ; Halton ; Pds = 3 $
RPLOGIT   ; Lhs  = mode
          ; Choices = air,train,bus,car
          ; Rhs  = mgc,mttme ; Rh2 = one
          ; Fcn  = mgc(w),mttme(w) ? Weibull distributed parameters
          ; Maxit = 50 ; Pts = 25 ; Halton ; Pds = 3 $
RPLOGIT   ; Lhs  = mode
          ; Choices = air,train,bus,car
          ; Rhs  = mgc,mttme ; Rh2 = one
          ; Fcn  = mgc(r),mttme(r) ? Modified Weibull distributed parameters
          ; Maxit = 50 ; Pts = 25 ; Halton ; Pds = 3 $
```

These are the reported random parameter estimates. (The nonrandom alternative specific constants are not shown.) The values for the random parameters are β and σ. For the normally distributed variables, these are the means and standard deviations. For the other distributions, they are only the structural parameters. To see the similarity, however, note for the coefficient on *mgc* in the Rayleigh model, exp(-3.3585415) is about 0.035, which resembles the value for the normal distribution. Accounting for σ would likely bring them yet closer.

```
---------+Multinomial logit with nonrandom parameters
 MGC     |     .01578374       .00438279     3.601   .0003
 MTTME   |     .09709052       .01043509     9.304   .0000
+--------+-------------------------------------------------------------
---------+Normal Random parameters in utility functions
 MGC     |     .02179446       .00691475     3.152   .0016
 MTTME   |     .14140119       .01958762     7.219   .0000
---------+Derived standard deviations of parameter distributions
 NsMGC   |     .00867259       .01372168      .632   .5274
 NsMTTME |     .07424180       .01489564     4.984   .0000
+--------+-------------------------------------------------------------
---------+Weibull Random parameters in utility functions
 MGC     |     .03546490       .02059812     1.722   .0851
 MTTME   |     .23934417       .03347361     7.150   .0000
---------+Derived standard deviations of parameter distributions
 WsMGC   |     .00683164       .00915062      .747   .4553
 WsMTTME |     .05604114       .01237808     4.527   .0000
+--------+-------------------------------------------------------------
---------+Rayleigh  Random parameters in utility functions
 MGC     |   -3.35385415      1.36800576    -2.452   .0142
 MTTME   |   -1.26324106       .21598047    -5.849   .0000
---------+Derived standard deviations of parameter distributions
 RsMGC   |     .33488248       .89249718      .375   .7075
 RsMTTME |     .47230483       .10945909     4.315   .0000
```

10.3.2 Spreads, Scaling Parameters and Standard Deviations

The RPL model is complicated. It is also necessary to note that the interpretation of the parameters is partly a function of the specification chosen. What are described earlier as the 'means' and 'variances' are actually only those parameters in the simplest cases. The reported parameters may need to be interpreted, and manipulated further to obtain the expected results. We consider several examples. In a model with a normally distributed parameter,

$$\beta_i = \beta + \delta z_i + \sigma v_i,\ v_i \sim N[0,1],$$

$(\beta + \delta z_i)$ is, indeed, the conditional mean and σ is the standard deviation. The model results might appear as follows, in which the parameter on variable *mgc* is specified to have a normal distribution with a mean that is a function of *hinc*, which has a mean of about 35. The specification is

RPLOGIT ; Lhs = mode ; Choices = air,train,bus,car
; Rhs = mgc,ttme,one
; RPL = hinc ; Pts = 15 ; Maxit = 10 ; Pds = 3 ; Fcn = mgc(n) $

```
---------+Random parameters in utility functions
 MGC     |     .01317029       .01052638     1.251   .2109
---------+Nonrandom parameters in utility functions
 TTME    |    -.09909916       .01076768    -9.203   .0000
 A_AIR   |    6.00438917       .69301957     8.664   .0000
 A_TRAIN |    4.09897595       .47136837     8.696   .0000
 A_BUS   |    3.39330467       .48215182     7.038   .0000
---------+Heterogeneity in mean, Parameter:Variable
 MGC:HIN |     .00023602       .00023165     1.019   .3083
---------+Derived standard deviations of parameter distributions
 NsMGC   |     .01723103       .00801915     2.149   .0317
```

According to these results, the population mean of parameters on *mgc* computed at the mean income, or an estimate of $E[\beta_i|E[z_i]] \approx E_z E[\beta_i|z]]$ is roughly .01317029 + 35(.00023602) = .02143099 and the population standard deviation is about .01723103. Suppose in the same model, we change the distribution to lognormal with **; Fcn = mgc(l)**. The results change to

```
---------+Random parameters in utility functions
 MGC     |   -4.69325083       .77314298    -6.070   .0000
---------+Nonrandom parameters in utility functions
 TTME    |    -.09826080       .01030574    -9.535   .0000
 A_AIR   |    5.91314197       .70223867     8.420   .0000
 A_TRAIN |    4.04515395       .49255837     8.213   .0000
 A_BUS   |    3.32819477       .51874172     6.416   .0000
---------+Heterogeneity in mean, Parameter:Variable
 MGC:HIN |     .01415413       .01472297      .961   .3364
---------+Derived standard deviations of parameter distributions
 LsMGC   |     .68944036       .54389813     1.268   .2049
```

But, the reported parameters are those of the underlying normal distribution. In this model,

$$\beta_i = \exp(\beta + \delta z_i + \sigma v_i),\ v_i \sim N[0,1].$$

The conditional (population) mean of the distribution will be

$$E[\beta_i|z_i] = \exp(\beta + \delta z_i + \sigma^2/2).$$

Inserting the estimated parameters and the mean of 35 for income, we obtain an estimate of the overall population mean of 0.0190543, which is quite similar to the .02143099 for the normal distribution. The variance for the lognormal is obtained as

$$\mathrm{Var}[\beta_i|z_i] = \{E[\beta_i|z_i]\}^2 [\exp(\sigma^2) - 1].$$

Inserting our estimates and taking the square root produces an estimate of the population standard deviation of 0.014864085. The result for the normal distribution is .01723103. (We emphasize, we are implicitly averaging over incomes in these computations – the results are close to, but not exactly equal to the analytical results.)

The results for the lognormal distribution, correctly interpreted, are quite similar to those for the normal distribution. The structural parameters, however, are quite different. A similar characterization applies to the other distributions that are obtained as transformations of the underlying random terms. In most cases, it is not possible to obtain closed form results for the overall means and variances – the lognormal distribution is a convenient special case. The program will report its estimates of the structural parameters, but it is not generally possible to disentangle the reduced form to report the actual 'mean' and 'standard deviation' in spite of the labeling of the estimates in the program output.

Random parameter distributions that depend on the uniform distribution present another ambiguity in the interpretation of the results. For the uniform distribution, we estimate the spread of the distribution, not the standard deviation or the variance. Suppose we now change the earlier model to **; Fcn = mgc(u)**. By this construction,

$$\beta_i = \beta + \delta z_i + \sigma v_i,\ v_i \sim U[-1,1],$$

the values of β_i are distributed uniformly between $(\beta + \delta z_i - \sigma)$ and $(\beta + \delta z_i + \sigma)$. The mean is $\beta + \delta z_i$, but the variance is $4\sigma^2/12$, with a standard deviation of $\sigma/\sqrt{3}$. The estimated parameters are as follows:

```
---------+Random parameters in utility functions
 MGC     |     .00893792       .00978908      .913   .3612
---------+Nonrandom parameters in utility functions
 TTME    |    -.09779935       .01063867    -9.193   .0000
 A_AIR   |    5.86320087       .68262859     8.589   .0000
 A_TRAIN |    3.99147415       .46159989     8.647   .0000
 A_BUS   |    3.28433873       .47262187     6.949   .0000
---------+Heterogeneity in mean, Parameter:Variable
 MGC:HIN |     .00021461       .00022919      .936   .3491
---------+Derived standard deviations of parameter distributions
 UsMGC   |     .02222135       .01975890     1.125   .2607
```

Based on these results, the overall mean is about .00893792 + 35(.00021461) = .0164492, again comparable, and the standard deviation is .01289502. What is reported is a scale factor, or spread parameter, not the standard deviation of the distribution.

The triangular distribution presents the same ambiguity. In this model,

$$\beta_i = \beta + \delta z_i + \sigma v_i,\ v_i \sim \text{Triangular}[-1,1],$$

The mean is $\beta + \delta z_i$, but the variance is $\sigma^2/6$, which is one half the variance of the uniform distribution with the same spread (and mean). Repeating the previous estimation, now with **; Fcn = mgc(t)**, we obtain the results below.

```
---------+Random parameters in utility functions
 MGC     |     .01396869       .01082759     1.290   .1970
---------+Nonrandom parameters in utility functions
 TTME    |    -.09931295       .01083732    -9.164   .0000
 A_AIR   |    6.00304781       .69769310     8.604   .0000
 A_TRAIN |    4.10077954       .47428938     8.646   .0000
 A_BUS   |    3.39796835       .48316868     7.033   .0000
---------+Heterogeneity in mean, Parameter:Variable
 MGC:HIN |     .00021077       .00023228      .907   .3642
---------+Derived standard deviations of parameter distributions
 TsMGC   |     .05487307       .02445605     2.244   .0248
```

Now, the mean is .02134585 and the standard deviation is $.05487307/\sqrt{6} = .022401837$.

The preceding serves to emphasize the need to interpret the estimated model parameters on a case by case basis. Each distribution has different characteristics. Worse yet, in some of those cases, we do not even have the convenient formulas given above to use to convert the parameters to population moments. Consider the Weibull distribution, which we obtain with **; Fcn = mgc(w)**. For this model,

$$\exp(\beta + \delta z_i + \sigma v_i),\ v_i = (-2\log u_i)^{\sqrt{.5}},\ u_i \sim U[0,1].$$

The estimated parameters of the model are as follows:

```
---------+Random parameters in utility functions
 MGC     |   -3.44822322      1.06929334    -3.225   .0013
---------+Nonrandom parameters in utility functions
 TTME    |    -.09807615       .01018490    -9.630   .0000
 A_AIR   |    5.90493475       .69570219     8.488   .0000
 A_TRAIN |    4.04347670       .49138509     8.229   .0000
 A_BUS   |    3.32608885       .51475257     6.462   .0000
---------+Heterogeneity in mean, Parameter:Variable
 MGC:HIN |     .01555286       .01425775     1.091   .2753
---------+Derived standard deviations of parameter distributions
 WsMGC   |     .79003797       .83918130      .941   .3465
```

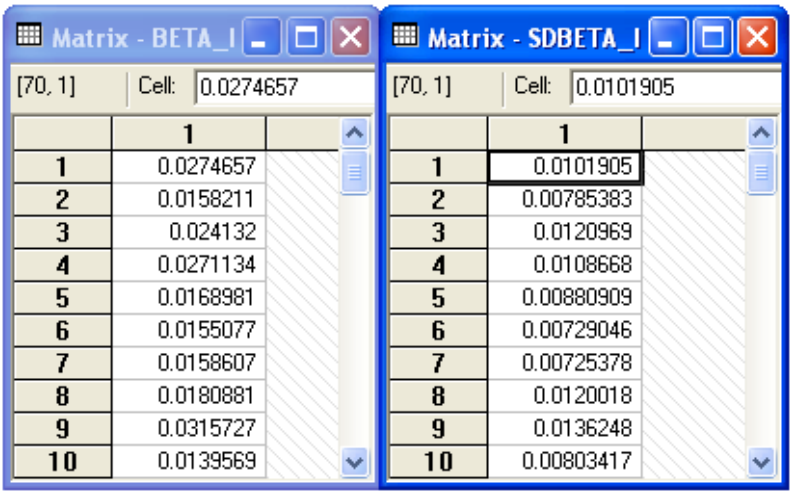
There is no obvious way to translate these back to a mean and variance. But, there is an indirect method. If you add

; Parameters

to your **RPLOGIT** command, then *NLOGIT* creates two matrices from the model results. The matrix *beta_i* contains for each random parameter (column) and each individual (row), an estimate of

$$\hat{\beta}_{ik} = \hat{E}[\beta_{ik} \mid \text{all information about individual } i].$$

The information about individual *i* includes their choices, so this is not quite the same as the estimator that we are using above, $E[\boldsymbol{\beta}_i|\mathbf{z}_i]$. But, since the average of conditional means gives the unconditional mean, the average of the estimates contained in *beta_i* provides an estimator of the unconditional population mean that we are estimating above. Figure 10.1 below shows the first 10 rows of this 70×1 matrix as created by the model command that generated the Weibull results above.



	1
1	0.0274657
2	0.0158211
3	0.024132
4	0.0271134
5	0.0168981
6	0.0155077
7	0.0158607
8	0.0180881
9	0.0315727
10	0.0139569

	1
1	0.0101905
2	0.00785383
3	0.0120969
4	0.0108668
5	0.00880909
6	0.00729046
7	0.00725378
8	0.0120018
9	0.0136248
10	0.00803417

Figure 10.1 Estimated Conditional Means and Standard Deviations

We can estimate the overall mean by averaging the elements in *beta_i*. This produces

```
MATRIX      ; ebi = list ; 1/70*beta_i'1 $
              +--------------
             1|      .01984
```

which is the now familiar result. Estimating the population variance is a bit more complicated because the population variance is not the average of the conditional variances. Rather, the variance we seek equals the average of the conditional variances (squares of the elements in *sdbeta_i*) plus the variance of the conditional means. The computation can be done (a bit inelegantly) with

```
MATRIX      ; vi = Dirp(sdbeta_i,sdbeta_i) $
MATRIX      ; evi = 1/70*vi'1 ; vei = 1/70*beta_i'beta_i - ebi*ebi $
MATRIX      ; v = evi + vei ; Peek ; sd = Sqrt(v) $
```

The result of this computation is 0.015192655. Recall, the counterpart for the normal distribution that we examined at the outset was .01723103.

10.3.3 Alternative Specific Constants

If you have used the **; Rhs = list** specification with choices specific constants, then the constants will be labeled *a_name*. For example, if you have used

; Choices = bus,train,car
; Rhs = one,cost

then to specify the model for random ASCs, you might use

; Fcn = a_bus(n),a_train(n)

If you are using the **; Model:** form, then you will have supplied your own names for the ASCs.

Random choice specific constants in the random utility model with cross section data produce a random term that is a convolution of the original extreme value random variable and the one specified in your model command. Suppose, for example, that you specify a normally distributed random constant for '*car*.' Then, the utility function for *car* will be

$$U(car) = \alpha_{car} + \text{(the rest of the utility function)} + \sigma_{car} v_{car} + \varepsilon_{car}$$
$$= \alpha_{car} + \text{(the rest of the utility function)} + u_{car}.$$

The random term in this equation is the sum of a normally distributed variable and one with an extreme value distribution. This produces a different stochastic model, but probably not a useful extension of the model in general. For this reason, unless you are using panel data – it is generally not useful to specify random constant terms in the random parameters logit model. That said, however, there is an exception which might prove useful. Random constant terms that are correlated will produce correlation across the alternatives, which is one of the oft cited virtues of the multinomial probit model. In addition, the error components logit specification produces a useful extension that serves much the same function as a random constant term.

10.3.4 Heterogeneity in the Means of the Random Parameters

The **RPLOGIT** command requests the random parameters model generally, with the parameters specified in the **; Fcn** list varying around a mean that is the same for all individuals. The variables in $\mathbf{z}_i$ provide the variation of the mean across individuals. To specify the variables in $\mathbf{z}_i$, use

; RPL = list of variables in $\mathbf{z}_i$

If you desire to specify that $\mathbf{z}_i$ enter the means of some of the coefficients but not all, you can change the specification of the random coefficients in the **; Fcn** specification as follows:

name (type) implies $\mathbf{z}_i$ enters the mean
name [type] implies that $\mathbf{z}_i$ does not enter the mean.

The difference here is the parentheses in the first as opposed to the brackets in the second. The second of these forces the applicable row of Δ to contain zeros instead of free parameters. There are also some variations on this specification that allow some flexibility in the construction of Δ. First, an alternative, equivalent form of **name [type]** is

name (type | #)

This requests that if there are RPL variables (**; RPL = list**), these not appear in the mean for this parameter. This puts a row of zeros in the Δ matrix. For example,

```
; RPL = income
; Fcn  = gc(n),ttme(n|#)
```

specifies that *income* does not appear in the mean of the *ttme* parameter. This form may be extended to exclude and include specific variables from the RPL list in the mean of a particular parameter. The specification is

name (type | # pattern)

where the pattern consists of ones and zeros which indicate which variables in the list are included (ones) and excluded (zeros). There must be the same number of items in the pattern as there are in the list. For example, the specification

```
; RPL = age,sex,income
; Fcn  = gc(n),
         ttme(n|#101)
         invt (n|#011)
         invc (n|#000)
```

includes all three variables in the mean of *gc*, excludes *sex* from the mean of *ttme*, excludes *age* from the men of *invt*, and excludes all three variables from the mean of *invc*. All parameters may be specified independently, and there is no restriction on how this feature is used. Do note, however, if you exclude an RPL variable from all parameters, the model becomes inestimable.

10.3.5 Correlated Parameters

The model specified thus far assumes that the random parameters are uncorrelated. Use

; Correlation

to allow free correlation among the parameters. In this case, estimates of the below diagonal elements of Γ will be obtained with the other parameters of the model. No restrictions may be imposed on these new parameters. After these are presented, the elements of $\Sigma = \Gamma\Gamma'$ are given. An example appears below. Some ambiguity in the results will be unavoidable when this feature is used with other modifications of the model, such as mixed distributions and heteroscedasticity. The most favorable case for use of this feature would be a sparse model,

$$\beta_i = \beta + \Gamma \mathbf{v}_i.$$

We would note, many, perhaps most of the received applications of the mixed logit model are of this form – it is much less restrictive than its bare appearance would suggest.

In the model developed thus far, the covariance matrix for the random components for the simple distributions (normal, uniform, triangle) is

$$\mathrm{Var}[\boldsymbol{\beta}_i|\mathbf{x}_i,\mathbf{z}_i] = \boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Gamma}'.$$

In the uncorrelated case, $\boldsymbol{\Gamma}$ is a diagonal matrix, and the variance of β_{ik} is simply σ_k^2. When the parameters are correlated, then the diagonal element of $\boldsymbol{\Sigma}$ is $\boldsymbol{\gamma}_k'\boldsymbol{\gamma}_k$ where $\boldsymbol{\gamma}_k$ is the kth row of $\boldsymbol{\Gamma}$. The model results will show the elements of $\boldsymbol{\Gamma}$ and the implied standard deviations. The following demonstrates the computations. The command below specifies two correlated random parameters.

```
RPLOGIT     ; Lhs = mode ; Choices = air,train,bus,car
            ; Rhs = gc,ttme
            ; Rh2 = one
            ; Fcn = gc(n),ttme(n)
            ; Correlated
            ; Maxit = 50 ; Pts = 25 ; Halton ; Output = 3 ; Pds = 3 $
```

The relevant results from estimation are as follows. The coefficients reported are, first, β from the random parameter distributions, then the nonstochastic β from the distributions of the nonrandom alternative specific constants. The next results display the elements of the 2×2 lower triangular matrix, Γ. The diagonal elements appear first, then the below diagonal element(s). The matrix Γ is shown again, in natural form at the end of the results, labeled 'Cholesky matrix.' The 'Standard deviations of parameter distributions' are derived from $\boldsymbol{\Gamma}$. The first is $(.011001342)^{1/2} = .001100134$. The second is $((-.07458)^2 + .03678^2)^{1/2} = .08315251$. The standard errors for these estimators are computed using the delta method. Hensher, Rose and Greene (2005) discuss the Cholesky decomposition in detail with numerous examples.

```
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
---------+Random parameters in utility functions
 GC      |    -.02260684       .00724332    -3.121   .0018
 TTME    |    -.14522848       .02205029    -6.586   .0000
---------+Nonrandom parameters in utility functions
 A_AIR   |    8.70238058      1.22465947     7.106   .0000
 A_TRAIN |    6.95973395      1.03548341     6.721   .0000
 A_BUS   |    6.12199207      1.13357506     5.401   .0000
---------+Diagonal values in Cholesky matrix, L.
 NsGC    |     .01100134       .01124017      .979   .3277
 NsTTME  |     .03678160       .03024421     1.216   .2239
---------+Below diagonal values in L matrix. V = L*Lt
 TTME:GC |    -.07457516       .02353048    -3.169   .0015
---------+Standard deviations of parameter distributions
 sdGC    |     .01100134       .01124017      .979   .3277
 sdTTME  |     .08315251       .01967123     4.227   .0000
Correlation Matrix for Random Parameters
Matrix COR.MAT. has  2 rows and  2 columns.
         GC            TTME
        +----------------------------
GC      |    1.00000      -.89685
TTME    |    -.89685      1.00000
```

```
Covariance Matrix for Random Parameters
Matrix COV.MAT. has  2 rows and  2 columns.
         GC            TTME
        +----------------------------
GC      |     .00012      -.00082
TTME    |    -.00082       .00691

Cholesky Matrix for Random Parameters
Matrix Cholesky has  2 rows and  2 columns.
         GC            TTME
        +----------------------------
GC      |     .01100    .0000000D+00
TTME    |    -.07458       .03678
```

We emphasize, these results apply to the linear functions of the underlying random variables, not necessarily to the implied distributions of the random parameters themselves. In most of the specifications, the parameters involve nonlinear transformations of these variables.

10.3.6 Command Builders for the RPL Models

With a few important exceptions the random parameters logit (RPL) model can be specified with the command builder by selecting Model:Discrete Choice/Multinomial Probit, HEV, RPL. The Main page, shown in Figure 10.2, requests specification of the choice variable and the utility functions. This page provides both ways to do this specification. The random parameters model is set up on the Options page, shown in Figure 10.3. Note that there are a few options not specified in the command builder, notably the **; Sdv** specification and the technical controls of the simulation. However, in the random parameters window, you can add these additional specifications as text. Thus, where we have typed '**gc(n),ttme(n)**' we could have typed '**gc(n),ttme(n) ; Stv = s1,1.0**' which would have added the additional specification, as a text string.

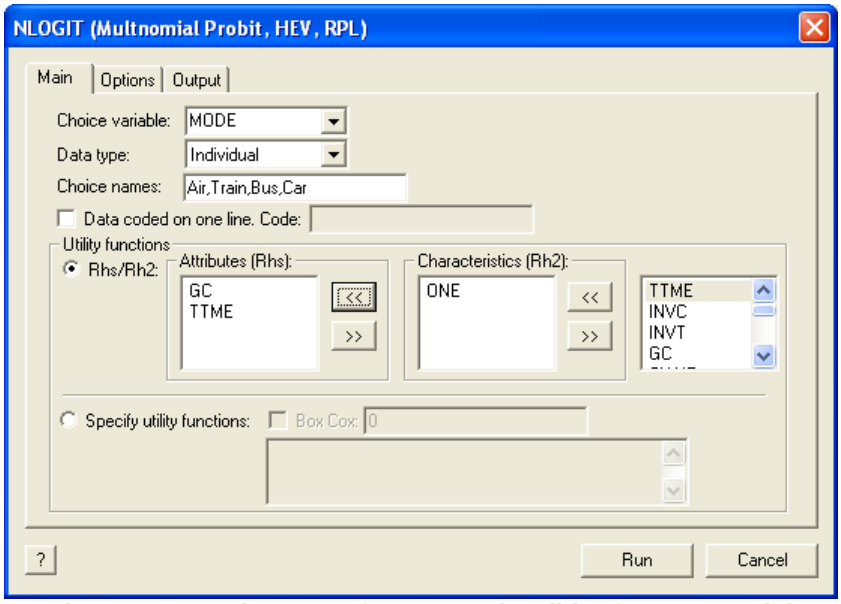


Figure 10.2 Main Page of Command Builder for RPL Model

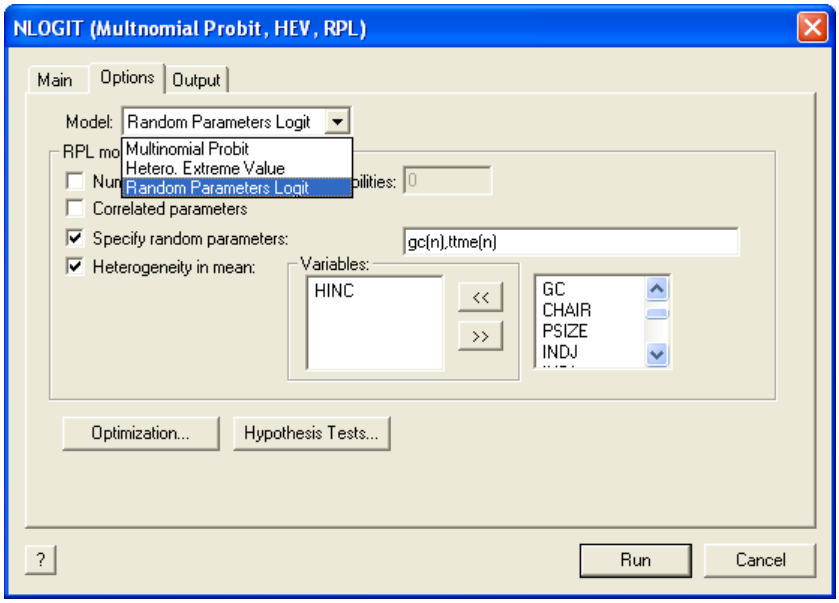


Figure 10.3 Options Page of Command Builder for RPL Model

General options for *NLOGIT*'s models are requested on the Output page, shown in Figure 10.4. A separate page for model estimates may be opened by clicking Model Estimates in the lower right of the Output page. See Figure 10.5

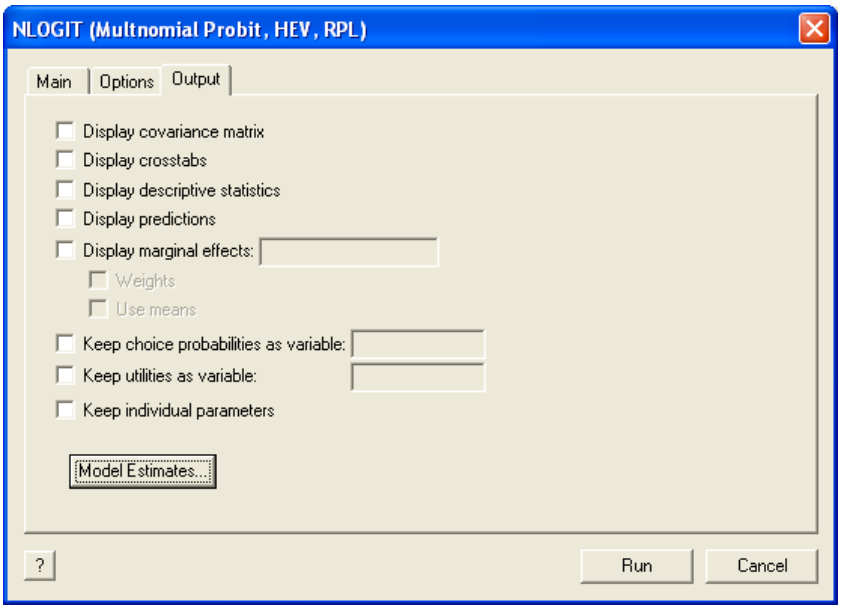


Figure 10.4 Output Page of Command Builder for RPL Model

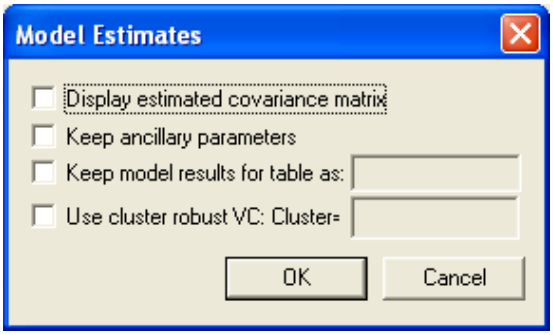


Figure 10.5 Model Estimates Page of Command Builder for RPL Model

10.4 Heteroscedasticity and Heterogeneity in the Variances

The random parameters model allows heterogeneity in the variances as well as in the means in the distributions of the random parameters. The model is expanded to

$$\sigma_{ik} = \sigma_k \exp[\boldsymbol{\omega}_k{}'\mathbf{hr}_i],$$

If $\boldsymbol{\gamma}$ equals $\mathbf{0}$, this returns the homoscedastic model. The implied form of the RPL model is

$$\begin{aligned}\beta_{ik} &= \beta + \boldsymbol{\delta}_k{}'\mathbf{z}_i + \sigma_{ik}v_{ik}.\\ &= \beta + \boldsymbol{\delta}_k{}'\mathbf{z}_i + \sigma_k \exp(\boldsymbol{\omega}_k{}'\mathbf{hr}_i)v_{ik}.\end{aligned}$$

Request the heteroscedasticity model with

; Hfr = list of variables in $\mathbf{hr}_i$

The variables in $\mathbf{hr}_i$ may be any variables, but they must be choice invariant. Only the last value in J rows for choice situation *it* is used. This specification will produce the same form of heteroscedasticity in each parameter distribution – note that each parameter has its own parameter vector, $\boldsymbol{\gamma}_k$.

There is a method of modifying the specification of the heterogeneous means of the parameters so that some RPL variables in $\mathbf{z}_i$ may appear in the means of some parameters and not others. A similar construction may be used for the variances. The general form of the specification is as follows: For any parameter specification,

; Fcn = name (type ...)

(it may contain more information beyond just the distribution type), the specification may end with an exclamation point, '**!**' to indicate that the particular parameter is to be homoscedastic even if others are heteroscedastic. For example, the following produces a model with heterogeneous means, and one heteroscedastic variance:

```
; RPL = age,sex
; Hfr  = income
; Fcn  = gc(n),ttme(n | # 01 !)
```

The parameter on *gc* has both heterogeneous mean and heteroscedastic variance. The parameter on *ttme* has heterogeneous mean, but *age* is excluded, and homogeneous variance. Note that there are no commas before or after the **!**. As in the case of the means, when there is more than one Hfr variable, you may add a pattern to the specification to include and exclude them from the model. To continue the previous example, consider

; RPL = age,sex
; Hfr = income,family,urban
; Fcn = gc(n),ttme(n | # 01 ! 101)

Now, the variance for *gc* includes all three variables, but the variance for *ttme* excludes *family*.

NOTE: The model with both correlated parameters (**; Correlated**) and heteroscedastic random parameters is not estimable. If your model command contains both **; Correlated** and **; Hfr = list**, the heteroscedasticity takes precedence, and the **; Correlated** is ignored.

10.5 Controlling the Simulations

There are two parameters of the simulations that you can change, the number of draws used in the replications and the type of sequence used to effect the integration.

10.5.1 Number and Initiation of the Random Draws

R is the number of points (replications) in the simulation. Authors differ in the appropriate value. Generally, the more complex the model is, and the greater the number of random parameters in it, the larger will be the number of draws required to stabilize the estimates. Train recommends several hundred. Bhat suggests 1,000 is an appropriate value. The program default is 100. You can choose the value with

; Pts = number of draws, R

The RPL model is fairly time consuming to estimate. For exploratory work while you develop a final model specification, you will find that setting *R* to a small value such as 10 or 20 (as we do in the examples in this chapter) will be a useful time saver. Once a specification is finalized, a larger value will be appropriate.

In order to replicate an estimation, you must use the same random draws. One implication of this is that if you give the identical model command twice in sequence, you will not get the identical set of results because the random draws in the sequences will be different. To obtain the same results, you must reset the seed of the random number generator with a command such as

CALC ; Ran(seed value) $

We generally use **CALC ; Ran(12345) $** before each of our examples, precisely for this reason. The specific value you use for the seed is not of consequence; any odd number will do.

10.5.2 Halton Draws and Random Draws for Simulations

The standard approach to simulation estimation is to use random draws from the specified distribution. As suggested immediately above, good performance in this connection usually requires fairly large numbers of draws. The drawback to this approach is that with large samples and large models, this entails a huge amount of computation and can be very time consuming. A currently emerging literature has documented dramatic speed gains with no degradation in simulation performance through the use of a smaller number of Halton draws instead of a large number of random draws. Some authors have found that a Halton sequence with a far small number of replications (as low as a tenth for a single parameter) is often as effective as a far larger number of random draws. To use this approach, add

; Halton

to your model command.

10.6 Model Estimates

Because of the numerous components of the model, the results for a random parameters model are somewhat more involved than for other specifications. For an example, we use the command below, which specifies a fairly involved, heterogeneous RPL model with two error components.

```
RPLOGIT     ; Lhs  = mode ; Choices = air,train,bus,car
            ; Rhs  = gc,ttme,one
            ; Effects: gc(air)
            ; RPL = hinc
            ; Pts = 25
            ; Maxit = 100
            ; Halton
            ; Fcn = gc(n),ttme(n)
            ; Correlated
            ; ECM = (air,car),(train,bus) $
```

The initial display options for the model requested with **; Show** are the same as in other cases. The **; Describe** and **; Crosstab** are as well. These were not requested below. As usual, the estimates for the MNL model are given first. These are used as starting values for the estimates. Other parameters of the distributions of the random components are started at zeros.

```
+---------------------------------------------+
| Start values obtained using MNL model       |
| Dependent variable               Choice     |
| Number of observations              210     |
| Log likelihood function       -199.9766     |
| Number of parameters                  5     |
| Info. Criterion: AIC =          1.95216     |
|   Finite Sample: AIC =          1.95356     |
| Info. Criterion: BIC =          2.03185     |
| Info. Criterion:HQIC =          1.98438     |
```

```
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -283.7588  .29526  .28389 |
| Chi-squared[ 2]          =    167.56429     |
| Prob [ chi squared > value ] =   .00000     |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 GC      |    -.01578374       .00438279    -3.601   .0003
 TTME    |    -.09709052       .01043509    -9.304   .0000
 A_AIR   |    5.77635901       .65591873     8.807   .0000
 A_TRAIN |    3.92300113       .44199360     8.876   .0000
 A_BUS   |    3.21073472       .44965283     7.140   .0000
```

Results from the random parameters logit model display the standard pattern, an initial box containing diagnostic statistics, followed by an indication of the size (R) and type (random or Halton) of the simulation, then the output for the model. In this model, there are likely to be many different components of the probability function, such as in the earlier example. As shown in the sample output below, the results will contain the lowest level structural parameters, first the constant terms in the random parameters in the utility functions, then the nonrandom parameters, and, finally, the parameters of the underlying distribution. The final parameters shown are the scale factors for the underlying random terms in the parameters. The leading character matches your specification in the **; Fcn** part of your command. The ‘s’ to follow indicates this is a diagonal element of **Γ**. Finally, up to five characters of the original name are appended.

```
+---------------------------------------------+
| Random Parms/Error Comps. Logit Model       |
| Maximum Likelihood Estimates                |
| Dependent variable                 MODE     |
| Weighting variable                 None     |
| Number of observations              210     |
| Iterations completed                 36     |
| Log likelihood function       -178.3248     |
| Number of parameters                 12     |
| Info. Criterion: AIC =          1.81262     |
|   Finite Sample: AIC =          1.82016     |
| Info. Criterion: BIC =          2.00388     |
| Info. Criterion:HQIC =          1.88994     |
| Restricted log likelihood     -291.1218     |
| McFadden Pseudo R-squared      .3874565     |
| Chi squared                    225.5941     |
| Degrees of freedom                   12     |
| Prob[ChiSqd > value] =         .0000000     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| No coefficients   -291.1218  .38746  .37556 |
| Constants only    -283.7588  .37156  .35936 |
| At start values   -199.9766  .10827  .09096 |
| Response data are given as ind. choice.     |
+---------------------------------------------+
```

```
+---------------------------------------------+
| Notes No coefficients=> P(i,j)=1/J(i).      |
|       Constants only => P(i,j) uses ASCs    |
|         only. N(j)/N if fixed choice set.   |
|         N(j) = total sample frequency for j |
|         N    = total sample frequency.      |
|       These 2 models are simple MNL models. |
|       R-sqrd = 1 - LogL(model)/logL(other)  |
|       RsqAdj=1-[nJ/(nJ-nparm)]*(1-R-sqrd)   |
|         nJ   = sum over i, choice set sizes |
+---------------------------------------------+
+---------------------------------------------+
| Random Parms/Error Comps. Logit Model       |
| Replications for simulated probs. =  25     |
| Halton sequences used for simulations       |
| RPL model has correlated parameters         |
| Hessian was not PD. Using BHHH estimator.   |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
---------+Random parameters in utility functions
 GC      |    -.03344523       .02505267    -1.335   .1819
 TTME    |    -.23084818       .08682355    -2.659   .0078
---------+Nonrandom parameters in utility functions
 A_AIR   |    15.2077878      5.00957004     3.036   .0024
 A_TRAIN |    12.7374035      4.54631279     2.802   .0051
 A_BUS   |    11.4866808      4.49644235     2.555   .0106
---------+Heterogeneity in mean, Parameter:Variable
 GC:HIN  |    -.00048503       .00052537     -.923   .3559
 TTME:HIN|    -.00098231       .00095140    -1.032   .3018
---------+Diagonal values in Cholesky matrix, L.
 NsGC    |     .01920669       .02520301      .762   .4460
 NsTTME  |     .04635102       .04963601      .934   .3504
---------+Below diagonal values in L matrix. V = L*Lt
 TTME:GC |     .14938411       .06697675     2.230   .0257
---------+Standard deviations of latent random effects
 SigmaE01|    1.47749388      1.42144790     1.039   .2986
 SigmaE02|    1.65809550      1.69694056      .977   .3285
---------+Standard deviations of parameter distributions
 sdGC    |     .01920669       .02520301      .762   .4460
 sdTTME  |     .15640981       .06299625     2.483   .0130

 Random Effects Logit Model
 Appearance of Latent Random Effects in Utilities
 Alternative   E01 E02
+-------------+---+---+
|  AIR        | * |   |
+-------------+---+---+
|  TRAIN      |   | * |
+-------------+---+---+
|  BUS        |   | * |
+-------------+---+---+
|  CAR        | * |   |
+-------------+---+---+
```

```
Parameter Matrix for Heterogeneity in Means.
Matrix Delta    has  2 rows and  1 columns.
         HINC
        +--------------
GC      |    -.00049
TTME    |    -.00098
Correlation Matrix for Random Parameters
         GC            TTME
        +----------------------------
GC      |    1.00000       .95508
TTME    |     .95508      1.00000
Covariance Matrix for Random Parameters
         GC            TTME
        +----------------------------
GC      |     .00037       .00287
TTME    |     .00287       .02446
Cholesky Matrix for Random Parameters
         GC            TTME
        +----------------------------
GC      |     .01921    .0000000D+00
TTME    |     .14938       .04635
```

Note two important points about the estimated covariance matrix of the distribution of the random parameters:

- If Γ is diagonal, then the diagonal elements are used to scale the random elements in the parameters. However, these scale parameters are only the standard deviations of the random terms when these variables are normally distributed. Otherwise, there is some specific scale parameter that must be added to the calculation.

- If Γ is not diagonal, then Γ is not the covariance matrix of the random terms, and the diagonal elements of Γ are not the standard deviations even in the normal case. In this instance, Γ is the Cholesky decomposition of the covariance matrix, which must be recovered from the estimates. The results given will include this decomposition, as shown below for this application.

Partial effects for the RPL model are computed in the same fashion as for other models, with one important exception. As in other cases, the elasticities are computed by individual, and averaged to obtain the estimate. However, in the RPL model, the individual specific estimates of the parameters described in the next section, not the population averages, are used to compute the estimates.

```
+-----------------------------------------------------+
| Elasticity             averaged over observations.|
| Attribute is GC       in choice AIR                |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.    |
|                                  Mean    St.Dev   |
| *     Choice=AIR               -.7700     .4918   |
|       Choice=TRAIN              .8787    1.0465   |
|       Choice=BUS                .9346    1.0685   |
|       Choice=CAR                .6412    1.7282   |
+-----------------------------------------------------+
```

Results saved automatically by this estimator are the same as the other estimators in *NLOGIT*, i.e.,

Matrices: *b* and *varb*

Scalars: *logl*, *kreg*, *nreg*
(Note that *nreg* is the number of individuals, not the number of rows of data in the sample.)

Last Model: See Chapter 6 for discussion of how to recover previous results.

You can also save the probabilities and utilities as follows:

; Prob = saves the unconditional probabilities, based on individual parameters,
; Utility = saves the values of utility functions, based on individual parameters.

This estimator will also save various matrices. These are discussed in the next section.

10.7 Individual Specific Estimates

If you include

; Parameters

in your **RPLOGIT** command, *NLOGIT* will create an $n \times K$ matrix named *beta_i* that contains in a row for each individual an estimate of the random parameters in $E[\beta_i|\text{all data for individual } i]$. The model command,

```
RPLOGIT    ; Lhs  = mode ; Choices = air,train,bus,car
           ; Rhs  = mgc,ttme,one
           ; RPL = hinc ; Pts = 15 ; Maxit = 10 ; Pds = 3 ; Parameters
           ; Fcn  = mgc(n) $
```

specifies one random parameter. The sample in use has 210/3 = 70 individuals. The matrix shown below contains the conditional estimates of the mean of the parameter on *mgc*. (The additional matrix *sdbeta_i*, is explained below.)

Matrix - BETA_I

[70, 1] Cell: 0.0274657

	1
1	0.0274657
2	0.0158211
3	0.024132
4	0.0271134
5	0.0168981
6	0.0155077
7	0.0158607
8	0.0180881
9	0.0315727
10	0.0139569

Matrix - SDBETA_I

[70, 1] Cell: 0.0101905

	1
1	0.0101905
2	0.00785383
3	0.0120969
4	0.0108668
5	0.00880909
6	0.00729046
7	0.00725378
8	0.0120018
9	0.0136248
10	0.00803417

Figure 10.6 Estimated Conditional Means and Standard Deviations

The next section will describe how these matrices are computed.

10.7.1 Computing Individual Specific Parameter Estimates

The random parameters model and the simulation based estimator used to estimate it allow the analyst to derive more information from the data than is usually available from models with fixed parameters. In particular, the model specifies that

$$\boldsymbol{\beta}_i = \boldsymbol{\beta} + \boldsymbol{\Delta}\mathbf{z}_i + \boldsymbol{\Gamma}\,\boldsymbol{\Omega}_i\,\mathbf{v}_i,$$

where, for simplicity, if there are any, we include the alternative specific constants in $\boldsymbol{\beta}_i$, and where, if there are nonrandom parameters in the model, these are accommodated simply by having rows and columns of zeros in the appropriate places in $\boldsymbol{\Gamma}$ and $\boldsymbol{\Omega}_i$. There may also be rows of zeros in $\boldsymbol{\Delta}$ for parameters that have homogeneous means. We are interested in learning as much as possible about $\boldsymbol{\beta}_i$ and functions of $\boldsymbol{\beta}_i$ from the data. The unconditional mean of $\boldsymbol{\beta}_i$ is

$$E[\boldsymbol{\beta}_i \mid \mathbf{z}_i] = \boldsymbol{\beta} + \boldsymbol{\Delta}\mathbf{z}_i.$$

Absent any other information, this provides the template that one would use to form their best estimate of $\boldsymbol{\beta}_i$. However, there is other information about individual i in the sample, namely the choices they made, $\mathbf{y}_i$ and other information about their heterogeneity, $\mathbf{hr}_i$. Moreover, we may also have information about individual specific error components, E_{im}, specifically in the form of $\mathbf{he}_i$, the observed heterogeneity in the variation of the error components. The following details a method of forming a conditional estimator, $E[\boldsymbol{\beta}_i \mid \text{all data on individual } i]$.

By using Bayes Theorem, we can form the joint distribution of $\boldsymbol{\beta}_i$ and $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{it})$ as follows: Denote the unconditional (marginal) distribution of $\boldsymbol{\beta}_i|\mathbf{z}_i,\mathbf{hr}_i$ as $p(\boldsymbol{\beta}_i|\mathbf{z}_i,\mathbf{hr}_i)$. This distribution is implied by whatever is assumed about $\mathbf{v}_i$ in the general model,

$$\boldsymbol{\beta}_i = \boldsymbol{\beta} + \boldsymbol{\Delta}\mathbf{z}_i + \boldsymbol{\Gamma}\,\boldsymbol{\Omega}_i\,\mathbf{v}_i$$

where, if there is heteroscedasticity, $\omega_{ik} = \sigma_k \exp[\boldsymbol{\omega}_k'\mathbf{hr}_i]$. (Elements of $\boldsymbol{\beta}_i$ might also be functions of the exponent of this expression for the lognormal and Weibull distributions.) We can also form the conditional distribution of $(\mathbf{y}_i|\boldsymbol{\beta}_i,\mathbf{x}_i,\mathbf{he}_i,\mathbf{E}_i)$ based on the assumptions about $\mathbf{v}_i$ and $\mathbf{E}_i = (E_{i1},E_{i2},...,E_{iM})$ in the conditional multinomial logit model,

$$\text{Prob}(y_{it} = j_{it}, t=1,...,T_i) = \prod_{t=1}^{T_i} \frac{\exp\left[\alpha_{ji} + \boldsymbol{\beta}_i'\mathbf{x}_{jit} + \Sigma_{m=1}^{M} d_{jm}\theta_m \exp(\boldsymbol{\gamma}_m'\mathbf{he}_i)E_{im}\right]}{\sum_{q=1}^{J_i}\exp\left[\alpha_{qi} + \boldsymbol{\beta}_i'\mathbf{x}_{qit} + \Sigma_{m=1}^{M} d_{qm}\theta_m \exp(\boldsymbol{\gamma}_m'\mathbf{he}_i)E_{im}\right]}.$$

(The conditional distribution is defined by the multinomial logit probabilities for the outcomes that have been assumed throughout.) We are looking ahead a bit here and treating the panel data case here rather than developing it separately later. Note as well that $\mathbf{x}_i$ denotes the collection of data on attributes and characteristics that appear in the utility functions for all the choices and in all periods or choice situations. Denote this implied conditional distribution as $p(\mathbf{y}_i|\boldsymbol{\alpha}_i,\boldsymbol{\beta}_i,\mathbf{x}_i,\mathbf{he}_i,\mathbf{E}_i)$ where $\boldsymbol{\alpha}_i$ is the set of ASCs. With these in hand, we will form $p(\boldsymbol{\beta}_i|\mathbf{y}_i,\mathbf{x}_i,\mathbf{z}_i,\mathbf{hr}_i,\mathbf{he}_i,\mathbf{E}_i)$ as follows:

First, we will have to eliminate $\mathbf{E}_i$ from the conditional distribution of $\mathbf{y}_i$. The unconditional distribution is

$$p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i) = \int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\mathbf{E}_i)\, d\mathbf{E}_i .$$

Note that the marginal distribution is actually known – it is the M-variate standard normal distribution. Nonetheless, it will be more convenient to carry it through in generic form below. We now obtain the conditional density of $\boldsymbol{\beta}_i$ using Bayes theorem:

$$p(\boldsymbol{\beta}_i \mid \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, \mathbf{he}_i, \mathbf{hr}_i) = \frac{\int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\mathbf{E}_i)\, d\mathbf{E}_i\, p(\boldsymbol{\beta}_i \mid \mathbf{z}_i, \mathbf{hr}_i)}{\int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{z}_i, \mathbf{he}_i, \mathbf{hr}_i, \mathbf{E}_i)\, p(\mathbf{E}_i)\, d\mathbf{E}_i}$$

$$= \frac{\int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\mathbf{E}_i)\, d\mathbf{E}_i\, p(\boldsymbol{\beta}_i \mid \mathbf{z}_i, \mathbf{hr}_i)}{\int_{\boldsymbol{\beta}_i}\int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\mathbf{E}_i)\, d\mathbf{E}_i\, p(\boldsymbol{\beta}_i \mid \mathbf{z}_i, \mathbf{hr}_i)\, d\boldsymbol{\beta}_i}.$$

Note that it is the joint density, $p(\boldsymbol{\beta}_i,\mathbf{y}_i|\mathbf{x}_i,\mathbf{z}_i,\mathbf{hr}_i,\mathbf{hf}_i)$ that appears in the fraction, the product of the conditional density times the marginal density. Proceeding, we are interested in forming the conditional expectation, $E(\boldsymbol{\beta}_i|\mathbf{y}_i,\mathbf{x}_i,\mathbf{z}_i,\mathbf{hr}_i,\mathbf{hf}_i)$. Since the preceding gives the conditional density, the conditional expectation is formed in the usual manner,

$$E(\boldsymbol{\beta}_i \mid \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, \mathbf{he}_i, \mathbf{hr}_i) = \frac{\int_{\boldsymbol{\beta}_i} \boldsymbol{\beta}_i \int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\mathbf{E}_i)\, d\mathbf{E}_i\, p(\boldsymbol{\beta}_i \mid \mathbf{z}_i, \mathbf{hr}_i)\, d\boldsymbol{\beta}_i}{\int_{\boldsymbol{\beta}_i}\int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\mathbf{E}_i)\, d\mathbf{E}_i\, p(\boldsymbol{\beta}_i \mid \mathbf{z}_i, \mathbf{hr}_i)\, d\boldsymbol{\beta}_i}$$

$$= \frac{\int_{\boldsymbol{\beta}_i}\int_{\mathbf{E}_i} \boldsymbol{\beta}_i\, p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\mathbf{E}_i)\, p(\boldsymbol{\beta}_i \mid \mathbf{z}_i, \mathbf{hr}_i)\, d\mathbf{E}_i\, d\boldsymbol{\beta}_i}{\int_{\boldsymbol{\beta}_i}\int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\mathbf{E}_i)\, p(\boldsymbol{\beta}_i \mid \mathbf{z}_i, \mathbf{hr}_i)\, d\mathbf{E}_i\, d\boldsymbol{\beta}_i}.$$

The reordering of terms to obtain the second expression is permissible because $\mathbf{E}_i$ and $\boldsymbol{\beta}_i$ are independent. Moreover, since they are independent, their joint distribution equals the product of the marginal distributions, so we may rewrite the preceding in a more useful form as

$$E(\boldsymbol{\beta}_i \mid \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, \mathbf{he}_i, \mathbf{hr}_i) = \frac{\int_{\boldsymbol{\beta}_i} \int_{\mathbf{E}_i} \boldsymbol{\beta}_i \, p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i) \, p(\boldsymbol{\beta}_i, \mathbf{E}_i \mid \mathbf{z}_i, \mathbf{hr}_i) d\mathbf{E}_i d\boldsymbol{\beta}_i}{\int_{\boldsymbol{\beta}_i} \int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i) \, p(\boldsymbol{\beta}_i, \mathbf{E}_i \mid \mathbf{z}_i, \mathbf{hr}_i) d\mathbf{E}_i d\boldsymbol{\beta}_i}.$$

This would provide the basis of the conditional estimator. Note that it is precisely the form of the posterior mean if this were a Bayesian application.

The integrals in the conditional mean for $\boldsymbol{\beta}_i$ will not exist in closed form, so some other method must be used to do the integration. Note, first, that in the expression above, the term $p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)$ is the contribution to the conditional likelihood function (not its log) of individual *i*, $L(\textbf{parameters} \mid \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, \mathbf{he}_i, \mathbf{hr}_i)$, and the integral is the unconditional likelihood. Second, integration over the range of $(\boldsymbol{\beta}_i, \mathbf{E}_i)$ with weighting function equal to the joint marginal density of $\boldsymbol{\beta}_i$ and $\mathbf{E}_i$ can be done by simulation. The implication is that the preceding integrals can be approximated using the simulation method used to maximize the simulated likelihood. Combining our results, we have the simulation based conditional estimator

$$\hat{E}(\boldsymbol{\beta}_i \mid \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, \mathbf{he}_i, \mathbf{hr}_i) = \frac{\frac{1}{R}\sum_{r=1}^{R} \hat{\boldsymbol{\beta}}_{ir} \, p(\mathbf{y}_i \mid \hat{\boldsymbol{\beta}}_{ir}, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_{ir})}{\frac{1}{R}\sum_{r=1}^{R} p(\mathbf{y}_i \mid \hat{\boldsymbol{\beta}}_{ir}, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_{ir})},$$

where

$$\hat{\boldsymbol{\beta}}_{ir} = \hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\Delta}}\mathbf{z}_i + \hat{\boldsymbol{\Gamma}}\hat{\boldsymbol{\Omega}}_i \mathbf{v}_{ir},$$

$$\hat{\boldsymbol{\Omega}}_i = diag[\exp(\hat{\boldsymbol{\omega}}_k' \mathbf{hr}_i)],$$

$$p(\mathbf{y}_i \mid \hat{\boldsymbol{\beta}}_{ir}, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_{ir}) = \prod_{t=1}^{T_i} \frac{\exp\left[\hat{\alpha}_{jir} + \hat{\boldsymbol{\beta}}_{ir}' \mathbf{x}_{jit} + \boldsymbol{\Sigma}_{m=1}^{M} d_{jm} \hat{\theta}_m \exp(\hat{\boldsymbol{\gamma}}_m' \mathbf{he}_i) E_{im,r}\right]}{\sum_{q=1}^{J_i} \exp\left[\hat{\alpha}_{qir} + \hat{\boldsymbol{\beta}}_{ir}' \mathbf{x}_{qit} + \boldsymbol{\Sigma}_{m=1}^{M} d_{qm} \hat{\theta}_m \exp(\hat{\boldsymbol{\gamma}}_m' \mathbf{he}_i) E_{im,r}\right]}.$$

The simulation over $(\boldsymbol{\beta}_i, \mathbf{E}_i)$ is actually a simulation over the structural random components, $\mathbf{v}_i$ and $\mathbf{E}_i$. The preceding shows how to do the simulation once the maximum likelihood estimates of the structural parameters, $[\boldsymbol{\beta}, \boldsymbol{\Delta}, \boldsymbol{\Gamma}, \boldsymbol{\Omega}, \boldsymbol{\theta}, \boldsymbol{\gamma}]$, are in hand. A final representation of the results is useful;

$$\hat{E}(\boldsymbol{\beta}_i \mid \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, \mathbf{he}_i, \mathbf{hr}_i) = \sum_{r=1}^{R} \hat{w}_{ir} \hat{\boldsymbol{\beta}}_{ir}$$

where

$$\hat{w}_{ir} = \frac{L(\mathbf{y}_i \mid \hat{\boldsymbol{\beta}}_{ir}, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_{ir}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\gamma}})}{\Sigma_{r=1}^{R} L(\mathbf{y}_i \mid \hat{\boldsymbol{\beta}}_{ir}, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_{ir}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\gamma}})}$$

and $L(\mathbf{y}_i \mid \hat{\boldsymbol{\beta}}_{ir}, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_{ir}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\gamma}})$ is the likelihood function for individual *i* computed at the maximum simulated likelihood estimates of all the parameters, the individual's own data, and the *r*th simulated draw on $(\mathbf{v}_i, \mathbf{E}_i)$

The preceding shows how *NLOGIT* simulates 'estimates' of $\boldsymbol{\beta}_i$. These form the inputs for the computation of elasticities and partial effects. There is a parameter vector computed for each individual in the sample. If you include **; Parameters** in the **RPLOGIT** command, *NLOGIT* creates the matrix named *beta_i* that contains these estimates. In the preceding, any nonrandom parameter is simply identically reproduced. As such, *beta_i* contains only the conditional means for the random parameters in the model.

Whether this estimator, $\hat{E}(\boldsymbol{\beta}_i \mid \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, \mathbf{he}_i, \mathbf{hr}_i) = \sum_{r=1}^{R} \hat{w}_{ir}\hat{\boldsymbol{\beta}}_{ir}$ is an *estimator* of $\boldsymbol{\beta}_i$ is subject to interpretation. The vector $\boldsymbol{\beta}_i$ is a draw from a distribution that has an unconditional mean,

$$E[\boldsymbol{\beta}_i|\mathbf{z}_i,\mathbf{hr}_i] \;=\; \boldsymbol{\beta} \;+\; \boldsymbol{\Delta}\mathbf{z}_i$$

and a conditional mean

$$E(\boldsymbol{\beta}_i \mid \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, \mathbf{he}_i, \mathbf{hr}_i) = \frac{\int_{\boldsymbol{\beta}_i}\int_{\mathbf{E}_i} \boldsymbol{\beta}_i\, p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\boldsymbol{\beta}_i, \mathbf{E}_i \mid \mathbf{z}_i, \mathbf{hr}_i)\, d\mathbf{E}_i\, d\boldsymbol{\beta}_i}{\int_{\boldsymbol{\beta}_i}\int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\boldsymbol{\beta}_i, \mathbf{E}_i \mid \mathbf{z}_i, \mathbf{hr}_i)\, d\mathbf{E}_i\, d\boldsymbol{\beta}_i}.$$

What we are computing here are estimates of the means of these distributions. In principle, these are conditioned on the particular data sets associated with individual *i*, not individual *i* themselves as such. To underscore the point, note that the computations would produce the same predictions for two individuals, say *i* and *i′*, if they have the same measured data, even though they would have different draws from the underlying population, $(\mathbf{v}_i,\mathbf{E}_i)$ and $(\mathbf{v}_{i'},\mathbf{E}_{i'})$. So, the mean computed here is an estimate of the center of this distribution, not a formal estimator of $\boldsymbol{\beta}_i$ as such.

We can take this a step further and examine the unconditional and conditional distributions. The variance of the unconditional distribution is

$$Var[\boldsymbol{\beta}_i|\mathbf{z}_i,\mathbf{hr}_i] \;=\; \boldsymbol{\Gamma}\boldsymbol{\Omega}_i^2\boldsymbol{\Gamma}'$$

for a particular element of $\boldsymbol{\beta}_i$, the variance is

$$Var[\beta_{ik}] \;=\; [\exp(\hat{\boldsymbol{\omega}}_k'\mathbf{hr}_i)]^2 \times \boldsymbol{\Sigma}_{s=1}^{k}\boldsymbol{\Gamma}_{sk}^2\,.$$

For the conditional distribution, no such expression exists. For a particular element of β_i,

$$Var(\beta_{ik} \mid \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, \mathbf{he}_i, \mathbf{hr}_i) = \frac{\int_{\boldsymbol{\beta}_i}\int_{\mathbf{E}_i} \beta_{ik}^2\, p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\boldsymbol{\beta}_i, \mathbf{E}_i \mid \mathbf{z}_i, \mathbf{hr}_i)\, d\mathbf{E}_i\, d\boldsymbol{\beta}_i}{\int_{\boldsymbol{\beta}_i}\int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\boldsymbol{\beta}_i, \mathbf{E}_i \mid \mathbf{z}_i, \mathbf{hr}_i)\, d\mathbf{E}_i\, d\boldsymbol{\beta}_i}$$

$$- \left[\frac{\int_{\boldsymbol{\beta}_i}\int_{\mathbf{E}_i} \beta_{ik}\, p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\boldsymbol{\beta}_i, \mathbf{E}_i \mid \mathbf{z}_i, \mathbf{hr}_i)\, d\mathbf{E}_i\, d\boldsymbol{\beta}_i}{\int_{\boldsymbol{\beta}_i}\int_{\mathbf{E}_i} p(\mathbf{y}_i \mid \boldsymbol{\beta}_i, \mathbf{x}_i, \mathbf{he}_i, \mathbf{E}_i)\, p(\boldsymbol{\beta}_i, \mathbf{E}_i \mid \mathbf{z}_i, \mathbf{hr}_i)\, d\mathbf{E}_i\, d\boldsymbol{\beta}_i}\right]^2.$$

The second term is the square of the mean that was estimated earlier. The first is the expected square, which can, like the mean, be estimated by simulation. Combining the results already obtained, then, we have an estimator of the conditional variance,

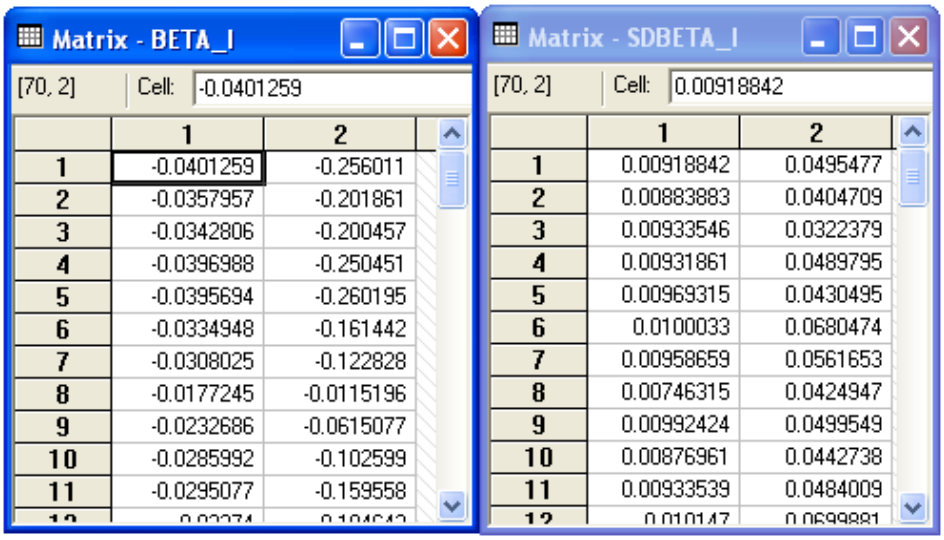
$$\hat{Var}(\boldsymbol{\beta}_i \mid \mathbf{y}_i, \mathbf{x}_i, \mathbf{z}_i, \mathbf{he}_i, \mathbf{hr}_i) = \sum_{r=1}^{R} \hat{w}_{ir} (\hat{\beta}_{ir,k})^2 - \left[\sum_{r=1}^{R} \hat{w}_{ir} \hat{\beta}_{ir,k} \right]^2 .$$

The square root of this quantity provides an estimate, for individual *i*, for each random parameter, an estimate of the conditional standard deviation. These diagonal elements appear in the matrix *sdbeta_i*.

We illustrate this with a model that includes most of the features described above:

```
RPLOGIT     ; Lhs = mode ; Choices = air,train,bus,car
            ; Rhs = gc,ttme ; Rh2 = one
            ; ECM = (air,car),(train,bus)
            ; RPL = hinc
            ; Fcn = gc(n),ttme(n) ; Correlated
            ; Parameters ; Halton ; Pds = 3 ; Pts = 200 $
```

(Model results are omitted.) The elements in the matrices are shown in Figure 10.7. As shown there, there is a considerable amount of variation in the estimated conditional means.



Matrix - BETA_I

[70, 2] Cell: -0.0401259

	1	2
1	-0.0401259	-0.256011
2	-0.0357957	-0.201861
3	-0.0342806	-0.200457
4	-0.0396988	-0.250451
5	-0.0395694	-0.260195
6	-0.0334948	-0.161442
7	-0.0308025	-0.122828
8	-0.0177245	-0.0115196
9	-0.0232686	-0.0615077
10	-0.0285992	-0.102599
11	-0.0295077	-0.159558

Matrix - SDBETA_I

[70, 2] Cell: 0.00918842

	1	2
1	0.00918842	0.0495477
2	0.00883883	0.0404709
3	0.00933546	0.0322379
4	0.00931861	0.0489795
5	0.00969315	0.0430495
6	0.0100033	0.0680474
7	0.00958659	0.0561653
8	0.00746315	0.0424947
9	0.00992424	0.0499549
10	0.00876961	0.0442738
11	0.00933539	0.0484009

Figure 10.7 Conditional Means and Standard Deviations

10.7.2 Examining the Distribution of the Parameters

The structural parameters often give a misleading picture of the parameters in a model. Consider the following modification of the model estimated in the previous section: We are going to fit the model as above, but change the distribution of the random parameters from normal to Weibull. The Weibull model forces parameters to be positive, so we also reverse the signs on the two attributes in the model.

```
CREATE     ; mgc  = -gc ; mttme = -ttme $
RPLOGIT    ; Lhs  = mode ; Choices = air,train,bus,car
           ; Rhs  = mgc,mttme ; Rh2 = one
           ; ECM = (air,car),(train,bus)
           ; RPL = hinc
           ; Parameters ; Halton ; Pds = 3 ; Pts = 200
           ; Fcn = mgc(n),mttme(n) ; Correlated $
MATRIX     ; bn = beta_i ; sn = sdbeta_i $
```

The estimation and analysis is repeated with the Weibull distribution.

```
           ; Fcn = mgc(w),ttme(w) ; Correlated $
MATRIX     ; bw = beta_i ; sw = sdbeta_i $
```

The unconditional values in the first column of the matrix in Figure 10.7 and the nonstochastic estimates for the MNL model should suggest the likely values of the two random parameters. However, it would be difficult to deduce this from the estimated structural parameters for the Weibull model, which are completely different. The Weibull distribution, which involves the exponent of $\boldsymbol{\beta} + \boldsymbol{\Delta}\mathbf{z}_i + \boldsymbol{\Gamma}\,\boldsymbol{\Omega}_i\,\mathbf{v}_i$, looks quite different from the normal.

These are the basic MNL estimates, with both parameters fixed.

```
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 MGC     |     .01578374       .00438279     3.601   .0003
 MTTME   |     .09709052       .01043509     9.304   .0000
 A_AIR   |    5.77635901       .65591873     8.807   .0000
 A_TRAIN |    3.92300113       .44199360     8.876   .0000
 A_BUS   |    3.21073472       .44965283     7.140   .0000
```

This is the same model, with two correlated normally distributed random parameters with heterogeneous means. There are also two random error components in the model.

```
---------+Random parameters in utility functions
 MGC     |     .03170589       .01949180     1.627   .1038
 MTTME   |     .13551247       .02907461     4.661   .0000
---------+Nonrandom parameters in utility functions
 A_AIR   |    10.1292509      1.85832856     5.451   .0000
 A_TRAIN |    8.20598683      1.73422590     4.732   .0000
 A_BUS   |    7.19813304      1.91386320     3.761   .0002
```

```
---------+Heterogeneity in mean, Parameter:Variable
 MGC:HIN |     .450634D-05      .00044082      .010   .9918
 MTTM:HIN|      .00078233       .00056928     1.374   .1694
---------+Diagonal values in Cholesky matrix, L.
 NsMGC   |      .01138907       .02143530      .531   .5952
 NsMTTME |      .06637718       .07932744      .837   .4027
---------+Below diagonal values in L matrix. V = L*Lt
 MTTM:MGC|      .05922416       .09092201      .651   .5148
---------+Standard deviations of latent random effects
 SigmaE01|    1.44001740      3.62060512      .398   .6908
 SigmaE02|    1.70126558      2.89949978      .587   .5574
---------+Standard deviations of parameter distributions
 sdMGC   |      .01138907       .02143530      .531   .5952
 sdMTTME |      .08895746       .02886146     3.082   .0021
```

This is the same model once again, now with Weibull distributed parameters.

```
---------+Random parameters in utility functions
 MGC     |   -2.84950808       .77609997    -3.672   .0002
 MTTME   |   -1.31927880      1.29537623    -1.018   .3085
---------+Nonrandom parameters in utility functions
 A_AIR   |    10.1003905      1.80393746     5.599   .0000
 A_TRAIN |    8.04274180      1.64404555     4.892   .0000
 A_BUS   |    7.04388422      1.83042969     3.848   .0001
---------+Heterogeneity in mean, Parameter:Variable
 MGC:HIN |    -.00573905       .01801325     -.319   .7500
 MTTM:HIN|      .00675433       .00355598     1.899   .0575
---------+Diagonal values in Cholesky matrix, L.
 WsMGC   |      .24147776       .37415604      .645   .5187
 WsMTTME |      .00503652       .68300577      .007   .9941
---------+Below diagonal values in L matrix. V = L*Lt
 MTTM:MGC|    -.49268390       .11909663    -4.137   .0000
---------+Standard deviations of latent random effects
 SigmaE01|      .99017999      6.82984761      .145   .8847
 SigmaE02|    2.08605480      3.40861849      .612   .5405
---------+Standard deviations of parameter distributions
 sdMGC   |      .24147776       .37415604      .645   .5187
 sdMTTME |      .49270964       .12066246     4.083   .0000
```

The ASCs in the three models resemble one another, but the coefficients on the attributes are vastly different, and would seem to suggest very different models. In fact, that is not the case, as we now examine. In order to compare these sets of estimates, we propose to examine the estimated conditional means. We will use two devices. A direct approach is to examine the distribution of estimates of $E[\boldsymbol{\beta}_i|*]$ across the observations in the sample. The averages of the conditional means will estimate the population mean (averaged across $\mathbf{z}_i$ as well). The variances require a bit of manipulation, since as noted, the variance of the conditional means underestimates the overall variance (by the mean of the conditional variances). We will also examine the distribution of conditional means in the sample with a kernel density estimator.

First estimate the models. The parameter estimates are shown above.

```
SAMPLE      ; All $
CREATE      ; mgc = -gc ; mttme = -ttme $
CLOGIT      ; Lhs  = mode ; Choices = air,train,bus,car
            ; Rhs  = mgc,mttme ; Rh2 = one $
CALC        ; bgmnl = b(1) ; btmnl = b(2) $
RPLOGIT     ; Lhs  = mode ; Choices = air,train,bus,car
            ; Rhs  = mgc,mttme ; Rh2 = one
            ; ECM = (air,car),(train,bus) ; RPL = hinc
            ; Parameters ; Halton ; Pds = 3 ; Pts = 200
            ; Fcn  = mgc(n),mttme(n) ; Correlated $
MATRIX      ; bn = beta_i ; sn = sdbeta_i $
RPLOGIT     ; Lhs  = mode ; Choices = air,train,bus,car
            ; Rhs  = mgc,mttme ; Rh2 = one
            ; ECM = (air,car),(train,bus) ; RPL = hinc
            ; Parameters ; Halton ; Pds = 3 ; Pts = 200
            ; Fcn  = mgc(w),mttme(w) ; Correlated $
MATRIX      ; bw = beta_i ; sw = sdbeta_i $
```

Now, move the matrices to the data area so we can examine them.

```
SAMPLE      ; 1 - 70 $
CREATE      ; bgn  = 0 ; btn = 0 ; bgw = 0 ; btw= 0  $
CREATE      ; sgn  = 0 ; stn = 0  ; sgw = 0  ; stw = 0  $
NAMELIST    ; betan = bgn,btn  ; betaw = bgw,btw   $
NAMELIST    ; sbetan = sgn,stn ; sbetaw = sgw,stw   $
CREATE      ; betan   = bn  $
CREATE      ; betaw  = bw  $
CREATE      ; sbetan = sn   $
CREATE      ; sbetaw = sw  $
```

Now compare the different estimates. The results below show that the normal and Weibull coefficients are much more similar than the raw parameter estimates would suggest. We first estimate the population means by averaging the conditional means.

```
CALC        ; List ; bgmnl ; Xbr(bgn) ; Xbr(bgw) $
CALC        ; List ; btmnl  ; Xbr(btn) ; Xbr(btw)$
```

These are the three estimates of $E[\beta_{gc}]$

```
BGMNL   =       .015784
Result  =       .031987 ←        (Normally distributed)
Result  =       .031688          (Weibull distributed)
```

These are the three estimates of $E[\beta_{ttme}]$

```
BTMNL   =       .097091
Result  =       .166242 ←       (Normally distributed)
Result  =       .169459         (Weibull distributed)
```

Are the correlations the same? Note these are the correlations of the conditional means, not the correlations of the coefficients.

```
CALC        ; List ; Cor(bgn,btn) ; Cor(bgw,btw) $
```

```
Result  =       .962738         (Two normally distributed parameters)
Result  =       .723847         (Two Weibull distributed parameters)
```

What about the population standard deviations? The following estimate the standard deviations of the population marginal distribution of the two parameters. Once again, the similarity is striking given the quite large differences in the estimates of the structural parameters.

```
CREATE      ; vbgn = sgn^2 ; vbtn = stn^2 ; vbgw = sgw^2 ; vbtw = stw^2 $
CALC        ; List  ; sdbgn  = Sqr(xbr(vbgn) + Var(bgn))
                    ; sdbgw = Sqr(xbr(vbgw)+ Var(bgw))
                    ; sdbtn  = Sqr(xbr(vbtn)  + Var(btn))
                    ; sdbtw  = Sqr(xbr(vbtw) + Var(btw)) $
```

```
SDBGN   =       .011456
SDBGW   =       .009629
SDBTN   =       .089567
SDBTW   =       .088368
```

A final comparison is based on the kernel density estimators for the distributions of the conditional means. Only the two for β_{gc} are shown.

```
KERNEL      ; Rhs = bgn  ; Title = Kernel Density for E[b_gc|*,normal]
            ; Endpoints = .01,.05 $
KERNEL      ; Rhs = bgw ; Title = Kernel Density for E[b_gc|*,Weibull]
            ; Endpoints = .01,.05 $
KERNEL      ; Rhs = btn  ; Title = Kernel Density for E[b_ttme|*,normal]$
KERNEL      ; Rhs = btw  ; Title = Kernel Density for E[b_ttme|*,Weibull]$
```

Based on the results obtained thus far, it seems that the impact of the Weibull specification is to increase the variance of the empirical distribution.

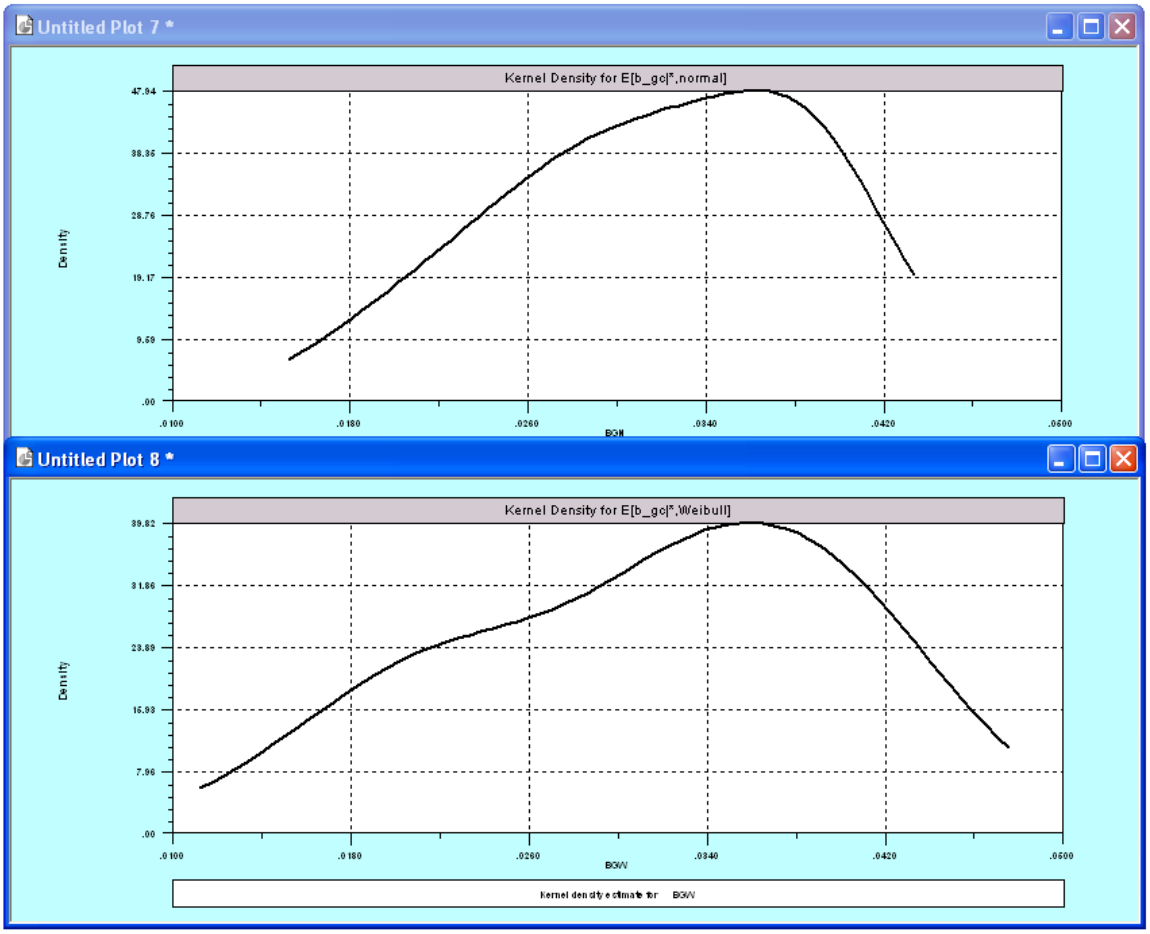


Figure 10.8 Kernel Densities for Parameter Distributions

10.7.3 Conditional Confidence Intervals for Parameters

Finally, we consider an alternative approach to examining the distribution of parameters across individuals. We have for each individual, an estimate of the mean of the conditional distribution of parameters from which their specific vector is drawn. This is the estimate of $E[\boldsymbol{\beta}_i|i]$ that is in row i of *beta_i*. We also have an estimate of the standard deviation of this conditional distribution. As a general result, an interval in a distribution for a continuous random variable defined by the mean plus and minus two standard deviations will encompass 95% or more of the mass of the distribution. This enables us to form a sort of confidence interval for $\boldsymbol{\beta}_i$ itself, conditioned on all the information known about the individual. To roughly this level of confidence, the interval

$$E[\beta_{ik}|\text{all information on individual } i] \pm 2\times\text{SD}[\beta_{ik}|\text{all information on individual } i]$$

will contain the actual draw for individual i. (The probability is somewhat reduced because we are using estimates of the structural parameters, not the true values.) The centipede plot feature of **PLOT** allows us to produce this figure, as follows: We plot the figure for β_{gc} for the Weibull model:

The commands are:

```
CREATE    ; lowerbgc = bgw - 2*sgw ; upperbgc = bgw + 2*sgw $
CREATE    ; person = Trn(1,1) $
CALC      ; meanbgw = Xbr(bgw) $
CALC      ; highbgw = meanbgw + 2*sdbgw $
CALC      ; lowbgw = meanbgw - 2*sdbgw $
PLOT      ; Lhs = person ; Rhs = lowerbgc,upperbgc
          ; Centipede
          ; Title = Confidence Limits for b_gc for Weibull Model
          ; Bars = meanbgw,highbgw,lowbgw
          ; Endpoints = 0,75 $
```

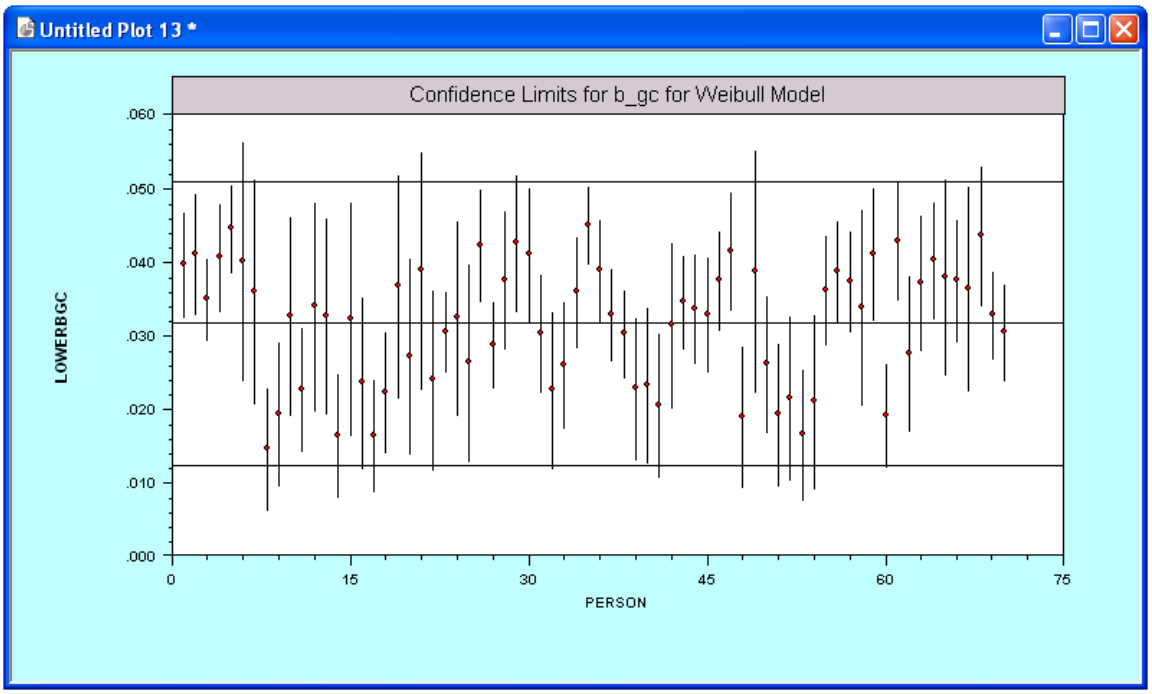


Figure 10.9 Conditional and Unconditional Distributions of Parameters

In the figure, each vertical 'leg' of the centipede plot shows the conditional confidence interval for β_{gc} for that person. The dot is the midpoint of the interval, which is the point estimate. The center horizontal bar in the figure shows the mean of the conditional means, which estimates the population mean. This was reported earlier as 0.031688. The upper and lower horizontal bars show the overall mean plus and minus twice the estimated population standard deviation – this was reported earlier as 0.009629. Thus, the unconditional population range of variation is estimated to be about .01 to .05. Note that this is the range of variation in the kernel density estimates given in Figure 10.8. Figure 10.9 demonstrates clearly how the additional information for each individual is used to reduce the 'uncertainty' about the individual specific estimates.

10.7.4 Willingness to Pay Estimates

The previous section showed how to estimate a function of the random (or nonrandom) parameters using the simulation method. We estimated the conditional variance using a simulation based estimator of $E[\beta_i^2|\text{all information on individual } i]$. Another useful function of the parameters in the model is the 'willingness to pay function.' This is typically measured using

WTP = attribute coefficient / income or price coefficient

The random parameters logit model will compute and retain person specific WTP measures. Use

; WTP = name/name

where names are either variable names if **; Rhs** is used or parameter names if utility functions are specified directly. In general, the WTP calculation will have an attribute level coefficient in the numerator and a cost or income measure in the denominator. Parameters can be random or nonrandom. This will create two matrices, *wtp_i* and *sdwtp_i*. These are computed the same way that *beta_i* and *sdbeta_i* are computed, where *wtp_i* contains estimates of the conditional expectation of WTP and *sdwtp_i* contains estimates of the conditional standard deviation. These matrices can be examined and analyzed in precisely the same way that *beta_i* was used earlier. You may compute more than one WTP variable by adding additional ratios in the command separated by commas. For example,

; WTP = time/income, space/price

To illustrate, we use the Weibull model once again, with a small modification:

```
RPLOGIT     ; Lhs = mode ; Choices = air,train,bus,car
            ; Rhs = mgc,mttme,hinca ; Rh2 = one
            ; ECM = (air,car),(train,bus)
            ; WTP = mttme/hinca
            ; Fcn = mgc(w),mttme(w) ; Correlated
            ; Parameters ; Halton ; Pds = 3 ; Pts = 200 $
```

The willingness to pay is computed as the ratio of the terminal time in minutes to the income variable, *hinca* – this equals income for the air alternative and zero otherwise. The basic coefficient estimates are

```
---------+Random parameters in utility functions
 MGC     |   -3.10624315       .69007784    -4.501   .0000
 MTTME   |   -1.22068334      1.01138340    -1.207   .2275
---------+Nonrandom parameters in utility functions
 HINCA   |     .02916424       .02180170     1.338   .1810
 A_AIR   |    8.30401343      1.72839556     4.804   .0000
 A_TRAIN |    7.44116617      1.45457284     5.116   .0000
 A_BUS   |    6.50022714      1.60098502     4.060   .0000
```

As before, the structural parameters do not suggest what the implied parameters will look like. For these data, the estimated WTP values for the first 10 individuals (copied from *wtp_i*) are

8.048, 7.11862, 6.41581, 8.01403, 8.31522, 5.56074, 4.42096, 1.58768, 2.36362, 3.1795,

The overall average computed by averaging the 70 values in the matrix is 5.23031. This is in $/minute.

10.8 Applications

The preceding sections contain numerous examples of the mixed logit model. The applications below show a few of the most basic procedures. This is a basic formulation with two random parameters and heterogeneity in the means as a function of household income.

```
RPLOGIT     ; Lhs  = mode
            ; Choices = air,train,bus,car
            ; Rhs  = gc,ttme
            ; Rh2  = one
            ; RPL = hinc
            ; Fcn  = gc(n),ttme(n)
            ; Effects: gc(air) $
```

```
+---------------------------------------------+
| Discrete choice and multinomial logit models|
+---------------------------------------------+
+---------------------------------------------+
| Start values obtained using MNL model       |
| Maximum Likelihood Estimates                |
| Dependent variable               Choice     |
| Number of observations              210     |
| Log likelihood function       -199.9766     |
| Info. Criterion: AIC =          1.95216     |
|   Finite Sample: AIC =          1.95356     |
| Info. Criterion: BIC =          2.03185     |
| Info. Criterion:HQIC =          1.98438     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -283.7588  .29526  .28504 |
| Chi-squared[ 2]          =    167.56429    |
| Prob [ chi squared > value ] =   .00000     |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
+---------------------------------------------+
| Notes No coefficients=> P(i,j)=1/J(i).      |
|       Constants only => P(i,j) uses ASCs    |
|         only. N(j)/N if fixed choice set.   |
|         N(j) = total sample frequency for j |
|         N    = total sample frequency.      |
|       These 2 models are simple MNL models. |
|       R-sqrd = 1 - LogL(model)/logL(other)  |
|       RsqAdj=1-[nJ/(nJ-nparm)]*(1-R-sqrd)   |
|         nJ   = sum over i, choice set sizes |
+---------------------------------------------+
```

```
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 GC      |    -.01578374       .00438279    -3.601   .0003
 TTME    |    -.09709052       .01043509    -9.304   .0000
 A_AIR   |    5.77635901       .65591873     8.807   .0000
 A_TRAIN |    3.92300113       .44199360     8.876   .0000
 A_BUS   |    3.21073472       .44965283     7.140   .0000

+---------------------------------------------+
| Random Parameters Logit Model               |
| Dependent variable                  MODE    |
| Number of observations               210    |
| Log likelihood function       -182.9290     |
| Number of parameters                  9     |
| Info. Criterion: AIC =          1.82789     |
|   Finite Sample: AIC =          1.83218     |
| Info. Criterion: BIC =          1.97134     |
| Info. Criterion:HQIC =          1.88589     |
| Restricted log likelihood     -291.1218     |
| McFadden Pseudo R-squared      .3716412     |
| Chi squared                    216.3857     |
| Degrees of freedom                    9     |
| Prob[ChiSqd > value] =         .0000000     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| No coefficients   -291.1218  .37164  .36253 |
| Constants only    -283.7588  .35534  .34599 |
| At start values   -199.9766  .08525  .07199 |
| Response data are given as ind. choice.     |
| Replications for simulated probs. = 500     |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
---------+Random parameters in utility functions
 GC      |    -.01871422       .01712611    -1.093   .2745
 TTME    |    -.17600015       .04467395    -3.940   .0001
---------+Nonrandom parameters in utility functions
 A_AIR   |    11.0829925      2.37916582     4.658   .0000
 A_TRAIN |    9.22867193      2.20639245     4.183   .0000
 A_BUS   |    8.19884828      2.10499796     3.895   .0001
---------+Heterogeneity in mean, Parameter:Variable
 GC:HIN  |    -.00029029       .00036724     -.790   .4293
 TTME:HIN|    -.00060674       .00061444     -.987   .3234
---------+Derived standard deviations of parameter distributions
 NsGC    |     .01364904       .02015496      .677   .4983
 NsTTME  |     .11712864       .03885999     3.014   .0026

Parameter Matrix for Heterogeneity in Means.

Matrix Delta    has  2 rows and  1 columns.
         HINC
        +--------------
GC      |    -.00029
TTME    |    -.00061
```

```
+-----------------------------------------------------+
| Elasticity             averaged over observations.|
| Attribute is GC       in choice AIR                |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.     |
|                                  Mean    St.Dev    |
| *     Choice=AIR               -.8246     .4614    |
|       Choice=TRAIN              .7617     .7464    |
|       Choice=BUS               1.0439     .9282    |
|       Choice=CAR                .2897     .6635    |
+-----------------------------------------------------+
```

This is a two level hierarchical model. There are no random parameters, but the coefficients on *gc* and *ttme* are modeled as linear functions of a constant and household income.

```
RPLOGIT     ; Lhs  = mode
            ; Choices = air,train,bus,car
            ; Rhs  = gc,ttme
            ; Rh2  = one
            ; RPL = hinc
            ; Fcn  = gc(c),ttme(c) $
```

```
Normal exit from iterations. Exit status=0.
+---------------------------------------------+
| Random Parameters Logit Model               |
| Dependent variable                 MODE     |
| Number of observations              210     |
| Log likelihood function       -198.3960     |
| Info. Criterion: AIC =          1.95615     |
|   Finite Sample: AIC =          1.95879     |
| Info. Criterion: BIC =          2.06772     |
| Info. Criterion:HQIC =          2.00126     |
| Restricted log likelihood     -291.1218     |
| McFadden Pseudo R-squared      .3185122     |
| Chi squared                    185.4517     |
| Degrees of freedom                    7     |
| Prob[ChiSqd > value] =         .0000000     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| No coefficients   -291.1218  .31851  .31086 |
| Constants only    -283.7588  .30083  .29297 |
| At start values   -199.9766  .00790 -.00324 |
| Response data are given as ind. choice.     |
| Replications for simulated probs. = 500     |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
---------+Random parameters in utility functions
 GC      |    -.01139658       .00920685    -1.238   .2158
 TTME    |    -.08786478       .01174507    -7.481   .0000
---------+Nonrandom parameters in utility functions
 A_AIR   |    5.84415090       .65860452     8.874   .0000
 A_TRAIN |    3.96545510       .44224936     8.967   .0000
 A_BUS   |    3.25638033       .45029696     7.232   .0000
```

```
---------+Heterogeneity in mean, Parameter:Variable
 GC:HIN  |    -.00010094       .00020965     -.481   .6302
 TTME:HIN|    -.00028132       .00017922    -1.570   .1165
---------+Derived standard deviations of parameter distributions
 CsGC    |       .000000    ......(Fixed Parameter).......
 CsTTME  |       .000000    ......(Fixed Parameter).......
Parameter Matrix for Heterogeneity in Means.
Matrix Delta    has  2 rows and  1 columns.
         HINC
        +--------------
GC      |    -.00010
TTME    |    -.00028
```

10.9 Panel Data

The random parameters model includes a treatment for panel data. Two forms are accommodated. For a simple clustering of T_i choice situations by the same individual, for example, a stated preference survey in which several different scenarios are offered, then a random effects type of treatment might be appropriate. For example, the sequencing of choices might be unknown. In this case, the usual random effects setup would apply

$$\boldsymbol{\beta}_{it} = \boldsymbol{\beta} + \boldsymbol{\Delta}\mathbf{z}_{it} + \boldsymbol{\Gamma}\mathbf{v}_i$$

where '*t*' indexes the multiple observations for individual '*i*.' The connection to 'time' might not hold here, but we use the same index regardless. Note that the heterogeneity in the mean may change from one observation to the next (or not, depending on your situation), but the random term, $\mathbf{v}_i$ is the same for all observations. As in all panel data situations in *NLOGIT*, the number of observations, T_i on individual *i* may vary by individual. An alternative situation might arise when choice situations are observed in sequence, and there is a long enough lag between situations that the effect of the passage of time might be to allow preferences to evolve – consider, for example, cases in which habit persistence influences the choice (mode of travel to work), but new information enters the system. In such a case, an autoregressive arrangement might be appropriate;

$$\boldsymbol{\beta}_{it} = \boldsymbol{\beta} + \boldsymbol{\Delta}\mathbf{z}_{it} + \boldsymbol{\Gamma}\mathbf{v}_{it}$$

$$\mathbf{v}_{it} = \mathbf{R}\mathbf{v}_{i,t-1} + \mathbf{u}_{it}$$

where $\mathbf{R}$ is a diagonal matrix of autocorrelation coefficients and $\mathbf{u}_{it}$ constitutes the primitive randomness in the system.

The two situations are requested by first specifying the panel as usual with

; Pds = Ti

where T_i is either a fixed number of observations or a variable which gives the number of observations. In this setting, the panel consists of groups of T_i sets of J_i observations. In all cases, T_i tells the number of *groups* of data. You may have a variable number of observations *and* a variable number of choices within a group or any of the other three possible combinations. In our examples below, $J = 4$ – a fixed number of choices. In one case, $T_i = 3$, so in this case, there are 12 rows of data for each person. In the other case, there are six observations in a group, so 24 rows of data per person. If the number of observations in a group varies, so T_i is the name of a count variable, this count is repeated on every row of data within an observation, and for every observation in the group.

The autoregressive model is requested by adding

; AR1

to the **NLOGIT** command. You may also constrain the autoregressive model with

; AR1 = list of values

where the list may contain symbols for free parameters or specific numerical values, including zero if you do not wish for specific coefficients to evolve in this fashion.

To illustrate the panel data models, we will artificially treat our clogit data as if it were a panel. (It is not.) For the first model, we collect the observations in groups of three, and treat it as a random effects model. For the second, we collect the observations in groups of six, and fit an AR1 model to them. Since these data are, in fact, a cross section, we should not expect much of the estimates.

Chapter 11: The Multinomial Probit Model

11.1 Introduction

In the multinomial probit (MNP) model, the individual's choice among J alternatives is the one with maximum utility, where the utility functions are

$$U_{ji} = \boldsymbol{\beta}'\mathbf{x}_{ji} + \varepsilon_{ji},$$

where U_{ji} = utility of alternative j to individual i,

$\mathbf{x}_{ji}$ = union of all attributes that appear in all utility functions. For some alternatives, $x_{i,tk}$ may be zero by construction for some attribute k which does not enter their utility function for alternative j,

ε_{ji} = unobserved heterogeneity for individual i and alternative j.

The multinomial logit model specifies that ε_{ji} are draws from independent extreme value distributions (which induces the IIA condition). In the multinomial probit model, we assume that ε_{ji} are normally distributed with standard deviations Sdv[ε_{ji}] = σ_j and correlations Cor[ε_{ji}, ε_{mi}] = ρ_{jm} (the same for all individuals). Observations are independent, so Cor[ε_{ji},ε_{ms}] = 0 if i is not equal to s, for all j and m. A variation of the model allows the standard deviations and covariances to be scaled by a function of the data, which allows some heteroscedasticity across individuals.

The correlations ρ_{jm} are restricted to $-1 < \rho_{jm} < 1$, but they are otherwise unrestricted save for a necessarily normalization. The correlations in the last row of the correlation matrix must be fixed at zero. The standard deviations are unrestricted with the exception of a normalization – two standard deviations are fixed at 1.0 – *NLOGIT* fixes the last two. In principle, up to 20 alternatives may be in the model, but our experience thus far is that this model is extremely difficult to estimate, and will usually not be estimable with a completely free correlation matrix even with only five alternatives. The difficulty increases greatly with the number of alternatives. (Imposition of constraints which may improve this situation is discussed below.)

This model may also be fit with panel data. In this case, the utility function is modified as follows:

$$U_{ji,t} = \boldsymbol{\beta}'\mathbf{x}_{jt,t} + \varepsilon_{ji,t} + v_{ji,t},$$

where 't' indexes the periods or replications. There are two formulations for $v_{ji,t}$,

Random effects $v_{ji,t} = v_{ji,s}$ (the same in all periods),

First order autoregressive $v_{ji,t} = \alpha_j v_{ji,t-1} + a_{ji,t}$.

11.2 Model Command

This is a one level (nonnested) model. The setup is identical to the multinomial logit model with one level. To request it, use

MNPROBIT ; Lhs = ... ; Choices = ...
; Rhs = ... or ; Model: U (...) =... / U (...) = ... all as usual
; ... any other options $

(The alternative model command used in earlier versions of *NLOGIT*, **NLOGIT ; MNP** is equivalent and may be used instead.)

Options include

; Prob = name to use for estimated probabilities
; Utility = name to use for estimated utilities

and the usual other options for output, technical output, elasticities, descriptive statistics, etc. (See Chapters 6 and 7 for details.) There are some special cases for this estimator:

- The number of alternatives must be fixed – it may not vary across observations.
- The choice set must be fixed.
- Choice based sampling is not supported, though you can use ordinary weights.
- Data may be individual, proportions, or frequencies.

(The second derivatives matrix is not computed for this model, so it is not possible to compute a robust covariance matrix estimator.) An additional option is

; Pts = number of replications to compute multivariate normal probabilities

The following features of *NLOGIT* are *not* available for this model:

; Tree ... This is not a nested logit model.
; Ivb = name, **; Ivl = name**, **; Ivt = name** No inclusive values are computed.
; IIA = list IIA is not testable here, since it is not imposed.
; Cprob = name Conditional and unconditional probabilities are the same.
; Ranks This estimator may not be based on ranks data.
; Scale ... Data scaling is only for the nested logit model.

The command builder may also be used for this model by selecting Model/Discrete Choice/Multinomial Probit, HEV, RPL. The choice set and utility functions for the model are defined on the Main page and the MNP format of the model is selected on the Options page. See Figures 11.1 and 11.2 for the setup of the model shown in the application below.

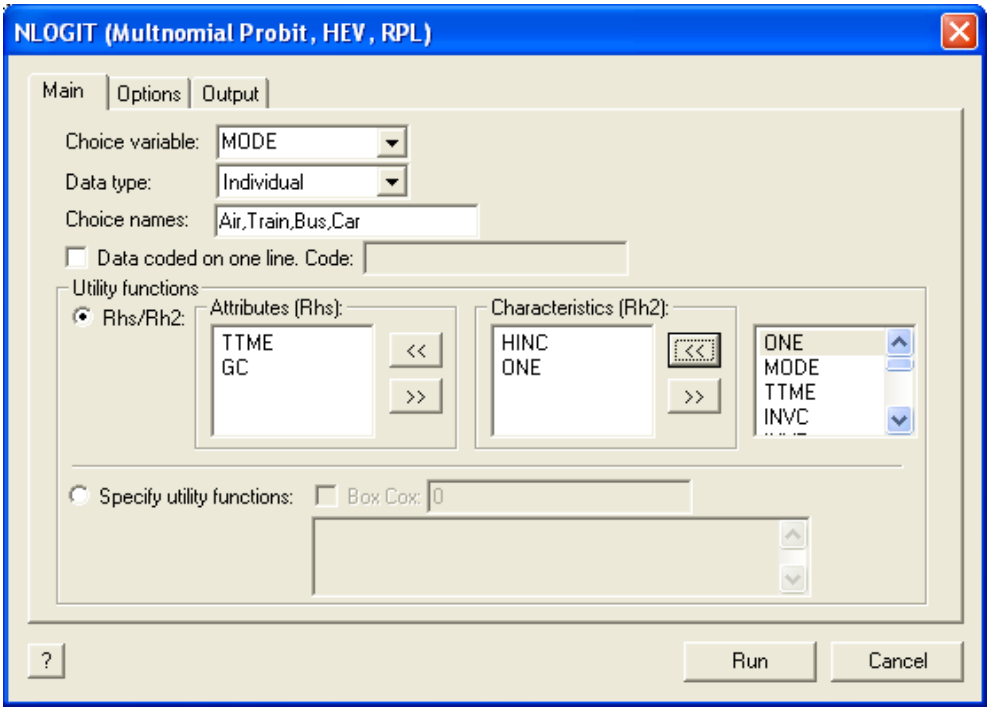


Figure 11.1 Main Page of Command Builder for the MNP model

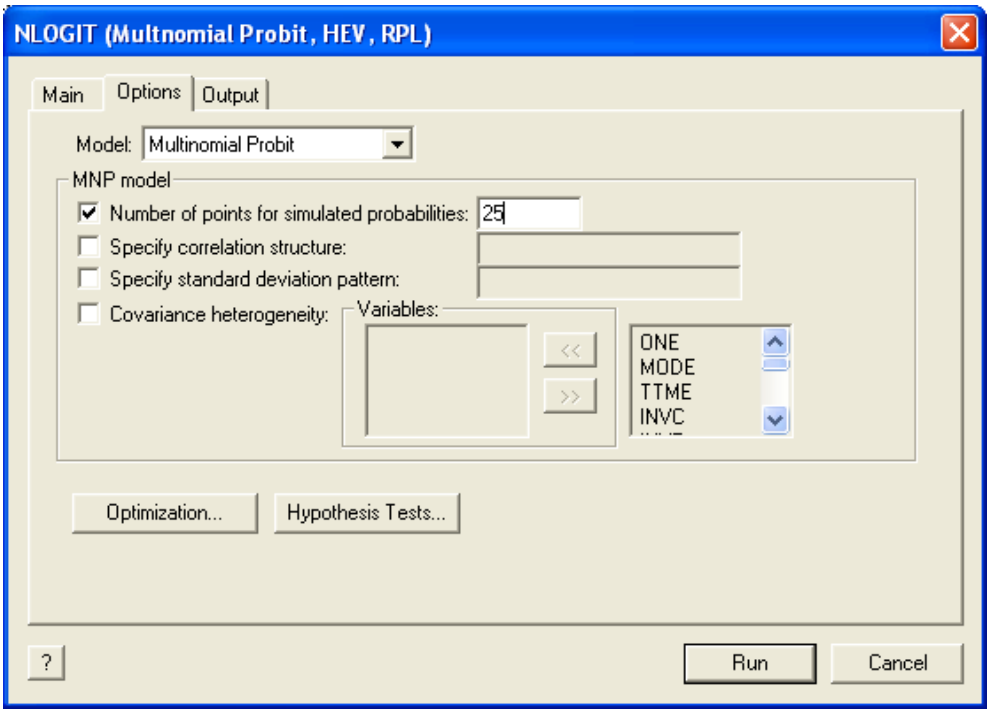


Figure 11.2 Options Page of Command Builder for the MNP model

11.3 An Application

The multinomial probit model based on the clogit data is estimated with the command

```
MNPROBIT   ; Lhs  = mode
           ; Choices = air,train,bus,car
           ; Rhs  = gc,ttme
           ; Rh2 = one,hinc
           ; Effects: gc(air)
           ; Pts   = 20 $
```

This is the model that was fit as an MNL model in Chapter 8. We have now relaxed the equal variances assumption and replaced the extreme value distribution with a multivariate normal distribution. The probabilities are computed with 20 replications, which is fairly small; we do this for purposes of a simple illustration. Results are shown below. The MNL model is fit first to obtain the starting values for the iterations. The results for the MNP model are given next. The two sets of results are merged in the display below.

```
+---------------------------------------------+
| Discrete choice (multinomial logit) model   |
| Dependent variable                 MODE     |
| Log likelihood function       -189.5252     |
| Info. Criterion: AIC =          1.88119     |
|   Finite Sample: AIC =          1.88460     |
| Info. Criterion: BIC =          2.00870     |
| Info. Criterion:HQIC =          1.93274     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| Constants only    -283.7588  .33209  .31802 |
| Chi-squared[ 5]          =    188.46723     |
| Prob [ chi squared > value ] =   .00000     |
| Response data are given as ind. choice.     |
| Number of obs.=   210, skipped   0 bad obs. |
+---------------------------------------------+

+---------------------------------------------+
| Multinomial Probit Model                    |
| Log likelihood function       -189.8452     |
| Info. Criterion: AIC =          1.93186     |
|   Finite Sample: AIC =          1.94070     |
| Info. Criterion: BIC =          2.13906     |
| Info. Criterion:HQIC =          2.01562     |
| Restricted log likelihood     -291.1218     |
| McFadden Pseudo R-squared      .3478840     |
| Chi squared                    202.5532     |
| Degrees of freedom                   13     |
| Prob[ChiSqd > value] =         .0000000     |
| R2=1-LogL/LogL*  Log-L fncn  R-sqrd  RsqAdj |
| No coefficients   -291.1218  .34788  .33414 |
| Constants only    -283.7588  .33096  .31687 |
| At start values   -216.5343  .12326  .10478 |
+---------------------------------------------+
```

These are the estimates for the multinomial logit model

```
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
 GC      |    -.01092735       .00458775    -2.382   .0172
 TTME    |    -.09546055       .01047320    -9.115   .0000
 A_AIR   |    5.87481336       .80209034     7.324   .0000
 AIR_HIN1|    -.00537349       .01152940     -.466   .6412
 A_TRAIN |    5.54985728       .64042443     8.666   .0000
 TRA_HIN2|    -.05656186       .01397335    -4.048   .0001
 A_BUS   |    4.13028388       .67636278     6.107   .0000
 BUS_HIN3|    -.02858418       .01544418    -1.851   .0642
```

These are the estimates for the multinomial probit model

```
+--------+--------------+----------------+--------+--------+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+
---------+Attributes in the Utility Functions (beta)
 GC      |    -.02333086       .00896463    -2.603   .0093
 TTME    |    -.09131236       .03629673    -2.516   .0119
 A_AIR   |    4.68057508      1.91530359     2.444   .0145
 AIR_HIN1|     .00832932       .02520384      .330   .7410
 A_TRAIN |    5.90782858      1.92699048     3.066   .0022
 TRA_HIN2|    -.06016958       .02223662    -2.706   .0068
 A_BUS   |    4.40097868      1.27339698     3.456   .0005
 BUS_HIN3|    -.01884772       .01615587    -1.167   .2434
---------+Std. Devs. of the Normal Distribution.
 s[AIR]  |    2.85536857      1.29978748     2.197   .0280
 s[TRAIN]|    1.96198515       .91344112     2.148   .0317
 s[BUS]  |    1.00000000    ......(Fixed Parameter).......
 s[CAR]  |    1.00000000    ......(Fixed Parameter).......
---------+Correlations in the Normal Distribution
 rAIR,TRA|     .12923578       .74351679      .174   .8620
 rAIR,BUS|     .11759913       .92452141      .127   .8988
 rTRA,BUS|     .61859572       .38300577     1.615   .1063
 rAIR,CAR|       .000000    ......(Fixed Parameter).......
 rTRA,CAR|       .000000    ......(Fixed Parameter).......
 rBUS,CAR|       .000000    ......(Fixed Parameter).......
```

The table below compares the elasticities from the MNP model to the MNL model. The MNL results appear first. They are clearly similar, but the specification does make a difference.

```
+-----------------------------------------------------+
| Elasticity             averaged over observations.|
| Attribute is GC       in choice AIR                 |
| Effects on probabilities of all choices in model: |
| * = Direct Elasticity effect of the attribute.    |
|                                   Mean    St.Dev   |
| *     Choice=AIR               -.8019     .3834   |
|       Choice=TRAIN              .3198     .3370   |
|       Choice=BUS                .3198     .3370   |
|       Choice=CAR                .3198     .3370   |
+-----------------------------------------------------+
| Effects on probabilities of all choices in model: |
| *     Choice=AIR              -1.0453     .4797   |
|       Choice=TRAIN              .3796     .3184   |
|       Choice=BUS                .5557     .3826   |
|       Choice=CAR                .4221     .2957   |
+-----------------------------------------------------+
```

11.4 Testing IIA with a Multinomial Probit Model

A multinomial probit model with all standard deviations equal to one and uncorrelated random terms specifies a model with the IIA property. This means that you could test this property by using an LR or LM test of the assumption that all of the standard deviations in a model *with uncorrelated disturbances* are equal. This parametric test is likely to be a more powerful test than the McFadden/Hausman test, first, because it is based on the Neyman-Pearson methodology and, second, because it will always use the entire sample. You could do it as follows:

```
CALC        ; Ran (seed for generator) $
MNPROBIT    ; ... specify the choices and utility functions
            ; Cor = 0 $
CALC        ; lu = logl $
CALC        ; Ran (same seed for generator) $
MNPROBIT    ; ... specify the choices and utility functions
            ; Sdv = 1
            ; Cor = 0 $
CALC        ; lr = logl
            ; List
            ; lrstat = 2 * (lu - lr) $
```

We applied this procedure in passing in the preceding section. The log likelihoods for the three models estimated were

Most restrictive:	$\sigma_j = 1$, $\rho_{jm} = 0$	Log likelihood = -195.5496
Restrictive:	$\sigma_j = 1$	Log likelihood = -194.9204
Unrestricted:		Log likelihood = -189.5252.

In principle, a test of the first assumption as the null hypothesis against the alternative of the second is sufficient to reject IIA. We found the chi squared to be 10.132 with two degrees of freedom. The critical value is 5.99, so the hypothesis is rejected. A test of the third model against the null of the first produced a chi squared of 12.048 with five degrees of freedom. The critical value is 11.07, so once again the hypothesis is rejected. Which test should be preferred is uncertain. Under the null hypothesis, the estimated parameters in the second model are more precisely estimated, so this may favor it. We are unaware of any other evidence on the question.

Chapter 12: Diagnostics and Error Messages

12.1 Introduction

The following is a complete list of diagnostics that will be issued by *NLOGIT*. Altogether, there are over 1,000 specific conditions that are picked up by the command translation and computation programs in *LIMDEP* and *NLOGIT*. Those listed here are specific to *NLOGIT*. The full set of diagnostics is given in Chapter R18 of the *LIMDEP Reference Guide*. Nearly all of the error messages listed below identify problems in commands that you have provided for the command translator to parse and then to pass on to the computation programs.

Most diagnostics are self explanatory and will be obvious. For example,

```
82  ;LHS - variable in list is not in the variable names table.
```

states that your Lhs variable in a model command does not exist. No doubt this is due to a typographical error – the name is misspelled. Other diagnostics are more complicated, and in many cases, it is not quite possible to be precise about the error. Thus, in many cases, a diagnostic will say something like 'the following string contains an unidentified name' and a part of your command will be listed – the implication is that the error is somewhere in the listed string. Finally, some diagnostics are based on information that is specific to a variable or an observation at the point at which it occurs. In that case, the diagnostic may identify a particular observation or value. In the listing below, we use the conventions:

<AAAAAAAA>	indicates a variable name that will appear in the diagnostic,
<nnnnnnnnnnnn>	indicates an integer value, often an observation number, that is given,
<xxxxxxxxxxxx>	indicates a specific value that may be invalid, such as a 'time' that is negative.

The listing below contains the diagnostics and, in some cases, additional points that may help you to find and/or fix the problem. The actual diagnostic you will see in your output window is shown in the Courier font, such as appears in diagnostic 82 above.

We note it should be extremely rare, but occasionally, an error message will occur for reasons that are not really related to the computation in progress. (We cannot give an example – if we knew where it was, we would remove the source before it occurred.) You will always know exactly what command produces a diagnostic – an echo of that command will appear directly above the error message in the output window. So, if an absolutely unfathomable error message shows up, try simplifying the command that precedes it to its bare essentials, and by building it up, reveal the source of the problem.

Finally, there are the 'program crashes.' Obviously, we hope that these never occur, but they do. The usual ones are division by zero and exponent overflow. Once again, we cannot give specific warnings about these, since if we could, we would fix the problem. If you do get one of these and you cannot get around it, please contact us at support@nlogit.com.

12.2 Discrete Choice (CLOGIT) and NLOGIT

1000 FIML/NLogit is not enabled in this program.

1001 Syntax problem in tree spec or expected ; or $ not found.

1002 Model defines too many alternatives (more than 100).

1003 A choice label appears more than once in the tree specification.

1004 Number of observations not a multiple of # of alternatives.
This is expected when you have a fixed choice set.

1005 Problem reading labels, or weights for choice based sample.

1006 Number of weights given does not match number of alternatives.

1007 A choice based sampling weight given is not between zero and one.

1008 The choice based sampling weights given do not sum to one.

1009 Expected [in limb specification was not found.

1010 Expected (in branch specification was not found.

1011 A branch label appears more than once in the tree.

1012 A choice label in a branch spec. is not in ;CHOICES list.

1013 Branch specifications are not separated by commas.

1014 One or more ;CHOICE labels does not appear in the tree.

1015 One or more ;CHOICE labels appears more than once in tree.

1016 The model must have either 1 or 3 LHS variables. Check spec.

1017 Nested logit model must include ;MODEL:... or ;RHS spec.
Found neither Model: nor RhS/Rh2.
Your model specification is incomplete.

1018 There is an unidentified variable name in the equation.
In the ; Model: U (...) part of the command, one of your specified utility functions contains a variable name that is not in your data set.

1019 Model specification exceeds an internal limit. See documentation.
RANK data can only be used for 1 level (nonnested) models.
You have specified a nested logit model and requested rank data for the observed outcomes. The nested logit model cannot be estimated with ranks data.

1020 Not used specifically. May show up with a self explanatory

```
      message.

1021  Using Box-Cox function on a variable that equals 0?

1022  Insufficient valid observations to fit a model.

1023  Mismatch between current and last models.
      This occurs when you are using the ; Simulation = ... part of NLOGIT.

1024  Failure estimating DISCRETE CHOICE model.
      Since this occurs during an attempt to compute the starting values for other models, if it
      fails here, it won't succeed in the more complicated model.

1025  Failed to fit model. See earlier diagnostic.
      This is a general diagnostic that precedes exit from the estimator.  An error condition has
      occurred, generally during estimation, not setup.

1026  Singular VC may mean model is unidentified. Check tree.
      What looks like convergence of a nested logit model may actually be an unidentified
      model.  In this case, the covariance matrix will show up with at least one column of
      zeros.  Sometimes it is more subtle than this.  In a complicated model, the configuration
      of the tree may lead to nonidentification.  A common source is too many constant terms
      in the model.

1027  Models - estimated variance matrix of estimates is singular.
      Non P.D. 2nd derivatives. Trying BHHH estimator instead.
      This is just a notice.  In almost all cases, the Hessian for a model that is not the simple
      MNL model will fail to be positive definite at the starting values.  This does not indicate
      any kind of problem.

1028  In ;SIMULATION=list of alts, a name is unknown.

1029  Did not find closing ] in labels[list].

1030  Error in specification of list in ;Choices=...labels[list].

1031  List in ;Choices=...labels[list] must be 1 or NALT values.

1032  Merging SP and RP data. Not possible with 1 line data setup.
      Merging SP and RP data requires LHS=choice,NALTi,ALTij form.
      Check :MERGERPSP(id=variable, type=variable) for an error.

1033  Indiv. <nnnnnn> with ID= <nnnnn> has same ID as another individual.
      This makes it impossible to merge the data sets.

1034  Specification error. Scenario must begin with a colon.

1035  Expected to find Scenario: specification = value.

1036  Unbalanced parentheses in scenario specified.
```

1037 Choice given in scenario: attr(choice...) is not in the model.

1038 Cannot identify attribute specified in scenario.

1039 Value after = in scenario spec is > 20 characters.

1040 Cannot identify RHS value in scenario spec.

1041 Transformation asks for divide by zero.

1042 Can only analyze 5 scenarios at a time.

1043 Did not find any valid observations for simulation.

1044 Expected to find ; LIST : name_x (choices). Not found.

1045 Did not find matching (or [in <scenario specification is given>.

1046 Cannot recognize the name <AAAAAAAA> in <scenario specification is given>.

1047 Same as 1046.

1048 None of the attributes requested appear in the model.

1049 Model has no free parameters among slopes!
This occurs during an attempt to fit the MNL model to obtain starting values for a nested logit or some other model.

1050 DISC with RANKS. Obs= <nnnnnn>. Alt= <nn>. Bad rank given = <nnnn>.
DISC w/ RANKS. Incomplete set of ranks given for obs. <nnnnnn>.
These are data problems with the coding of the Lhs variable.

1051 Singular VC matrix trying to fit MNL model.
When the MNL breaks down, it will be impossible to fit a more elaborate model such as a nested logit model.

1052 You did not provide ;FCN=label(distn),... for RPL model.

1053 Scaling option is not available with HEV, RPL, or MNP model.
Ranks data may not be used with HEV, RPL, or MNP model.
Nested models are not available with HEV, RPL, or MNP model.
Cannot keep cond. probs. or IVs with HEV, RPL, or MNP model.
Choice based sampling not useable in HEV, RPL, or MNP model.

These diagnostics are produced by problems setting up the scaling option for mixed data sets.

1054 Scaling option is not available with one line data setup.
Ranks data may not be used with one line data setup.
Choice set may not be variable with one line data setup.
One line data setup requires ;RHS and/or ;RH2 spec.
Nested models are not available with one line data setup.

```
      Cannot keep probabilities or IVs with one line data setup.

1055  Did not find closing paren in ;SCALE(list) spec.
      The list of variables to be scaled has an error.
      Only 40 or fewer variables may be scaled.
      You are attempting to scale the LHS variable.
      The list of values given for SCALE grid is bad.
      Grid must = Lo,Hi,N or Lo,Hi,N,N2. Check spec.
      Grid must have Low > 0 and High > low. Check #s.
      Number of grid points must be 2,3,... up to 20.

1056  Unidentified name in IIA list. Procedure omitted.

1057  More than 5 names in IIA list. Limit is 5.

1058  Size variables only available with (Nested) MNL.

1059  Cannot locate size variable specified.

1060  Model is too large: Number of betas up to 90.
      Model is too large: Number of alphas up to 30.
      Model is too large: Number of gammas up to 15.
      Model is too large: Number of thetas up to 10.

1061  Number of RHS variables is not a multiple of # of choices.
      This occurs when you are using a one line setup for your data.

1062  Expected ;FIX=name[...]. Did not find [ or ].

1063  In ;FIX=name[...], name does not exist: <name is given>.

1064  Error in fixed parameter given for <name is given>.

1065  Wrong number of start values given.
      This occurs with nested logit and other models, not the random parameters logit model.

1066  Command has both ;RHS and Model: U(alts). Inconsistent.

1067  Syntax problem in ;USET:(names list)= list of values.

1068  ;USET: list of parms contains an unrecognized name.

1069  Warning, ;IUSET: # values not equal to # names.

1070  Warning, ;IUSET: # values not equal to # names.

1071  Spec for RPL model is label(type) or [type]. Type=N,C,or L.

1072  Expected ,;$ in COR/SDV/HFN/REM/AR1=list not found.

1073  Invalid value given for correl. or std.dev. in list.
```

1074 ;COR/SDV=list did not give enough values for matrix.

1075 Error. Expected [in ;EQC=list[value] not found.
Error:Value in EQC=list[value] is not a correlation.
Error. Unrecognized alt name in ;EQC=list[value].
Error:List needs more than 1 name in EQC=list[value].
Error. A name is repeated in ;EQC=list[value].

1076 Your model forces a free parameter equal to a fixed one.

1077 Covariance heterogeneity model needs nonconstant variables.

1078 Covariance heterogeneity model not available with HEV model.
Covariance heterogeneity model is only for 2 level models.
Covariance heterogeneity model needs 2 or more branches.

1079 At least one variance in the HEV model must be fixed.
In *NLOGIT*, in the heteroscedastic extreme value, you have specified the model so that all the variances are free. But, for identification, one of them must be fixed.

1080 Multiple observation RPL/MNP data must be individual.

1081 Mismatch of # indivs. and number implied by groups.
WARNING Halton method is limited to 25 random parameters.

1082 Not used.

1083 MODEL followed by a colon was expected, not found.

1084 Expected equation specs. of form U(...) after MODEL.

1085 Unidentified name found in <string is given>.
This occurs during translation of ; Model: U (...) specifications.

1086 U(list) must define only choices,branches, or limbs.

1087 An equals sign was not found where expected in utility function definition.

1088 Mismatched [or (in parameter value specification.

1089 Could not interpret string; expected to find number.

1090 Expected to find ;IVSET:=defn. at this point.

1091 Expected to find a list of names in parens in IVSET.

1092 IVSET:(list) ... Unidentified name appears in (list).

1093 You have given a spec for an IV parm that is fixed at 1.

1094 You have specified an IV parameter more than once.

1095 `Count variable <nnnnnn> at row <nnnnnn> equals <nnnn>.`
The peculiar value for the count variable has thrown off the counter that keeps track of where the estimator is in the data set.

1096 `Choice variable <AAAAAAAA> at row <nnnnn>: Choice= <nnnnn>.`
The most likely cause is a coding error. Check for bad data.

1097 `Obs. <nnnnnn>: Choice set contains <nnnn> <nnnn> times.`
The choice variable for individual data has more than one 1.0 in it. *NLOGIT* cannot determine which alternative is chosen.

1098 `Obs. <nnnnnn> alt. <nnn> is not an integer nor a proportion.`

1099 `Obs. <nnnnnn> responses should sum to 1.0. Sum is <xxxxxx>.`

1100 `Cannot classify obs. <nnnnnn> as IND, PROPs, or FREQs.`
Your data appear to be a mix of individual and frequency data. This occurs when an individual's Lhs variable data include zeros. It then becomes difficult to determine what kind of data you have. You can settle the question by including ; Frequencies in your command, if that is appropriate.

1101 `# of parms in < list > greater than # choices in U(list).`

1102 `RANK data can only be used for 1 level (nonnested) models.`