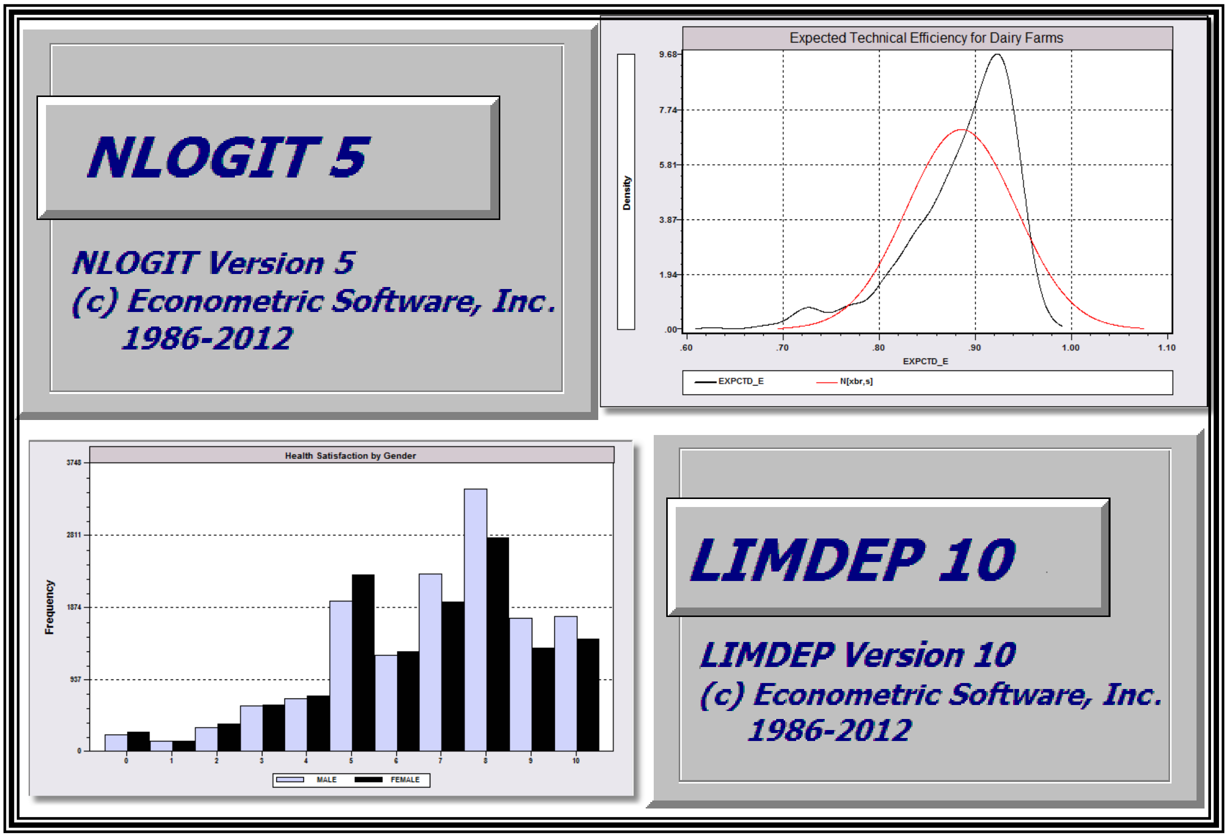


A Quick Start Introduction to NLOGIT 5 and LIMDEP 10

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

Contents

I. Introduction

This short getting started guide will show you how to operate *NLOGIT* and *LIMDEP*. The manuals for *NLOGIT* and *LIMDEP* are several thousand pages long, and document hundreds of models, estimators, and other program procedures. This guide will show you how to operate the program and use it to do some of the most common calculations. The program's interface uses the same basic forms for most of the functions it performs. Based on what we do here, you will be able to construct command streams to do complex analyses using many of the features of the program.

The two programs operate exactly the same way, with the same command set and user interface. *NLOGIT* 5 is in fact, *LIMDEP* 10 plus one (extremely large) command set. This short manual will show how to operate both programs. For convenience, the discussion will assume you are using *NLOGIT*, but everything noted applies equally to *LIMDEP* as well. A short discussion in Section VIII will introduce the specific difference between *NLOGIT* and *LIMDEP*.

II. The Desktop: Startup NLOGIT or LIMDEP 

Your program is installed on your computer and you are ready to begin. There is an icon for *NLOGIT* 5 or *LIMDEP* 10 on your desktop, and the program is included in your startup menu. Launch your program.

When you first start the program your desktop will look as in Figure 1. (*LIMDEP* and *NLOGIT* use the same desktop and functionality. You can see which program you are using by the name that appears at the upper left corner of the desktop. Notice for our discussion here, we are using *NLOGIT* 5. Operation of the two programs is identical. (The only difference between them is the (large) set of multinomial choice models that are supported in *NLOGIT* and not in *LIMDEP*.) We note two small differences that may appear between our desktop in Figure 1 and yours. First, the setting 'U:38888 Rows: 38888' appears at the top of the window at the left of our desktop. This is a setting that we (you) can make that relates to how large a data set you want your program to be able to store. A different value will appear the first time you launch the program. Second, the small editing window we call the 'command bar' that we have indicated with a red arrow in Figure 1 may not be present on your desktop. You can install this as follows: click Tools→Options→View – note the Tools menu item is above the tip of the red arrow – then click in the check box next to 'Display Command Bar' and finally, click OK. This setting is fixed until you change it. Finally, your row of buttons may be above your command bar, not below it. You can move this around the screen as you like.)

The window that is open at the left of the desktop is called the 'Project Window.' There is a large amount of functionality operated from this window, as will be clear shortly.

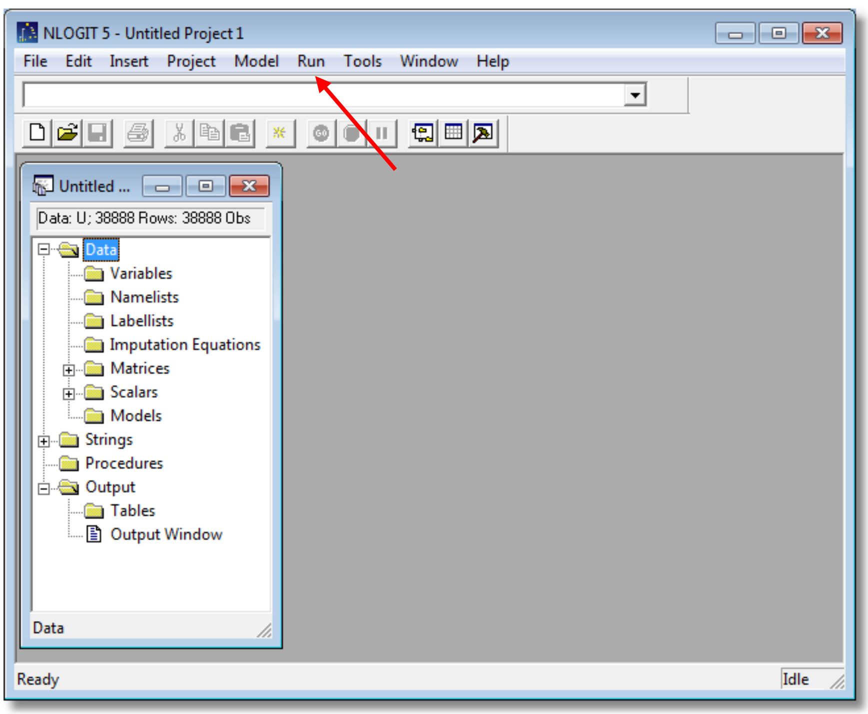
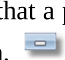


Figure 1. Initial Desktop

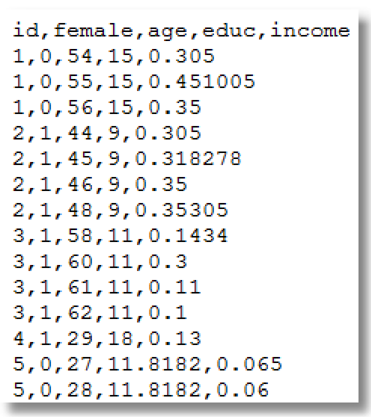
A Tip: *NLOGIT* uses a standard statistical package style, three window mode of operation. The first window you will see is the 'Project' window. A project consists of the data you are analyzing and the results of your computations, such as estimates of coefficients, other matrices you might have computed, and so on. As we'll see shortly, this window contains an inventory of the things you have computed – the inventory will grow as you manipulate your data. *You should never close the project window*. Nearly all of the program functions operate only when a project is active. You know that a project is active when the project window is present and open. (You can minimize it with the left sizing button,  . But, do not close it with the red × button,  .)

III. Operating NLOGIT and LIMDEP

NLOGIT provides both menu/dialog boxes and a command language that you can use to operate the program. All of the basic functions of the program can be operated with either. However, many of the more complex operations, including most of the involved models, are accessed only through the command language. We will take a quick look at both of these now.

A. Data Files

You will use *NLOGIT* to analyze data. To get started, we'll note a couple things about data. The data you use will have to come from somewhere – probably a public data source, or in a file that you obtained from some external source. (You can create data within *NLOGIT*, for example, by using the random number generators, but you will rarely do this exclusively. Usually, created data are added to existing data sets.) Data files come in many forms. *NLOGIT* can read many different kinds of files, and with modern interchange programs such as *Stat Transfer,* you can convert files from many more sources that might be foreign to *NLOGIT* to a form that *NLOGIT* is comfortable with. These issues are discussed in the manual. The most common generic file type used by contemporary researchers is the 'CSV' format. A CSV file (i.e., 'comma separated values' format) has a line of variable names at the top and rows of data below them, with values separated by commas, such as the data set in Figure 2 below. Figure 2 show the data in a small demonstration file that we will use named IncomeData.csv. (In Figure 2, we are viewing the contents of IncomeData.csv in *NLOGIT*'s text editor, which we'll discuss below.)



```
id,female,age,educ,income
1,0,54,15,0.305
1,0,55,15,0.451005
1,0,56,15,0.35
2,1,44,9,0.305
2,1,45,9,0.318278
2,1,46,9,0.35
2,1,48,9,0.35305
3,1,58,11,0.1434
3,1,60,11,0.3
3,1,61,11,0.11
3,1,62,11,0.1
4,1,29,18,0.13
5,0,27,11.8182,0.065
5,0,28,11.8182,0.06
```

Figure 2. A CSV File

B. Operating with the Menus and Dialogs

We'll start by importing the data in IncomeData.csv into the program so that we can analyze them. Select Project→Import→Variables… as shown in Figure 3. This will open a Windows Explorer as shown in Figure 4 that you can use to navigate to your file. Make your way to where the file is installed on your computer. On your computer, this should be in the C:\NLOGIT5 or C:\LIMDEP10 folder. It may be in some other folder depending on how you installed this tutorial on your computer. Select IncomeData.csv in the menu.

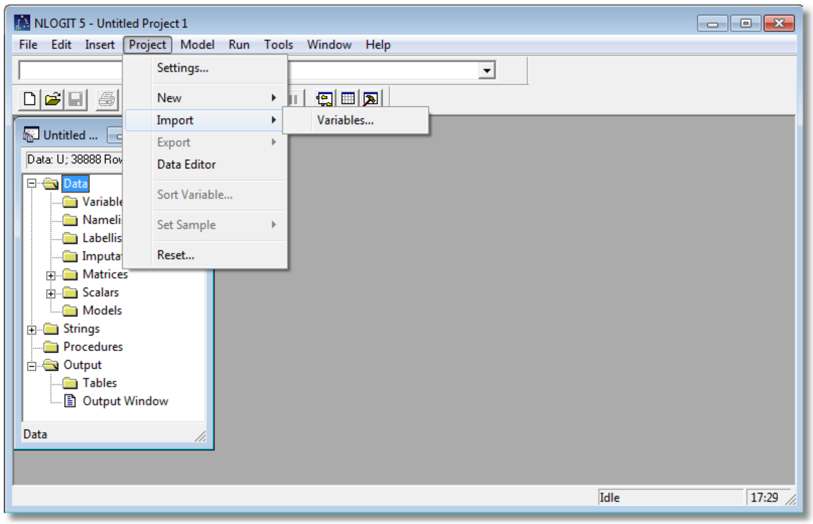


Figure 3. Desktop Project Menu for Importing a CSV File

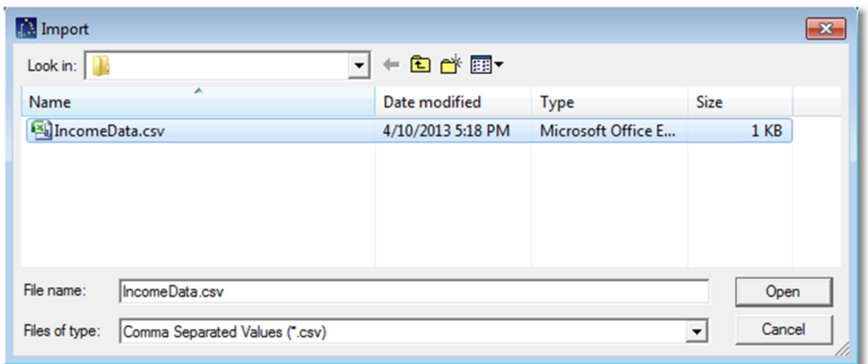


Figure 4. Windows Explorer

After you click Open, the data file will be imported into *NLOGIT*'s work area and will be ready for you to analyze them. Note in the project window in Figure 1, within the window, in the 'Data' area, the first item (folder) is 'Variables.' There is nothing at the left of the title, however. After you import your data, the Variables folder will indicate that it contains data, as shown in Figure 5. Note the + ne⊠t to the folder name.

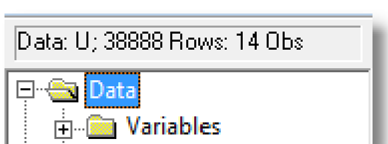


Figure 5. Variables Folder in Project Window

The project window will now indicate that there are 14 Rows of data – that is the number of observations in the data file that we just read. If you click the ⊞box at the left of Variables to open the folder, the list of variables that have been read will be displayed, as shown in Figure 6. This is our active data set. You can visit the actual data by activating the data editor. The button that will open the data editor is indicated by the red arrow in Figure 6. The spreadsheet style data editor is shown in Figure 7. You can enter and replace data in the editor. After you examine the data editor, you can minimize it or close it. (Closing the data editor only hides the display – it does nothing to the active data set.)

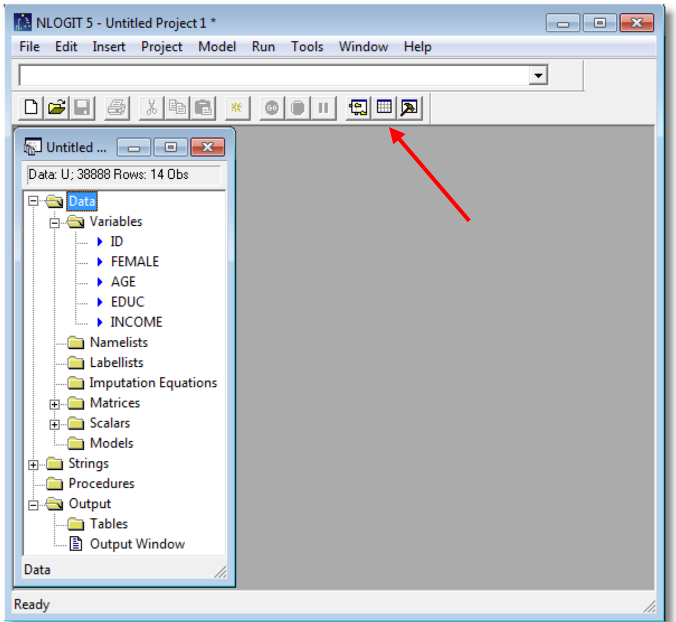


Figure 6. Active Data Set.

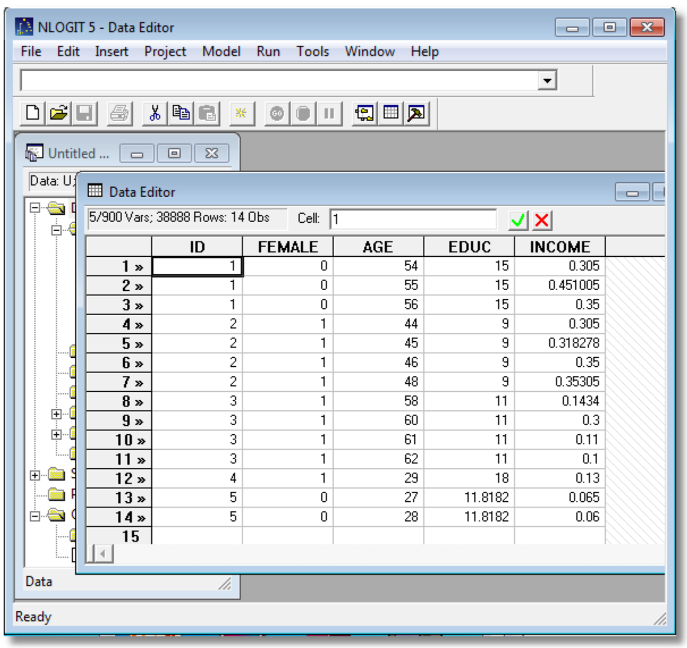


Figure 7. Data Editor.

Since the data are ready to use, we will do some computations. From the desktop, select Model→Linear Models→Regression… as shown in Figure 8. This will open a dialog box (Figure 9) that we call a 'Command Builder.' You'll see why momentarily. We'll build a regression command. First select the dependent variable (INCOME) from the drop down menu as shown in Figure 9. The independent variables are chosen in the windows below the dependent variable. Independent variables are selected by 'selecting' them in the right window, then using '<<' to move them to the left window, as shown in Figure 10. Select ONE, AGE and EDUC for our model.

A Tip: ONE is the constant term in the model. *NLOGIT* does not automatically place a constant term in any model. It must be requested by including ONE (a program created variable) in the list of independent variables.

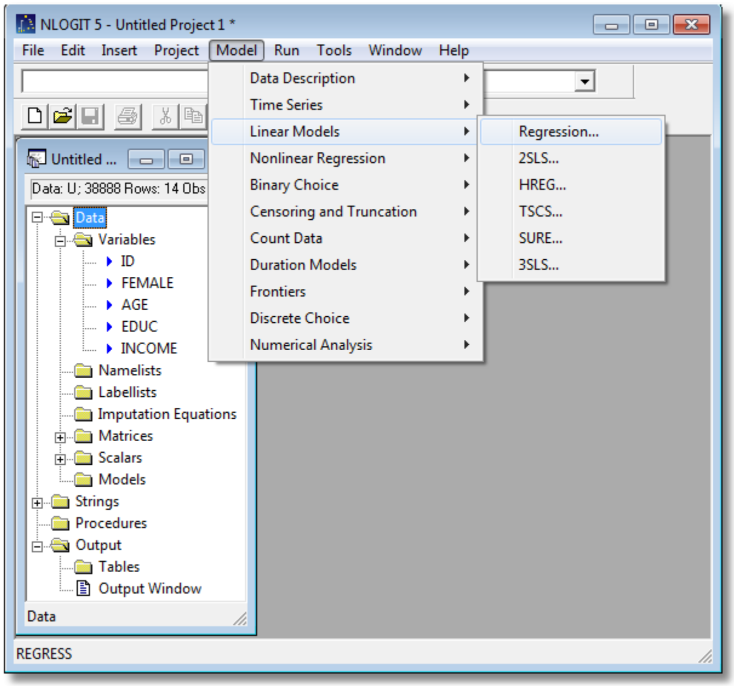


Figure 8. Model Menu

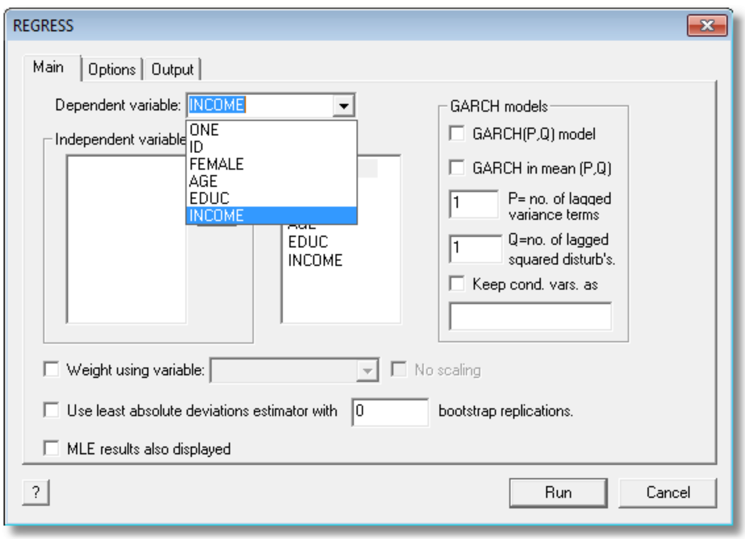


Figure 9. Command Builder: Dependent Variable

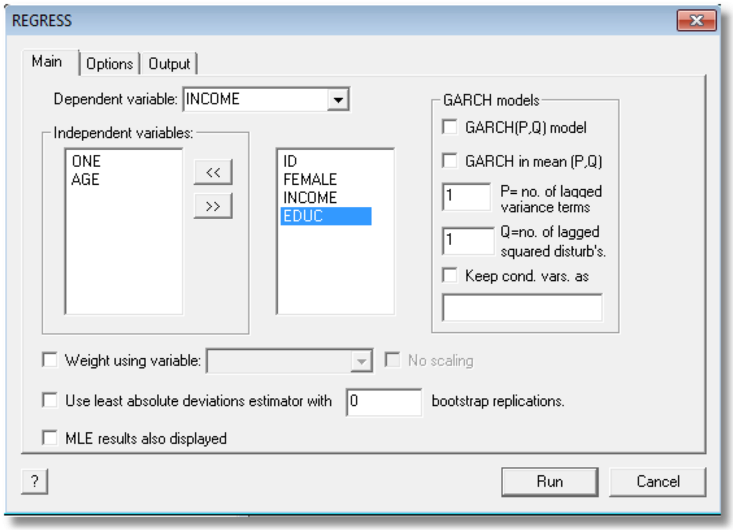


Figure 10. Command Builder: Independent Variables

After your model is specified in the command builder – note that there are other options on this page, and two more tabs that promise still more options – press the ‘Run’ button at the lower right of the dialog box. Run asks the program to compute the specified regression. A new window, the ‘Output Window’ opens and displays your regression results. Note above the regression results, there is a line of green text. This is the **REGRESS** command that was built by the command builder. This is the second window (the data/project window is the first) noted as the three window format. The editing window discussed next is the third.

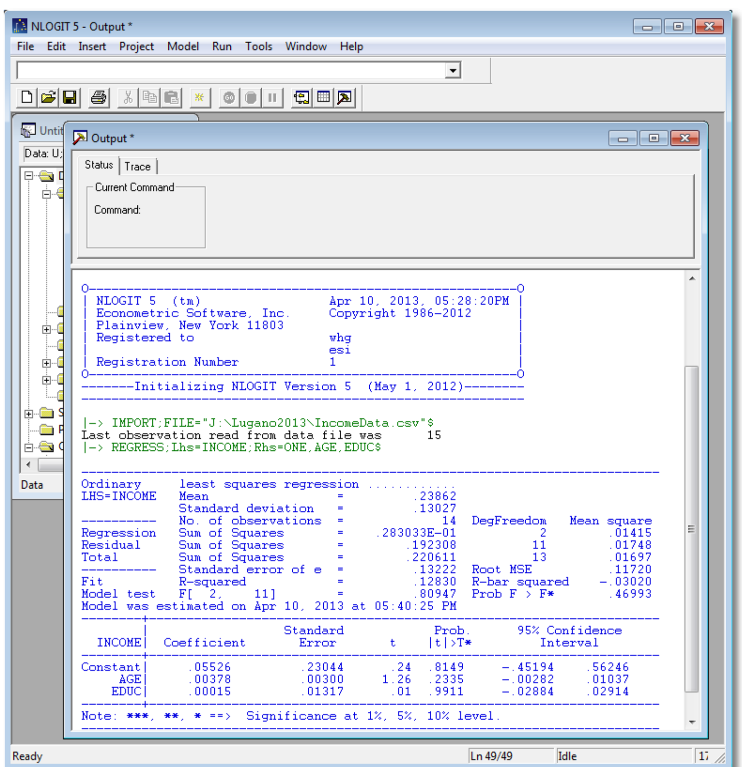


Figure 11. Regression Results in Output Window

You have now launched the program, read a data set and computed a regression (without touching your keyboard).Now, we will do the same operations using the *NLOGIT*'s command language. We'll start a new session to demonstrate this procedure. Like most other programs, you leave *NLOGIT* by using File→Exit. The File menu is where it always is in Windows programs, at the upper left of the desktop, and Exit is, as usual, at the bottom of the menu. On your way out, you will be asked about saving the project, Untitled Project 1, and the output window, Untitled Output 1. Click 'NO' both times and the program will close.

C. Using Commands and the Command Editor

We will now use *NLOGIT*'s command language to import the data and compute the regression. Commands are issued by typing them in a text editing window and 'submitting' them to *NLOGIT*'s command processor.

Restart the program as before to produce the empty desktop as in Figure 1. Click File to open the menu as shown in Figure 12. Select 'New' at the top of the menu, and the small dialog will open and offer to open a Text/Command Document window or a Project. Select the Text/Command Document option. (You already have a project open.)When you select the Text/Command Document option, the editing window shown in Figure 13 will open.

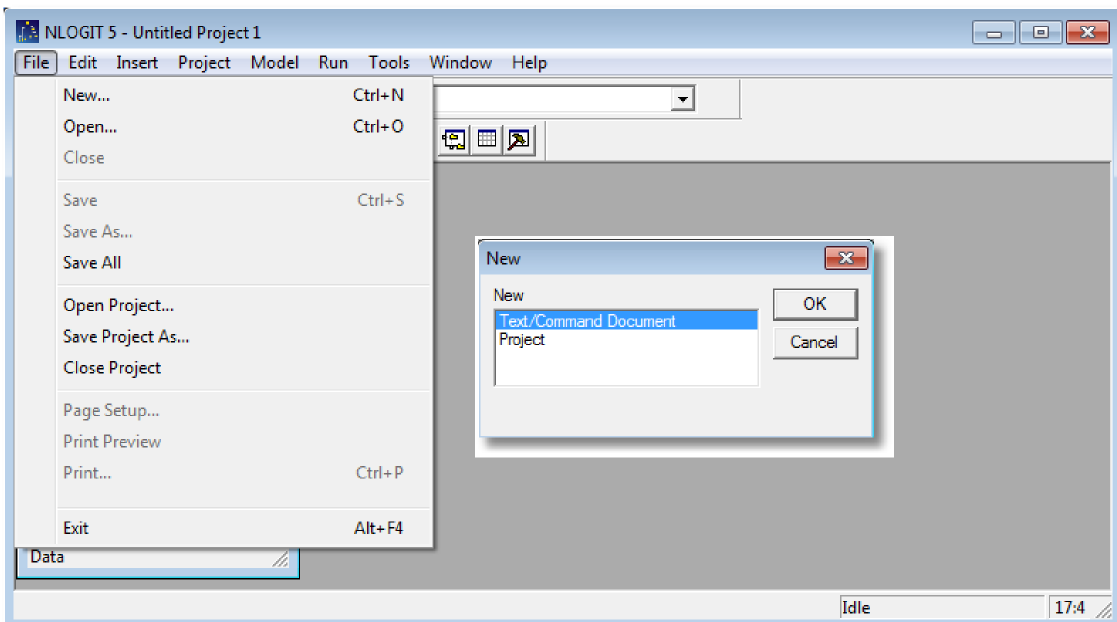


Figure 12. File Menu and New Dialog

This is the text editor. You can edit anything in it. (You can also have multiple text editing windows open at the same time.) We will enter our commands in this window, then submit them to the command processor.

We want to do two operations right now, import our data file and compute a linear regression. We type the commands in the editor. The two commands that we wish to carry out are '**IMPORT**' and '**REGRESS**.' We'll say more about the commands in a moment. First, type the string 'import;file=' in the first line of the window. Now, it's not sure exactly where the file is on your computer. If you know, you type the path to it after the equals sign. File names are enclosed in double quotes. A **$** character is used to end the instruction. (Always, all instructions.) If you don't know the path to the file, find it as follows: In the desktop menu, select Insert→File Path… and use the Windows explorer to find your file (IncomeData.csv). This will place the file path in the line where you want it, and you need only add the ending $ to complete the command. Press the Enter key. On the next line, type the REGRESS command as shown. Note that it has some parts separated by semicolons and, as always, ends with a $.

A Tip: If you are using a desktop computer with a separate keyboard, use the *alphabetic* Enter key here. The Enter key in the numeric keypad at the right of the keyboard is not the same when you are using *NLOGIT*. We'll note why shortly.

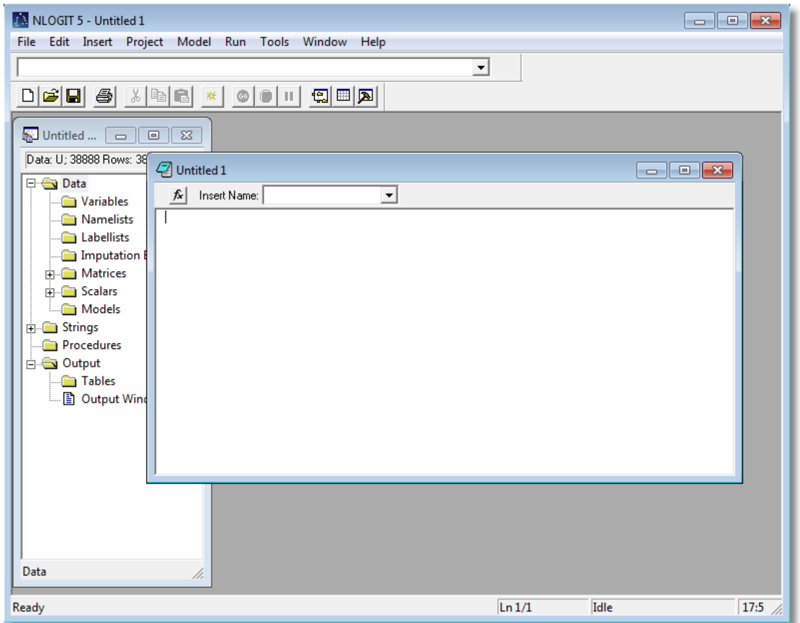


Figure 13. Text Editing Window

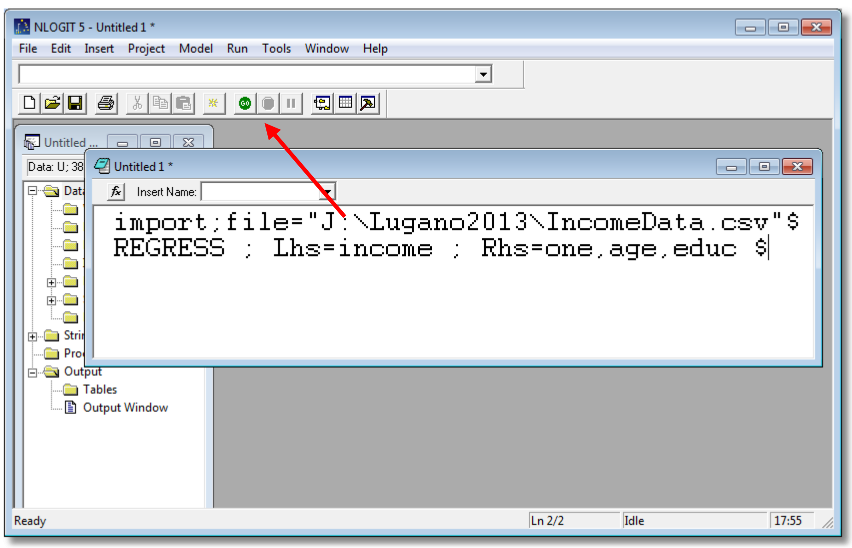


Figure 14. Text Editing Window

Now that your two commands are in the editing window, you can submit them. Highlight the two lines of text as if you were about to copy them in an editor such as *Microsoft Word*. When the two lines are highlighted, press the 'GO' button that is noted by the red arrow in Figure 14. The output window will appear and will indicate that your file was imported, and the regression was computed.

A Tip: Pressing the numeric keypad's Enter key is the same as highlighting the one line that the cursor is in and then clicking GO. This is how the two enter keys differ. You can always (and only) submit one line this way.

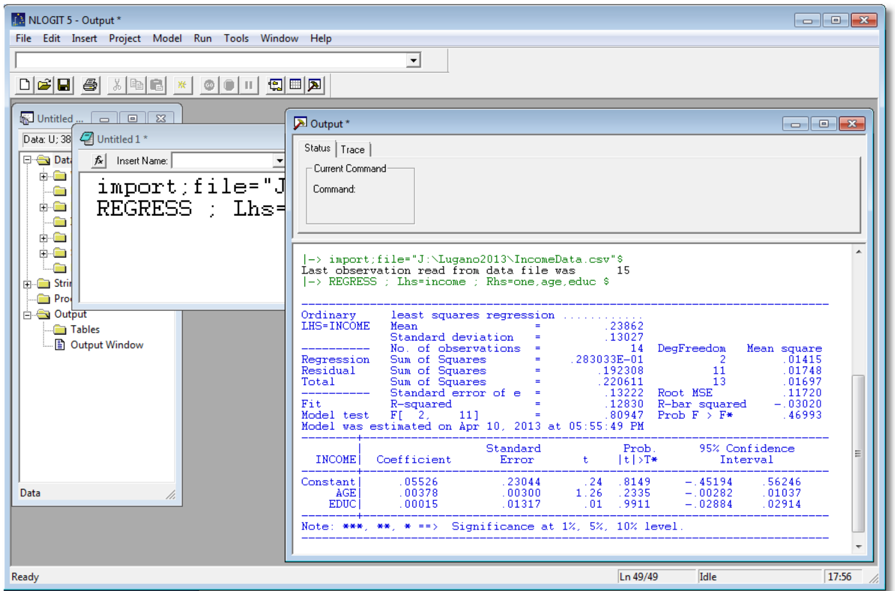


Figure 15. Commands Executed from Text Editor.

You are not limited to one or the other of these two modes of entering instructions. You can use either the command editor or the menus and dialog boxes whenever you wish. In Figure 16, we have imported the data set using the command editor, then run the regression with the command builder.

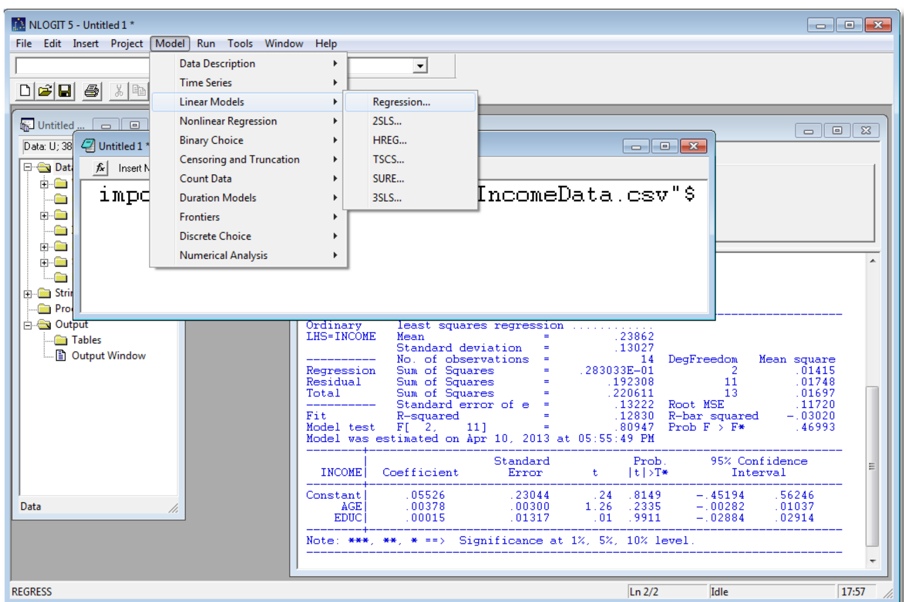


Figure 16. Using Both Commands and Dialog Boxes.

A Tip: Notice in Figure 11, the command builder has placed a copy of the command it created in the output window. You can 'copy' this command in the output window, 'paste' it into the editing window, and submit it again. This would be useful if you want to modify the command, for example by adding more independent variables. (The command processor will ignore the leading '|->' if you happen to include it in your copy of the command.

IV. Stopping, Restarting and Data Sets

You should only import a data set once. When you exit the program, you are offered a chance to save your data (as any modern program does). For *NLOGIT*, this is the project. The dialog in Figure 17 will appear when you select File→Exit. The project contains your active data set as well as a long list of other things you create as you operate the program. You can save the project with any name (and at any time) with File→Save Project As... It will be saved as an LPJ file – Windows recognizes this file extension. In Figure 18, we are saving the data in a work folder as IncomeData.lpj. When the program is restarted, instead of *importing* the original data, we merely *load* the project. The most recent 4 saved projects will appear in the File menu, In Figure 19, IncomeData.lpj appears in the File menu, and can be selected to resume the analysis of the data. Note in the lower panel of Figure 19, the project file name at the top of the desktop and at the top of the project window is IncomeData.lpj, rather than Untitled Project 1, as it was before.

A Tip: The project will always be current. When you add variables, create new ones, for example by transforming the raw data, the new variables are always saved in the project.

Another Tip: You can launch a project file from Windows Explorer – the same way that selecting a .docx file launches Microsoft Word then imports the document.

A Third Tip: You can also save the text editor as a LIM file. The pair of files constitutes your entire working session. You can resume a session exactly where you were when you exited by reloading these two files.

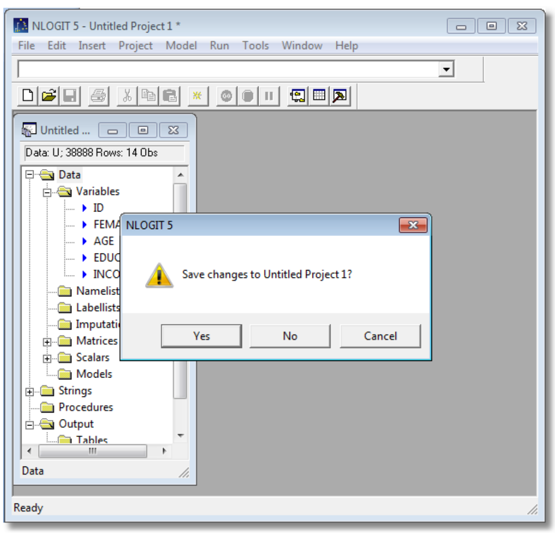


Figure 17. Saving the Project Upon Exit

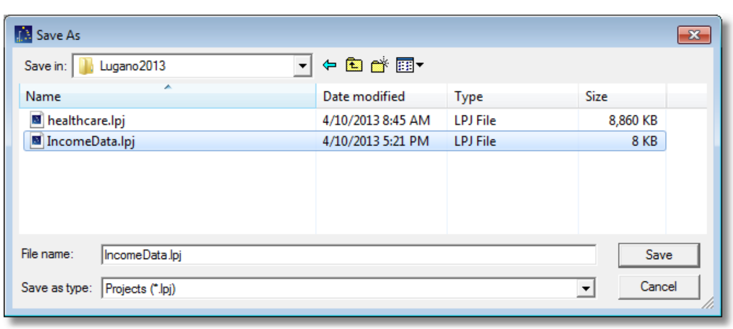


Figure 18. Windows Explorer Saving a Project File

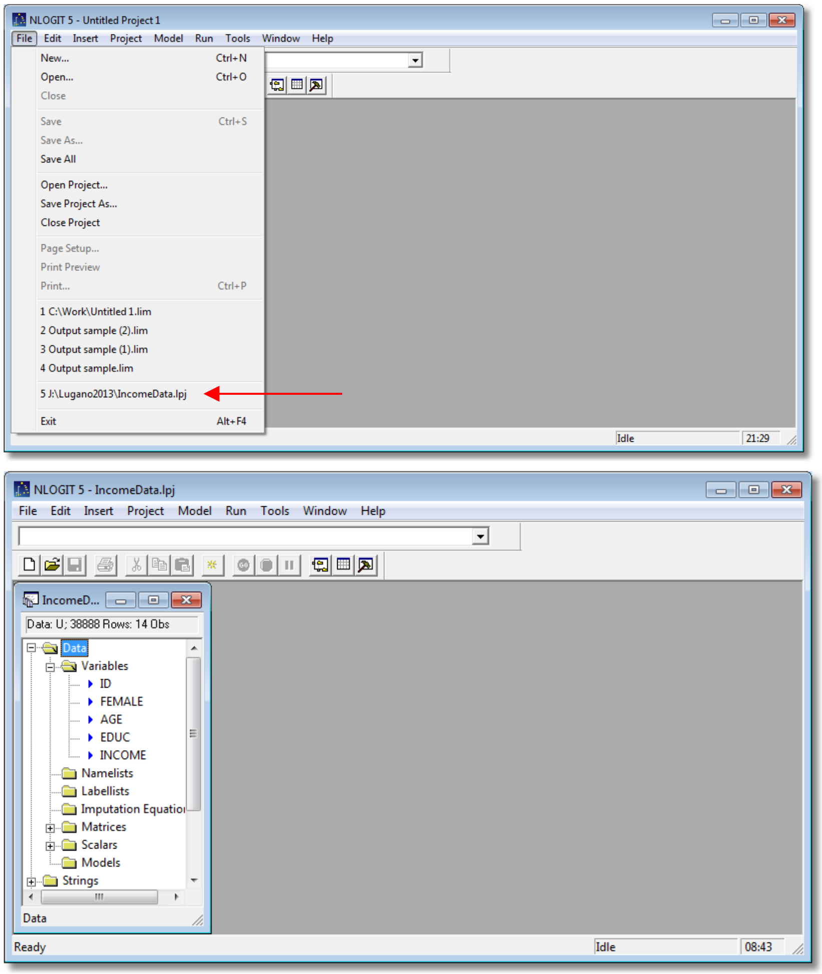


Figure 19. Reloading a Project from the File Menu

V. NLOGIT Commands

The menus and dialog boxes are helpful for operating the program. But, they are a bit inconvenient compared to the command editor. Users generally quickly migrate to the command structure for most operations. With that in mind, we will show the basic form of *NLOGIT* commands, and note some specific ones that you are certain to use. Altogether, there are several hundred different commands and functions. You can operate a large fraction of the program functionality with a few of the most important ones.

A. Commands in the Command Editor

There is a specific protocol for using the text editor to submit commands and a specific format for the commands, themselves. In general, there are very few structures or restrictions. The command language is designed to be convenient and self documenting. Instructions look like what they are requesting. For using the text editor:

- Case almost never matters. Notice in Figure 14, the verb in the **REGRESS** command is all in capital letters whereas the rest of the command is a mix of caps and lower case. Case only matters for file names and in the titles used for graphics and output tables that you construct (where you would want to use both cases). Otherwise throughout the program, commands can any mix of lower case and upper case letters.

- Spacing only matters in titles and file names. Notice there are some spaces put in the **REGRESS** command, for clarity. The spaces have no other meaning. In particular, lists of items are always delimited by punctuation, usually commas, never by spaces. You can use spaces in commands anywhere you wish to make them easier to read.

A Tip: You can copy commands out of documents such as Word files and paste them directly into the editor. Tab characters will be treated like spaces. A warning, however, the Word dash character, –, is not the same as an ASCII minus sign. You will generally have to change this manually.

- The number of lines used for a command is arbitrary. Line breaks are used for clarity and ease of interpretation of commands. No special connector is needed to connect the lines of multiple line commands. Some commands for complicated models have many parts, and breaking commands into multiple lines is helpful for self documentation. For example,

```
REGRESS     ; LHS = income ; RHS = one,age,educ $
```

is exactly the same as

```
REGRESS     ; Lhs = income
            ; Rhs = one,age,educ
            $
```

B. Names

You will create many items, including variables, that have names. Names are limited to 8 characters. The first must be a letter. Allowable characters are letters, digits and the underscore character. Since the program is not case sensitive, different cases of letters do not create different variable names. Of course, since spaces have no meaning, they may not appear in names (they are ignored) There are many types of names used in *NLOGIT*, including variables, matrices, scalars, synonyms for lists of names, label lists, names used for model definitions, names for output tables, and others. All obey the same conventions.

C. Command Structure

All commands are of the form

VERB ; information ; information ; … $

Note the two commands in the text editor in Figure 14, **IMPORT** and **REGRESS**. There are altogether about 200 verbs that manage files, manipulate the data, fit models and do ancillary computations such as test hypotheses. The common structure is as follows:

- Every command must begin on a new line
- Every command must end with a $ at the end of the last line.
- There is no restriction on how many lines may be used for a command
- There is no restriction on what may be included on specific lines.
- Commands may not have more than 10,000 nonblank characters. You will never come close to this limit.

You may have blank lines in your text editor even in the middle of the commands. Since you submit only the lines you want executed, you may put any other text anywhere you wish in the editor. Explicit comment lines may be inserted by beginning the text with a question mark. E.g.,

```
? This command computes a regression.
REGRESS ; Lhs = income ; Rhs = one,age,educ $
```

A block of lines of text may be marked as comment. For example,

```
/*
The following commands carry out two regressions.
The first uses x1.
The second uses x1 and x2.
*/
REGRESS ; Lhs = y ; Rhs = one,x1 $
REGRESS ; Lhs = y ; Rhs = one,x1,x2 $
```

This construction would seem to be of marginal usefulness. One way it would be helpful would be for having documentation in command files that you can execute directly with the Run menu Shown in Figure 20.

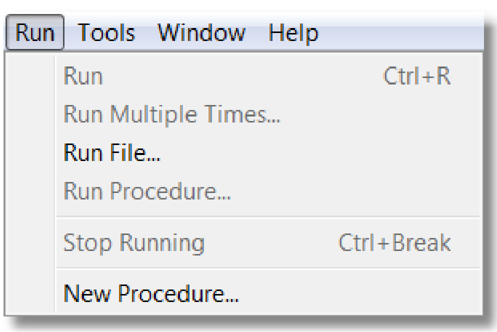


Figure 20. Run Menu for Run File…

VI. Some Essential Operations

The following lists a handful of operations that will be part of most analyses.

A. The Active Sample

When you import a data set, the active sample is all the observations in the data set. Figure 21 shows the income data we are examining in our demonstration. There are 14 observations in the data set. Note, the 14 rows are numbered and there is a chevron (») in each row. The » indicates that the observation is in the 'current sample.' The active sample can be changed in several ways. Three commands, **SAMPLE**, **REJECT** and**INCLUDE** are used specifically to change the sample.

SAMPLE ; n1 – n2 $

sets the sample to be rows n1 to row n2. For example,

SAMPLE ; 4 – 12 $

in the example would select the observations shown in Figure 21b. Note the chevrons are now only present for the active subset of the data. The excluded observations are not lost. But, any operation that manipulates the data set operates only on these observations. The full sample is restored with

SAMPLE ; All $

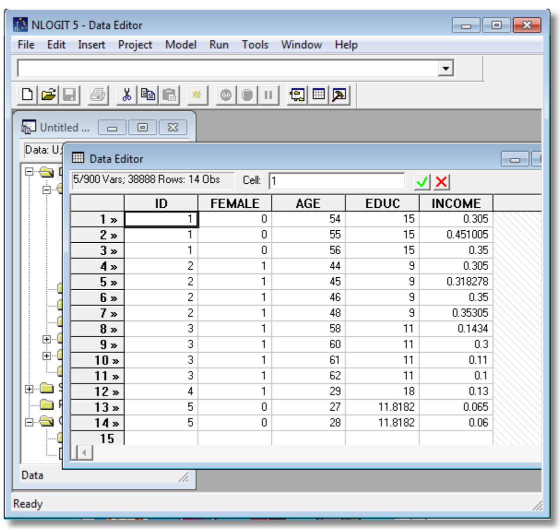


Figure 21a Active Data Set

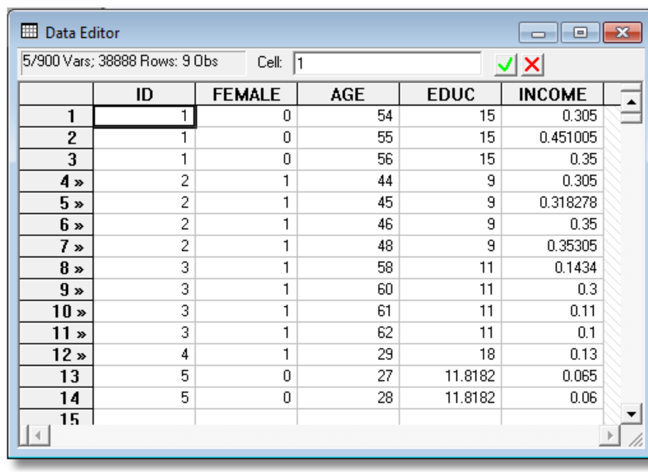


Figure 21b. Active Data Subset

The two other commands used directly to change the sample are

REJECT ; condition $ such as **REJECT ; age > 60 $,**

which removes observations from the active sample, whatever it happens to be, and

INCLUDE ; condition $ such as INCLUDE ; female = 1 $

which adds observations to the current sample, whatever it happens to be. These commands can be applied to the full data set, or no dat set, respectively, by including **;New**. For example,

INCLUDE ; New ; Female = 1 $

starts with no observations, then adds to the empty data set all observations in the full data set that have Female = 1.

A Tip: In many cases, you will want to fit a model using a subset of the active data set, but not wish actually to change the active data set. A model command can do that automatically. For example,

REGRESS ; If [age < 60] ; Lhs = income ; Rhs = one,age,educ $

B. Missing Values

The internal missing value code is -999. In the data editor, -999 will appear as a blank. In general, you must inform *NLOGIT* what to do about missing values. In general, *NLOGIT* only acts on missing data when you ask it to do so. If your sample contains missing values and you make no indication, the -999s will be treated as ordinary data. A global command to tell the program to bypass missing values when it fits models is

SKIP $

In the desktop, you can use Project→Settings→Execution and check the box for skipping missing data. **SKIP$** is a fixed setting. It persists from model to model. You can turn it off with **NOSKIP$** if you wish.

A Tip: *NLOGIT* contains a large package for *multiple imputation* of missing values.

C. Transformations

You will usually want to compute transformed variables. The command is

CREATE ; variable = expression ; variable = expression ; … $

The left hand variable may be a new variable created from existing variable(s) or may be an existing variable, which will be replaced. For example,

CREATE ; logincm = log(income) ; agesq = age^2 / 100 $

A common calculation is creating dummy variables. There are many ways to do so. For example, two ways to create the variable YOUNG equal to 1 if AGE is less than 25 and 0 otherwise would be

CREATE ; young = age < 50 $

and

CREATE ; if(age < 50)young = 1 $

Note that the log income variable uses a function, log(.). There are over 200 functions supported, including log, exp, abs, min, and many special functions. All functions have 3 character names. *NLOGIT* contains 20 different random number generators, such as Rnn(mean,standard deviation) which computes a random sample of observations from a normal distribution with the indicated mean and standard deviation. Functions may appear in expressions. For example, to create a sample of observations from the *F* distribution with 5 and 27 numerator and denominator degrees of freedom, you might use

CREATE ; fsample = (rnx(5)/5) / (rnx(27)/27) $ (But, this would be the same as **Rnf(5,27)**.**)**

The seed for the generators is set using

CALC ; ran(value) $

You will use this to be able to replicate your analyses that use random values.

D. Variable Lists in Model Commands

Model commands contain lists of variables. The lists can be extremely long – possibly hundreds of variables. There are several shortcuts provided. The primary device is

NAMELIST ; name = list of variables $

For example,

NAMELIST ; x = one,age,educ,female $
REGRESS ; Lhs = income ; Rhs = x $

Namelists provide a convenient shortcut for model commands. They also serve many other functions. One major one is defining data matrices. For example, to compute 'by hand' the least squares coefficient vector that is reported by **REGRESS** above, we could use

MATRIX ; bols = <x'x> * x'income $

The construction **<matrix>** is *NLOGIT*'s syntax for computing the inverse of a matrix. Note that the namelist and the variable become a data matrix and a data vector when used in a matrix command.

1. Categorical Variables

Categorical variables are often used in models in the form of a set of dummy variables with one of the dummy variables being dropped as the 'base case.' In the example below, rather than use EDUC in years, we have used RECODE to creat a category variable ED which is 0 when EDUC is 0-9, 1 when EDUC is 10-12, and 2 when EDUC is 13-20. The regression would then use dummy variables for the second and third categories. A special format, #name, is used for category variables. It is not necessary actually to compute the dummy variables. Note the results in Figure 22, which reports how the variable #ED has been used in the regression.

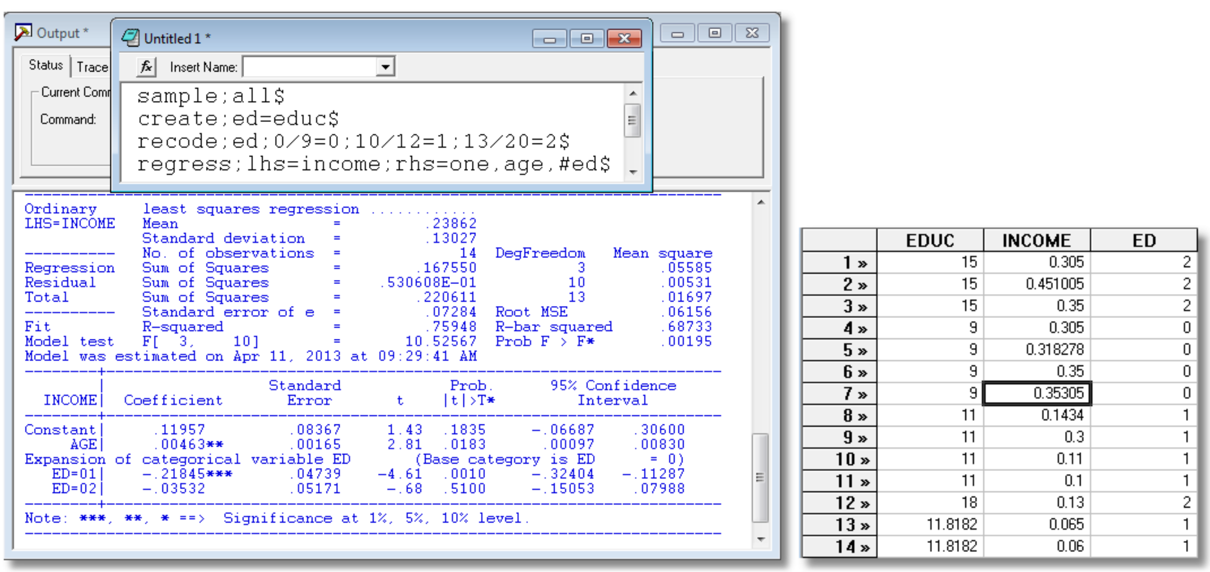


Figure 22. Categorical Variable in Regression

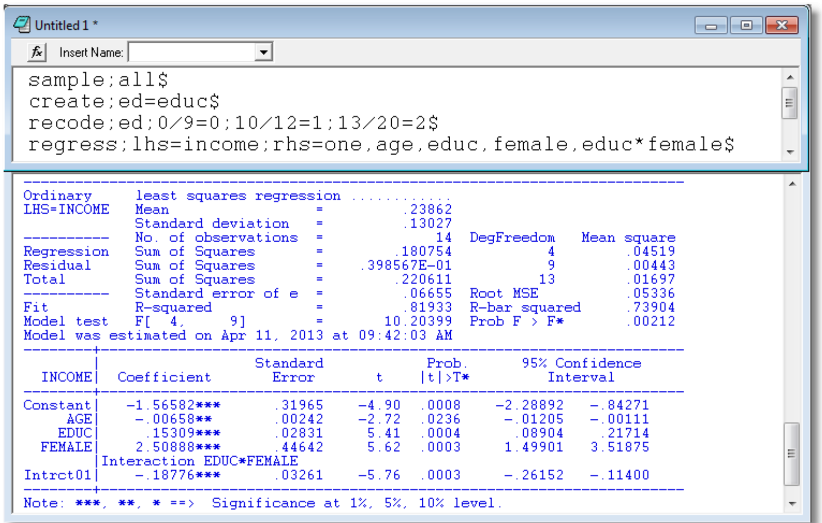
2. Interaction Terms

A second common feature of models is 'interaction terms.' In the model results in Figure 23, we have included education, female, and an interaction between education and female. Note that the command contains the interaction. We do this rather than computing a product variable, say **CREATE;EducFeml=Educ*Female$**

A Tip: Namelists may contain interactions. For example,

NAMELIST ; EdFem= female,educ*female $
REGRESS ; Lhs = income ; Rhs = one,age,edfem $

Note that EdFem is not a variable. It is a list of three variables, one of which is a product of two variables. You can also include the interaction terms directly in the model command, as shown in Figure 23.

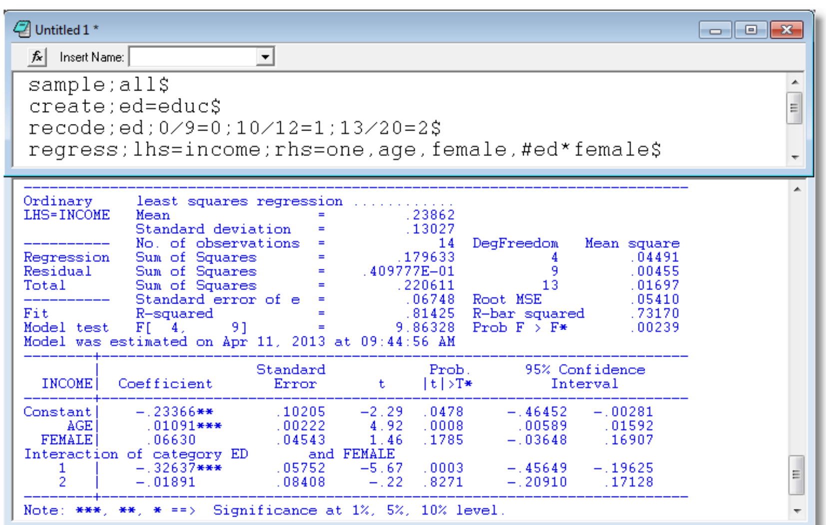


```
sample;all$
create;ed=educ$
recode;ed;0/9=0;10/12=1;13/20=2$
regress;lhs=income;rhs=one,age,educ,female,educ*female$
```

```
-----------------------------------------------------------------------------
Ordinary     least squares regression ............
LHS=INCOME   Mean                   =      .23862
             Standard deviation     =      .13027
----------   No. of observations    =          14  DegFreedom   Mean square
Regression   Sum of Squares         =     .180754           4        .04519
Residual     Sum of Squares         =  .398567E-01          9        .00443
Total        Sum of Squares         =     .220611          13        .01697
----------   Standard error of e    =      .06655  Root MSE          .05336
Fit          R-squared              =      .81933  R-bar squared     .73904
Model test   F[  4,     9]          =    10.20399  Prob F > F*       .00212
Model was estimated on Apr 11, 2013 at 09:42:03 AM
--------+--------------------------------------------------------------------
        |                Standard            Prob.      95% Confidence
  INCOME|  Coefficient     Error       t    |t|>T*         Interval
--------+--------------------------------------------------------------------
Constant|   -1.56582***    .31965    -4.90  .0008    -2.28892   -.84271
     AGE|    -.00658**     .00242    -2.72  .0236     -.01205   -.00111
    EDUC|     .15309***    .02831     5.41  .0004      .08904    .21714
  FEMALE|    2.50888***    .44642     5.62  .0003     1.49901   3.51875
        |Interaction EDUC*FEMALE
Intrct01|    -.18776***    .03261    -5.76  .0003     -.26152   -.11400
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
```

Figure 23. Interaction Effect in a Model

It is possible as well to have interactions of categorical variables and other variables, as shown in Figure 24. Although it is unlikely that you would need it, it is also possible to have interactions of categorical variables. The procedure is described in the manual.



```
sample;all$
create;ed=educ$
recode;ed;0/9=0;10/12=1;13/20=2$
regress;lhs=income;rhs=one,age,female,#ed*female$
```

```
-----------------------------------------------------------------------------
Ordinary     least squares regression ............
LHS=INCOME   Mean                   =      .23862
             Standard deviation     =      .13027
----------   No. of observations    =          14  DegFreedom   Mean square
Regression   Sum of Squares         =     .179633           4        .04491
Residual     Sum of Squares         =  .409777E-01          9        .00455
Total        Sum of Squares         =     .220611          13        .01697
----------   Standard error of e    =      .06748  Root MSE          .05410
Fit          R-squared              =      .81425  R-bar squared     .73170
Model test   F[  4,     9]          =     9.86328  Prob F > F*       .00239
Model was estimated on Apr 11, 2013 at 09:44:56 AM
--------+--------------------------------------------------------------------
        |                Standard            Prob.      95% Confidence
  INCOME|  Coefficient     Error       t    |t|>T*         Interval
--------+--------------------------------------------------------------------
Constant|    -.23366**     .10205    -2.29  .0478     -.46452   -.00281
     AGE|     .01091***    .00222     4.92  .0008      .00589    .01592
  FEMALE|     .06630       .04543     1.46  .1785     -.03648    .16907
Interaction of category ED      and FEMALE
   1    |    -.32637***    .05752    -5.67  .0003     -.45649   -.19625
   2    |    -.01891       .08408     -.22  .8271     -.20910    .17128
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
```

Figure 24. Interaction of Categorical Variable with Other Variable

A Tip: It is easy to create multicollinearity with category variables and interactions. In all cases, *NLOGIT* will do its best to compute the regression you specify. *NLOGIT* will *never*, upon detecting multicollinearity, drop some variables and fit some model that you did not specify that does not have a multicollinearity problem. Decisions about model specification are made by you, not the program.

E. Panel Data

All panel data applications are handled the same way. To set up the procedures, you will prepare an indicator variable that *NLOGIT* will use to manage the data handling. Our 14 observation,IncomeData file is a panel, as can be seen from the ID variable in Figure 25. *NLOGIT* assumes that a panel data set contains some kind of identifier variable such as ID in Figure 25. The ID variable does not have to be a sequential set of integers. It can be anything (it need not even be integers), so long as it takes the same value for every observation in a group and it changes (up or down) from one group to the next. To set up a panel,at the beginning of your session, use

SETPANEL ; Group = the id variable ; Pds = name for a variable that *NLOGIT* will now create $

The Pds variable will contain in each row of a group the number of observations in the group. You may use any name you wish. We usually use Ti. Figure 25 shows the results of

SETPANEL ; Group = id ; Pds = Ti $

NOTE: SETPANEL should be issued immediately after the data are imported.

A Tip: This works the same way for balanced or unbalanced panels. You need not worry about unbalanced panels.

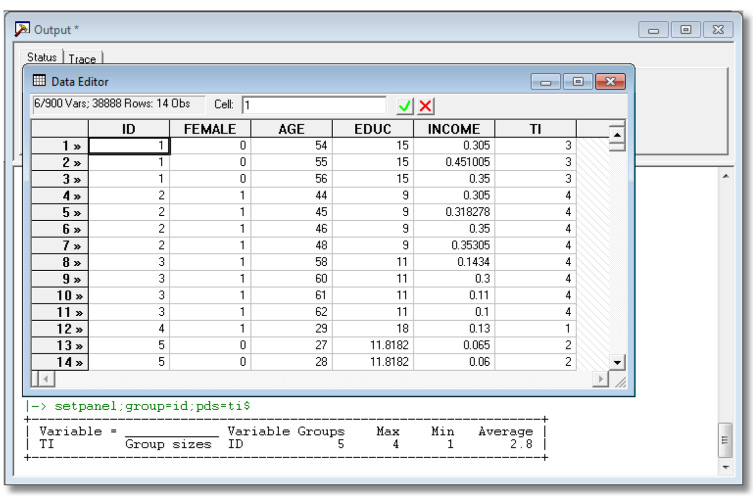
A Second Tip: If you have a balanced panel with *T*periods (whatever *T* is) and you don't have an ID variable, you can create one with

CREATE ; MyID = Trn(T,0) $ For example, Trn(10,0)

A third Tip: If you have an unbalanced panel and you do not have an ID variable, you cannot use this data set as a panel. You must create the ID variable somehow. *NLOGIT* cannot do it for you.

A Last Tip: The **SETPANEL** setting is not etched into the project. When you save the project, **SETPANEL** is not saved. When you reload the project, you must reissue the **SETPANEL** command.

SETPANEL creates some internal settings as well. Most panel versions of models are requested by just adding **;PANEL** to the model command. If you change the sample from what it was when you issued the **SETPANEL** command, this will break the counter variable. Not to worry. When your command contains **; PANEL**, *NLOGIT* recreates the counter so that it matches the observations in the active sample. In our example, if we were to **REJECT;Age>62$,** the count for group 3 would be incorrect – it would change from 4 to 3. **SETPANEL** takes care of this as it processes your model commands.


Output *
Status | Trace
Data Editor
6/900 Vars; 38888 Rows: 14 Obs   Cell: 1

	ID	FEMALE	AGE	EDUC	INCOME	TI
1 »	1	0	54	15	0.305	3
2 »	1	0	55	15	0.451005	3
3 »	1	0	56	15	0.35	3
4 »	2	1	44	9	0.305	4
5 »	2	1	45	9	0.318278	4
6 »	2	1	46	9	0.35	4
7 »	2	1	48	9	0.35305	4
8 »	3	1	58	11	0.1434	4
9 »	3	1	60	11	0.3	4
10 »	3	1	61	11	0.11	4
11 »	3	1	62	11	0.1	4
12 »	4	1	29	18	0.13	1
13 »	5	0	27	11.8182	0.065	2
14 »	5	0	28	11.8182	0.06	2

```
|-> setpanel;group=id;pds=ti$
+-----------------------------------------------------------+
| Variable = ___________ Variable Groups    Max    Min   Average |
| TI         Group sizes  ID            5      4      1      2.8 |
+-----------------------------------------------------------+
```

Figure 25. Panel Data

After the panel data set is defined with **SETPANEL**, the panel data versions of most models are invoked just by adding **;Panel** to the command, as shown in Figure 26.

```
Untitled 1 *
Insert Name:
SETPANEL ; Group = id ; Pds = ti $
REGRESS  ; Lhs = income ; Rhs = one,age,female;panel $
```

Figure 26. Panel Data Regression Command

The default linear panel data model produces quite a lot of results – it displays all three of the pooled model, fixed effects and random effects estimates. The Figures 27 and 28 show the preliminary results and the fixed effects results for our small data set. You can specialize the command with

REGRESS ; Lhs = income ; Rhs = one,age,female ; Panel ; Fixed Effects $

and likewise for random effects. All of the panel data models in *NLOGIT* (there are about 50) provide several versions (e.g., fixed vs. random effects) that are requested by adding **;Panel** and an additional specification in the model command.

```
|-> SETPANEL ; Group = id ; Pds = ti $
+----------------------------------------------------------------------+
| Variable = ____________ Variable Groups     Max     Min     Average |
| TI         Group sizes  ID             5       4       1         2.8 |
+----------------------------------------------------------------------+
|-> REGRESS  ; Lhs = income ; Rhs = one,age,female;panel $
+----------------------------------------------------------------------+
| Variable = ____________ Variable Groups     Max     Min     Average |
| TI         Group sizes  ID             5       4       1         2.8 |
+----------------------------------------------------------------------+
+------------------------------------------------------+
| Frequency count for group sizes of TI                |
| Group size =   1   Pct =  20.00%   CumPct =  20.00%  |
| Group size =   2   Pct =  20.00%   CumPct =  40.00%  |
| Group size =   3   Pct =  20.00%   CumPct =  60.00%  |
| Group size =   4   Pct =  40.00%   CumPct = 100.00%  |
+------------------------------------------------------+
```

Figure 27. Preliminary Report for Panel Data Model

F. Robust Covariance Matrices and Cluster Corrections

We mention this feature separately because it is so common in the contemporary literature. So called robust covariance matrices for least squares and maximum likelihood estimators are requested by using

; Robust

in the model command.

A Tip: The 'robust' for the linear model in a cross section is the White estimator, which is requested (only for the linear model) with **;Heteroscedasticity**. For time series, the Newey-West estimator is requested with **;Pds=T** *without* ***;Panel***.

```
--------------------------------------------------------------------------------
LSDV         least squares with fixed effects ....
LHS=INCOME   Mean                   =        .23862
             Standard deviation     =        .13027
----------   No. of observations    =            14  DegFreedom   Mean square
Regression   Sum of Squares         =       .181809          6         .03030
Residual     Sum of Squares         =   .388014E-01          7         .00554
Total        Sum of Squares         =       .220611         13         .01697
----------   Standard error of e    =        .07445  Root MSE          .05265
Fit          R-squared              =        .82412  R-bar squared     .67336
Model test   F[  6,     7]          =       5.46657  Prob F > F*       .02107
Estd. Autocorrelation of e(i,t)     =     -.511912
--------------------------------------------------
Panel:Groups Empty       0,     Valid data       5
             Smallest    1,     Largest          4
             Average group size in panel      2.80
Variances    Effects a(i)       Residuals e(i,t)
                 .016226                   .005543
Rho squared: Residual variation due to ai  .745367
Within groups variation in INCOME           .0388
R squared based on within group variation  .000178
Between group variation in INCOME           .1818
**************************************************
These   1 variables have no within group variation.
FEMALE
They are  not included in the fixed effects model.
--------+-----------------------------------------------------------------------
        |                Standard            Prob.       95% Confidence
  INCOME|  Coefficient     Error        t   |t|>T*          Interval
--------+-----------------------------------------------------------------------
     AGE|     .00059       .01665      .04  .9724      -.03568     .03686
  FEMALE|        0.0    .....(Fixed Parameter).....
--------+-----------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
Fixed parameter ... is constrained to equal the value or
had a nonpositive st.error because of an earlier problem.
--------------------------------------------------------------------------------
```

Figure 28. Results for Panel Data Model

The correction for clustering is applied in panel data sets (or clustered data sets that look like panels). All model commands are modified the same way:

; Cluster = an identity variable such as ID in figures 29 and 30

or **; Cluster = a fixed cluster size if all clusters are the same size, e.g., ; Cluster = 5.**

```
Untitled 1 *
Insert Name:
REGRESS   ; Lhs = income ; Rhs = one,age,female
          ; Cluster = id $
```

Figure 29. Cluster Correction in Regression

```
|-> REGRESS  ; Lhs = income ; Rhs = one,age,female
    ; Cluster = id $
+--------------------------------------------------------------------+
| Covariance matrix for the model is adjusted for data clustering.   |
| Sample of     14 observations contained       5 clusters defined by |
| variable ID        which identifies by a value a cluster ID.       |
+--------------------------------------------------------------------+

--------------------------------------------------------------------------------
Ordinary     least squares regression ............
LHS=INCOME   Mean                   =        .23862
             Standard deviation     =        .13027
----------   No. of observations    =            14  DegFreedom   Mean square
Regression   Sum of Squares         =   .326759E-01          2         .01634
Residual     Sum of Squares         =       .187935         11         .01708
Total        Sum of Squares         =       .220611         13         .01697
----------   Standard error of e    =        .13071  Root MSE          .11586
Fit          R-squared              =        .14812  R-bar squared    -.00677
Model test   F[  2,    11]          =        .95628  Prob F > F*       .41409
Model was estimated on Apr 20, 2013 at 05:38:09 PM
--------+-----------------------------------------------------------------------
        |                Standard            Prob.       95% Confidence
  INCOME|  Coefficient     Error        t   |t|>T*          Interval
--------+-----------------------------------------------------------------------
Constant|     .06304       .22103      .29  .7808      -.42344     .54953
     AGE|     .00416       .00447      .93  .3716      -.00567     .01400
  FEMALE|    -.03815       .11343     -.34  .7430      -.28782     .21152
--------+-----------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
--------------------------------------------------------------------------------
```

Figure 30. Results for Cluster Correction of Standard Errors for a Model

A Tip: **;Cluster** is supported for every model that is estimated using least squares or maximum likelihood estimation. It is not supported for quantile estimators or nonparametric estimators.

VII. Econometric Models

There are several hundred model specifications supported by *NLOGIT*. The set has roughly 70 basic forms such as linear regression, Poisson, Logit, Tobit, and so on. Nearly all of the basic specifications support multiple variants and extensions and about 50 also support several different panel data treatments. For example, Poisson also includes 5 forms of negative binomial models and several additional forms of count models, as well as fixed effects, random effects, latent class and random parameters specifications. Probit is the basic binary choice model, but you can also choose among 5 other forms including Logit, Arctangent, Weibull, Complementary log log, and some exotic forms that few people have ever heard of but are useful for studying the behavior of binary choice estimators. The list below will show the commands for some of the most common and familiar models. In each case, there are many variants described in the manual. And, of course, there are the hundreds of additional models. All of the models listed below are contained in both *LIMDEP* and*NLOGIT*. In each case there are many options that can be added to the model command. The list below shows a few in each case.

In the discussions to follow, we will present some examples based on a larger, 'real' data set named HealthData.csv. This is a subset of a larger data set from a health economics study by Riphahn, Wambach and Million that appeared in the *Journal of Applied Econometrics* in 2003. The original panel data set contains 27,326 observations on 7,293 households. Our subset contains 2,039 observations on 550 households. The data are imported with the usual command

IMPORT;File="…HealthData.csv"$

The discussions below show the results of various commands that illustrate the models. There is a script file for you to use to enter the commands by highlighting them one at a time. Use File→Open… and navigate to HealthData.lim to open the file in its own Text/Command window.

A. Essential Models: Estimation Commands

These are some of the most commonly used models and data analysis tools:

1. Descriptive statistics

DSTAT	**; Rhs = the list of variables $**
Useful options	**; Output = 2** requests a correlation matrix
	; Str = categorical variable requests statistics by strata
	; Quantiles requests order statistics for each variable
	**; Rhs = ***requests results for all variables.
Example:	**DSTAT ; Rhs = * $**

Descriptive Statistics for 15 variables

Variable	Mean	Std.Dev.	Minimum	Maximum	Cases	Missing
ID	271.2604	156.8122	1.0	550.0	2039	0
AGE	43.84649	10.52561	25.0	64.0	2039	0
EDUC	11.52332	2.348033	7.0	18.0	2039	0
FEMALE	.468367	.499121	0.0	1.0	2039	0
MARRIED	.709171	.454256	0.0	1.0	2039	0
HHKIDS	.383031	.486245	0.0	1.0	2039	0
INCOME	.343503	.166108	.040000	2.0	2039	0
DOCVIS	4.022070	7.893551	0.0	82.0	2039	0
HOSPVIS	.165277	1.406089	0.0	48.0	2039	0
PUBLIC	.892594	.309704	0.0	1.0	2039	0
ADDON	.008828	.093564	0.0	1.0	2039	0
DOCTOR	.627759	.483521	0.0	1.0	2039	0
HOSPITAL	.084846	.278720	0.0	1.0	2039	0
HEALTHY	.588033	.492310	0.0	1.0	2039	0
HLTHSAT	2.423737	1.123810	0.0	4.0	2039	0

DSTAT results are matrix LASTDSTA in current project.

Figure 31. Results for DSTAT

2. Scatter plot

PLOT	**; Lhs = variable on horizontal axis**
	; Rhs = variable(s) on vertical axis $
Useful options:	**; Title=Up to 80 characters for title**
	; Vaxis=Up to 60 characters for vertical axis
	; Grid to request background grid
	; Fill to request lines to connect dots in plot
	; Regression to display regression line of Rhs variable on Lhs variable
Example:	**PLOT ;if[Income <= 1.25]**
	;Lhs=educ
	;Rhs=income
	;Title=Income vs. Education (Income Under 1.25)
	;Grid ; Regression $

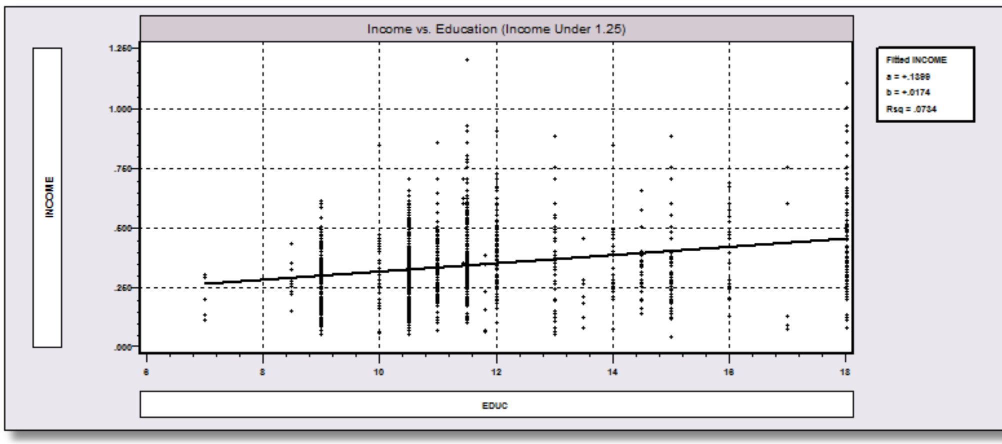


Figure 32. Scatter plot with Regression

3. Histogram

HISTOGRAM	**; Rhs = the variable $**
Useful options	**; Title=up to 80 characters for title**
	; Group = a categorical variable that defines up to 5 groups
Example:	**HISTOGRAM ; if[income <= 1.25] ; Rhs = hlthsat**
	; Title=Health Satisfaction by Gender
	; Group = Female ; Labels=Male,Female $

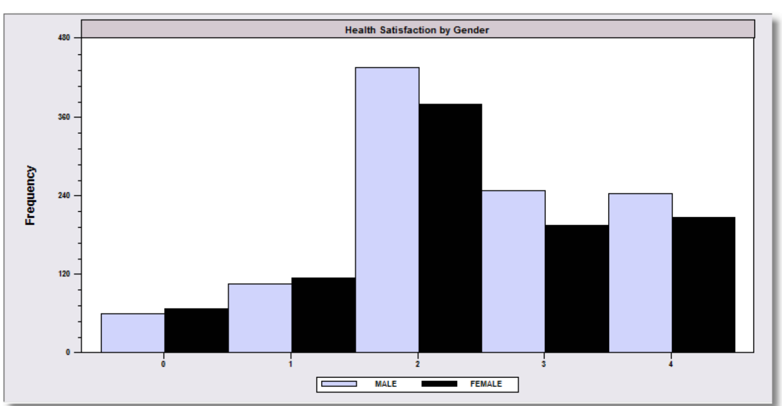


Figure 33. Histogram for Two Groups

4. Kernel Density Estimator

KERNEL	**; Rhs = list of variable(s) (up to 5) $**	
Useful options	**; Normal** – plots normal density with same mean and variance	
	; Title=up to 80 characters for title	
Example:	**KERNEL**	**; if[income <= 1.25] ; Rhs = Income**
		; Title=Income by Gender
		; Group = Female ; Labels=Male,Female $

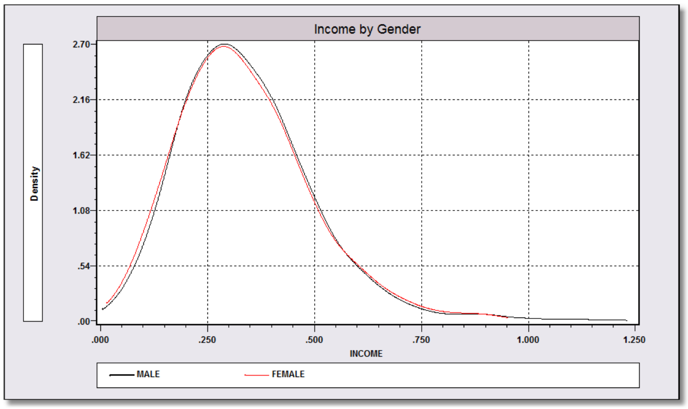


Figure 34. Kernel Density Estimators

5. Linear Regression

REGRESS	**; Lhs = dependent variable**	
	; Rhs = independent variables (include constant term ONE on Rhs) $	
Useful options	**; Cluster = specification**	
	; Heteroscedasticity to request White estimator	
	; Plot to request a plot of residuals	
	; Test: restrictions.	
	; Test:list of variables tests the hypothesis that the coefficients are all zero	
Example:	**REGRESS**	**; Lhs = income**
		; Rhs = one,age,educ,married,female,hhkids
		; Cluster = id $

```
+---------------------------------------------------------------------+
| Covariance matrix for the model is adjusted for data clustering.    |
| Sample of    2039 observations contained     550 clusters defined by |
| variable ID        which identifies by a value a cluster ID.         |
+---------------------------------------------------------------------+
-----------------------------------------------------------------------------
Ordinary    least squares regression ............
LHS=INCOME  Mean                 =        .34350
            Standard deviation   =        .16611
----------  No. of observations  =          2039  DegFreedom   Mean square
Regression  Sum of Squares       =       8.05473          5        1.61095
Residual    Sum of Squares       =       48.1775       2033         .02370
Total       Sum of Squares       =       56.2322       2038         .02759
----------  Standard error of e  =        .15394  Root MSE          .15371
Fit         R-squared            =        .14324  R-bar squared     .14113
Model test  F[  5,   2033]       =      67.97893  Prob F > F*       .00000
--------+--------------------------------------------------------------------
        |                  Standard            Prob.      95% Confidence
  INCOME|  Coefficient       Error       z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
Constant|    .03270         .04147      .79   .4304     -.04857    .11397
     AGE|   -.00024         .00049     -.49   .6230     -.00120    .00072
    EDUC|    .02094***      .00263     7.97   .0000      .01579    .02609
 MARRIED|    .10944***      .01441     7.59   .0000      .08119    .13768
  FEMALE|    .01079         .01218      .89   .3756     -.01308    .03465
  HHKIDS|   -.00684         .00973     -.70   .4819     -.02592    .01223
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

Figure 35. Linear Regression with Cluster Corrected Standard Errors

6. Instrumental Variables – 2SLS

2SLS **; Lhs = dependent variable ; Rhs = all right hand side variables**
; Inst = list of all exogenous variables including all exogenous variables (and ONE) that are in the Rhs list plus any instrumental variables not in the model $

Useful options: **; Cluster**

Example: **2SLS ; Lhs = income ; Rhs = one,age,educ,hlthsat**
; Inst = one,age,educ,married,hhkids $

```
-----------------------------------------------------------------------------
Two stage    least squares regression ...........
LHS=INCOME   Mean                  =         .34350
             Standard deviation    =         .16611
             Number of observs.    =           2039
Model size   Parameters            =              4
             Degrees of freedom    =           2035
Residuals    Sum of squares        =        421.241
             Standard error of e   =         .45497
Fit          R-squared             =       -6.50582
             Adjusted R-squared    =       -6.51689
Not using OLS or no constant. Rsqrd & F may be < 0
Instrumental Variables:
ONE       AGE       EDUC       MARRIED    HHKIDS
--------+--------------------------------------------------------------------
        |                  Standard            Prob.       95% Confidence
  INCOME|  Coefficient       Error       z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
Constant|   1.16576***      .31967      3.65  .0003       .53923   1.79229
     AGE|   -.00830***      .00279     -2.97  .0030     -.01377   -.00283
    EDUC|    .04170***      .00838      4.97  .0000      .02527    .05813
 HLTHSAT|   -.38735***      .11547     -3.35  .0008     -.61366   -.16104
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

Figure 36. Two Stage Least Squares

7. Binary Choice

PROBIT or LOGIT **; Lhs = dependent variable**
; Rhs = independent variables $

Useful options **; Hold** requests results be retained for use by **SELECTION** in the next step

Example: **SETPANEL ; Group = id ; Pds = Ti $**
PROBIT ; Lhs = doctor
; Rhs = one,age,educ,married,female,income
; Panel ; RandomEffects
; Test: income $

```
+----------------------------------------------------------------------+
| Variable = ____________ Variable Groups    Max    Min    Average |
| TI          Group sizes  ID              550      7      1        3.7 |
+----------------------------------------------------------------------+
+---------------------------------------------------------+
| Frequency count for group sizes of TI                   |
| Group size =    1   Pct =  20.36%   CumPct =  20.36%   |
| Group size =    2   Pct =  12.55%   CumPct =  32.91%   |
| Group size =    3   Pct =  12.36%   CumPct =  45.27%   |
| Group size =    4   Pct =  17.64%   CumPct =  62.91%   |
| Group size =    5   Pct =  14.55%   CumPct =  77.45%   |
| Group size =    6   Pct =  12.91%   CumPct =  90.36%   |
| Group size =    7   Pct =   9.64%   CumPct = 100.00%   |
+---------------------------------------------------------+
Normal exit:    4 iterations. Status=0, F=     1296.763
```

```
-----------------------------------------------------------------------------
Binomial Probit Model
Dependent variable                   DOCTOR
Log likelihood function         -1296.76286
Restricted log likelihood       -1346.02091
Chi squared [   5 d.f.]            98.51611
Significance level                   .00000
McFadden Pseudo R-squared          .0365953
Estimation based on N =     2039, K =     6
Inf.Cr.AIC  =    2605.5 AIC/N =       1.278
Hosmer-Lemeshow chi-squared =      16.57925
P-value=   .03480 with deg.fr. =          8
------ LM test for Random Effects -----
ChiSqd.[1]    84.927     P value     .00000
--------+--------------------------------------------------------------------
        |                  Standard            Prob.       95% Confidence
  DOCTOR|  Coefficient       Error       z     |z|>Z*         Interval
--------+--------------------------------------------------------------------
        |Index function for probability
Constant|     -.17065        .20921     -.82  .4147     -.58068     .23939
     AGE|      .01585***     .00280     5.66  .0000      .01036     .02134
    EDUC|     -.03447***     .01304    -2.64  .0082     -.06003    -.00891
 MARRIED|      .12723*       .06718     1.89  .0582     -.00443     .25890
  FEMALE|      .35218***     .05858     6.01  .0000      .23738     .46699
  INCOME|     -.12679        .18613     -.68  .4958     -.49161     .23802
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

```
-----------------------------------------------------------------------------
Random Effects Binary Probit Model
Dependent variable                   DOCTOR
Log likelihood function         -1180.67924
Restricted log likelihood       -1296.76286
Chi squared [   1 d.f.]           232.16724
Significance level                   .00000
(Cannot compute pseudo R2.  Use RHS=one
to obtain the required restricted logL)
Estimation based on N =     2039, K =     7
Inf.Cr.AIC  =    2375.4 AIC/N =       1.165
Unbalanced panel has      550 individuals
- ChiSqd[1] tests for random effects  -
LM    ChiSqd    84.927    P value    .00000
LR    ChiSqd   232.167    P value    .00000
Wald ChiSqd   166.237    P value    .00000
Wald test of       1 linear restrictions
Chi-squared =      .02, P value =   .8778
--------+--------------------------------------------------------------------
        |                  Standard            Prob.       95% Confidence
  DOCTOR|  Coefficient       Error       z     |z|>Z*         Interval
--------+--------------------------------------------------------------------
Constant|     -.34308         .38114     -.90  .3680    -1.09009     .40394
     AGE|      .02426***      .00519     4.68  .0000      .01409     .03443
    EDUC|     -.04678*        .02430    -1.92  .0543     -.09441     .00086
 MARRIED|      .03380         .11087      .30  .7605     -.18350     .25110
  FEMALE|      .51913***      .11488     4.52  .0000      .29396     .74430
  INCOME|      .04731         .30761      .15  .8778     -.55559     .65022
     Rho|      .48363***      .03751    12.89  .0000      .41011     .55715
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

Figure 37. Random Effects Probit with Test of Restriction

Example LOGIT ; Lhs = doctor
; Rhs = one,age,educ,married,female,income
; ROC $

```
-----------------------------------------------------------------------------
Binary Logit Model for Binary Choice
Dependent variable               DOCTOR
Log likelihood function     -1296.68925
Restricted log likelihood   -1346.02091
Chi squared [   5 d.f.]        98.66333
Significance level               .00000
McFadden Pseudo R-squared      .0366500
Estimation based on N =   2039, K =   6
Inf.Cr.AIC  =    2605.4 AIC/N =   1.278
Hosmer-Lemeshow chi-squared =  17.88908
P-value=  .02207 with deg.fr. =       8
--------+--------------------------------------------------------------------
        |                  Standard            Prob.      95% Confidence
  DOCTOR|  Coefficient       Error       z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
Constant|    -.27591        .33808     -.82  .4144     -.93854     .38673
     AGE|     .02584***     .00459     5.63  .0000      .01685     .03484
    EDUC|    -.05606***     .02104    -2.66  .0077     -.09730    -.01482
 MARRIED|     .20583*       .10931     1.88  .0597     -.00841     .42008
  FEMALE|     .57693***     .09603     6.01  .0000      .38872     .76515
  INCOME|    -.22760        .30084     -.76  .4493     -.81723     .36202
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

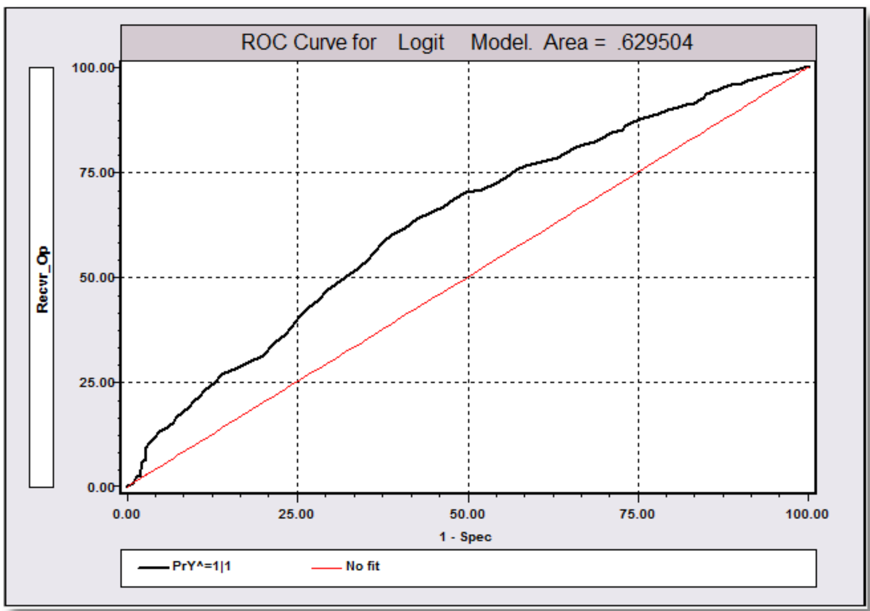


Figure 38. LOGIT Model with Receiver Operating Curve

8. Count Data

POISSON	**; Lhs = dependent variable**
	; Rhs = independent variables $
Useful options	**; Exposure = exposure variable**
NEGBIN	**; same as Poisson for negative binomial model**
Useful options	**; Model = NB1 or NB2 or NBP**
Example:	**NEGBIN ; Lhs = docvis ; Rhs = one,age,educ,married,income $**

```
-----------------------------------------------------------------------------
Poisson Regression
Dependent variable                DOCVIS
Log likelihood function     -10167.88134
Restricted log likelihood   -10730.14514
Chi squared [   4 d.f.]       1124.52759
Significance level                .00000
McFadden Pseudo R-squared       .0524004
Estimation based on N =   2039, K =    5
Inf.Cr.AIC  =  20345.8 AIC/N =     9.978
Chi- squared = 27891.27736  RsqP= .1166
G  - squared = 16229.37401  RsqD= .0648
Overdispersion tests: g=mu(i)  :  8.083
Overdispersion tests: g=mu(i)^2:  8.365
--------+--------------------------------------------------------------------
        |                Standard            Prob.       95% Confidence
  DOCVIS|  Coefficient     Error       z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
Constant|    1.59095***    .08991    17.70  .0000     1.41474    1.76717
     AGE|     .02110***    .00107    19.74  .0000      .01901     .02320
    EDUC|    -.10607***    .00631   -16.80  .0000     -.11845    -.09370
 MARRIED|     .25240***    .02797     9.02  .0000      .19758     .30722
  INCOME|    -.45474***    .08180    -5.56  .0000     -.61506    -.29441
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

```
Normal exit:   5 iterations. Status=0, F=     4820.676
For NB model, null logL=  -4820.6762
Normal exit:  11 iterations. Status=0, F=     4763.008
-----------------------------------------------------------------------------
Negative Binomial Regression
Dependent variable                DOCVIS
Log likelihood function      -4763.00834
Restricted log likelihood   -10167.88134
Chi squared [   1 d.f.]      10809.74601
Significance level                .00000
McFadden Pseudo R-squared       .5315633
Estimation based on N =   2039, K =    6
Inf.Cr.AIC  =   9538.0 AIC/N =     4.678
NegBin form 2; Psi(i) = theta
Tests of Model Restrictions on Neg.Bin.
Model               Logl ChiSquared[df]
Poisson(b=0)   -10730.15  ******** [**]
Poisson        -10167.88    1124.5 [ 4]
NB v. Poisson   -4763.01   10809.7 [ 1]
NB (b=0)        -4820.68     115.3 [ 4]
--------+--------------------------------------------------------------------
        |                Standard            Prob.       95% Confidence
  DOCVIS|  Coefficient     Error       z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
Constant|    1.71407***    .24362     7.04  .0000     1.23658    2.19157
     AGE|     .02081***    .00320     6.50  .0000      .01453     .02709
    EDUC|    -.10644***    .01462    -7.28  .0000     -.13510    -.07778
 MARRIED|     .21359***    .07619     2.80  .0051      .06426     .36293
  INCOME|    -.68631**     .29195    -2.35  .0187    -1.25852    -.11410
        |Dispersion parameter for count data model
   Alpha|    2.29507***    .08590    26.72  .0000     2.12670    2.46343
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

Figure 39. Negative Bonomial Model with Poisson Starting Values

9. Ordered Choice Models

ORDERED	**; Lhs = dependent variable**
or OPROBIT	**; Rhs = independent variables $ (ONE must be the first Rhs variable)**
Useful options	**; LOGIT** for ordered logit. Verb **OLOGIT** is the same as **ORDERED;Logit**.
Example:	**ORDERED ; Lhs = hlthsat**
	; Rhs = one,age,educ,income,female
	; Partials ; Full $

```
-----------------------------------------------------------------------------
Ordered Probability Model
Dependent variable              HLTHSAT
Log likelihood function     -2861.66224
Restricted log likelihood   -2934.80920
Chi squared [   4 d.f.]       146.29392
Significance level               .00000
McFadden Pseudo R-squared      .0249239
Estimation based on N =   2039, K =   8
Inf.Cr.AIC  =   5739.3 AIC/N =   2.815
Model estimated: Apr 21, 2013, 09:36:39
Underlying probabilities based on Normal
--------+--------------------------------------------------------------------
        |                Standard            Prob.       95% Confidence
 HLTHSAT|  Coefficient      Error       z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
        |Index function for probability
Constant|    1.90355***     .17022    11.18  .0000     1.56992   2.23718
     AGE|    -.02238***     .00227    -9.85  .0000     -.02684   -.01793
    EDUC|     .05454***     .01063     5.13  .0000      .03371    .07537
  INCOME|     .21960        .14649     1.50  .1338     -.06751    .50671
  FEMALE|    -.04029        .04789     -.84  .4002     -.13415    .05357
        |Threshold parameters for index
  Mu(01)|     .60073***     .02878    20.88  .0000      .54433    .65713
  Mu(02)|    1.78224***     .02793    63.81  .0000     1.72750   1.83698
  Mu(03)|    2.41924***     .03222    75.08  .0000     2.35609   2.48239
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

```
+----------------------------------------------------------------------------+
|                  CELL FREQUENCIES FOR ORDERED CHOICES                      |
+----------------------------------------------------------------------------+
|                 Frequency        Cumulative  < =      Cumulative  > =      |
|Outcome        Count   Percent   Count   Percent     Count   Percent        |
|----------- ------- ---------  ------- ---------   ------- ---------        |
|HLTHSAT=00      125    6.1305      125    6.1305      2039  100.0000        |
|HLTHSAT=01      217   10.6425      342   16.7729      1914   93.8695        |
|HLTHSAT=02      812   39.8234     1154   56.5964      1697   83.2271        |
|HLTHSAT=03      439   21.5302     1593   78.1265       885   43.4036        |
|HLTHSAT=04      446   21.8735     2039  100.0000       446   21.8735        |
+----------------------------------------------------------------------------+
```

```
Cross tabulation of predictions and actual outcomes
+------+-----+-----+-----+-----+-----+-----+
|y(i,j)|  0  |  1  |  2  |  3  |  4  |Total|
+------+-----+-----+-----+-----+-----+-----+
|  0   |    0|    0|  122|    0|    3|  125|
|  1   |    0|    0|  205|    0|   12|  217|
|  2   |    0|    0|  751|    0|   61|  812|
|  3   |    0|    0|  380|    0|   59|  439|
|  4   |    0|    0|  391|    0|   55|  446|
+------+-----+-----+-----+-----+-----+-----+
| Total|    0|    0| 1849|    0|  190| 2039|
+------+-----+-----+-----+-----+-----+-----+
Row = actual, Column = Prediction, Model = Probit
Prediction is number of the most probable cell.

Cross tabulation of outcomes and predicted probabilities.
+------+-----+-----+-----+-----+-----+-----+
|y(i,j)|  0  |  1  |  2  |  3  |  4  |Total|
+------+-----+-----+-----+-----+-----+-----+
|  0   |    9|   15|   53|   25|   22|  125|
|  1   |   16|   26|   90|   44|   41|  217|
|  2   |   54|   90|  330|  171|  166|  812|
|  3   |   23|   42|  171|   98|  105|  439|
|  4   |   23|   42|  172|  100|  109|  446|
+------+-----+-----+-----+-----+-----+-----+
| Total|  125|  215|  816|  439|  444| 2039|
+------+-----+-----+-----+-----+-----+-----+
Row = actual, Column = Prediction, Model = Probit
Value(j,m)=Sum(i=1,N)y(i,j)*p(i,m).
Column totals may not match cell sums because of rounding error.
```

Figure 40. Estimated Ordered Probit Model

```
--------------------------------------------------------------------------
Marginal effects for ordered probability model
M.E.s for dummy variables are Pr[y|x=1]-Pr[y|x=0]
Names for dummy variables are marked by *.
--------+-----------------------------------------------------------------
        |    Partial                          Prob.      95% Confidence
 HLTHSAT|     Effect     Elasticity      z    |z|>Z*         Interval
--------+-----------------------------------------------------------------
        |--------------[Partial effects on Prob[Y=00] at means]--------------
     AGE|    .00245***     1.98615     9.17  .0000       .00193     .00298
    EDUC|   -.00598***    -1.27186    -5.03  .0000      -.00831    -.00365
  INCOME|   -.02408        -.15266    -1.50  .1347      -.05563     .00747
 *FEMALE|    .00443         .08171      .84  .4015      -.00592     .01477
        |--------------[Partial effects on Prob[Y=01] at means]--------------
     AGE|    .00293***     1.24264     9.80  .0000       .00234     .00351
    EDUC|   -.00713***     -.79575    -5.12  .0000      -.00986    -.00440
  INCOME|   -.02872        -.09551    -1.50  .1339      -.06627     .00884
 *FEMALE|    .00527         .05104      .84  .4004      -.00701     .01756
        |--------------[Partial effects on Prob[Y=02] at means]--------------
     AGE|    .00341***      .36272     8.66  .0000       .00264     .00418
    EDUC|   -.00832***     -.23227    -4.94  .0000      -.01161    -.00502
  INCOME|   -.03348        -.02788    -1.49  .1351      -.07739     .01043
 *FEMALE|    .00613         .01485      .84  .3993      -.00812     .02037
        |--------------[Partial effects on Prob[Y=03] at means]--------------
     AGE|   -.00237***     -.46865    -7.86  .0000      -.00296    -.00178
    EDUC|    .00578***      .30011     4.78  .0000       .00341     .00815
  INCOME|    .02328         .03602     1.48  .1378      -.00747     .05402
 *FEMALE|   -.00428        -.01927     -.84  .4022      -.01429     .00573
        |--------------[Partial effects on Prob[Y=04] at means]--------------
     AGE|   -.00642***    -1.35331    -9.38  .0000      -.00776    -.00508
    EDUC|    .01565***      .86662     5.06  .0000       .00959     .02171
  INCOME|    .06300         .10402     1.50  .1338      -.01936     .14536
 *FEMALE|   -.01155        -.05550     -.84  .3997      -.03842     .01533
--------+-----------------------------------------------------------------
z, prob values and confidence intervals are given for the partial effect
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
--------------------------------------------------------------------------
```

```
+------------------------------------------------------------------------+
| Summary of Marginal Effects for Ordered Probability Model (probit)     |
| Effects computed at means.  Effects for binary variables (*) are       |
| computed as differences of probabilities, other variables at means.    |
| Binary variables change only by 1 unit so s.d. changes are not shown.  |
| Elasticities for binary variables=partial effect/probability = %chgP   |
+------------------------------------------------------------------------+
+------------------------------------------------------------------------+
|            Continuous Variable AGE          Changes in AGE       % chg|
|           ------------------------------   -------------------------- 
Outcome     Effect  dPy<=nn/dX dPy>=nn/dX   1 StdDev   Low to High   Elast
-------   ------------------------------   -------------------------- 
Y = 00      .00245     .00245     .00000     .02584     .09573   1.98615
Y = 01      .00293     .00538    -.00245     .03081     .11415   1.24264
Y = 02      .00341     .00879    -.00538     .03592     .13309    .36272
Y = 03     -.00237     .00642    -.00879    -.02497    -.09253   -.46865
Y = 04     -.00642     .00000    -.00642    -.06759    -.25044  -1.35331
+------------------------------------------------------------------------+
|            Continuous Variable EDUC         Changes in EDUC      % chg|
|           ------------------------------   -------------------------- 
Outcome     Effect  dPy<=nn/dX dPy>=nn/dX   1 StdDev   Low to High   Elast
-------   ------------------------------   -------------------------- 
Y = 00     -.00598    -.00598     .00000    -.01404    -.06579  -1.27186
Y = 01     -.00713    -.01311     .00598    -.01675    -.07845   -.79575
Y = 02     -.00832    -.02143     .01311    -.01952    -.09147   -.23227
Y = 03      .00578    -.01565     .02143     .01357     .06359    .30011
Y = 04      .01565     .00000     .01565     .03674     .17211    .86662
+------------------------------------------------------------------------+
|            Continuous Variable INCOME       Changes in INCOME    % chg|
|           ------------------------------   -------------------------- 
Outcome     Effect  dPy<=nn/dX dPy>=nn/dX   1 StdDev   Low to High   Elast
-------   ------------------------------   -------------------------- 
Y = 00     -.02408    -.02408     .00000    -.00400    -.04720   -.15266
Y = 01     -.02872    -.05280     .02408    -.00477    -.05628   -.09551
Y = 02     -.03348    -.08628     .05280    -.00556    -.06562   -.02788
Y = 03      .02328    -.06300     .08628     .00387     .04562    .03602
Y = 04      .06300     .00000     .06300     .01046     .12348    .10402
+------------------------------------------------------------------------+
|          Binary(0/1) Variable FEMALE        Changes in *FEMALE   % chg|
|           ------------------------------   -------------------------- 
Outcome     Effect  dPy<=nn/dX dPy>=nn/dX   1 StdDev   Low to High   Elast
-------   ------------------------------   -------------------------- 
Y = 00      .00443     .00443     .00000       -        .00443    .08171
Y = 01      .00527     .00970    -.00443       -        .00527    .05104
Y = 02      .00613     .01582    -.00970       -        .00613    .01485
Y = 03     -.00428     .01155    -.01582       -       -.00428   -.01927
Y = 04     -.01155     .00000    -.01155       -       -.01155   -.05550
--------------------------------------------------------------------------
```

Figure 41. Full Partial Effects Analysis for Ordered Probit

10. Stochastic Frontier and Data Envelopment Analysis

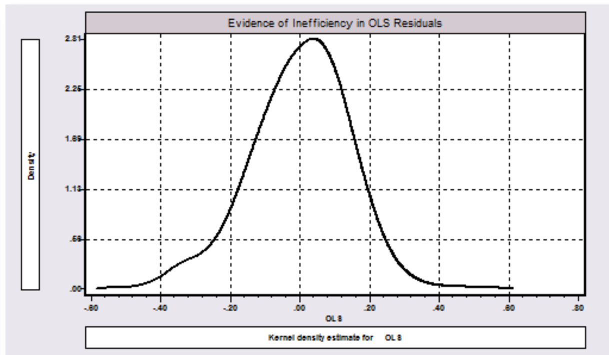
FRONTIER **; Lhs = dependent variable**
; Rhs = independent variables $ ONE must be first

Useful options **; Cost** to fit cost frontier. Production is the default
; Techeff = varible to hold estimate of technical efficiency, firm specific
; Eff = variable to hold estimate of inefficiency, firm specific
; ALG = DEA requests data envelopment analysis.

Example: This example uses a data set on production of Spanish Dairy farms.
NAMELIST ; x = one,x1,x2,x3,x4 $
REGRESS ; quietly ; Lhs=yit ; Rhs = x ; Res = ols $

```
KERNEL      ; Rhs = ols
            ; Title=Evidence of Inefficiency in OLS Residuals $
FRONTIER    ; Lhs = yit ; Rhs = x ; Techeff = eui $
KERNEL      ; Rhs = eui
            ; Title=Estimated Efficiency Distribution for Dairy Farms $
```

```
--------------------------------------
Kernel Density Estimator for OLS
Kernel Function      =      Logistic
Observations         =          1482
Points plotted       =          1482
Bandwidth            =       .029289
Statistics for abscissa values----
Mean                 =       .000000
Standard Deviation   =       .140164
Skewness             =     -.296061
Kurtosis-3 (excess)=       .802483
Chi2 normality test=      1.801944
Minimum              =     -.554019
Maximum              =       .583301
Results matrix       =        KERNEL
--------------------------------------
```




```
-----------------------------------------------------------------------------
Limited Dependent Variable Model - FRONTIER
Dependent variable                     YIT
Log likelihood function          822.68831
Estimation based on N =    1482, K =    7
Inf.Cr.AIC  =  -1631.4 AIC/N =    -1.101
Variances: Sigma-squared(v)=       .01075
           Sigma(v)         =      .10371
           Sigma-squared(u)=       .02425
           Sigma(u)         =      .15573
Sigma  = Sqr[(s^2(u)+s^2(v)]=      .18710
Gamma  = sigma(u)^2/sigma^2 =      .69277
Var[u]/{Var[u]+Var[v]}      =      .45037
Stochastic Production Frontier, e = v-u
-----[ Tests vs. No Inefficiency ]-----
LR test for inefficiency vs. OLS v only
Deg. freedom for sigma-squared(u):     1
Deg. freedom for heteroscedasticity:  0
Deg. freedom for truncation mean:     0
Deg. freedom for inefficiency model:  1
LogL when sigma(u)=0             809.67610
Chi-sq=2*[LogL(SF)-LogL(LS)] =    26.024
Kodde-Palm C*: 95%: 2.706,  99%:  5.412
LM test for sigma(u) = 0 based on ols e
Chi-sq[1]=(N/6)*[m3/s^3]^2         21.665
Wald tests based on MLEs shown in table
--------+--------------------------------------------------------------------
        |                 Standard            Prob.      95% Confidence
     YIT|  Coefficient      Error       z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
        |Deterministic Component of Stochastic Frontier Model
Constant|    11.7014***     .00447  2614.87  .0000     11.6926   11.7101
      X1|     .58369***     .01887    30.93  .0000      .54670    .62068
      X2|     .03555***     .01113     3.20  .0014      .01375    .05736
      X3|     .02256*       .01281     1.76  .0783    -.00256    .04768
      X4|     .44948***     .01035    43.42  .0000      .42919    .46977
        |Variance parameters for compound error
  Lambda|    1.50164***     .08748    17.17  .0000     1.33019   1.67310
   Sigma|     .18710***     .00011  1698.90  .0000      .18688    .18732
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

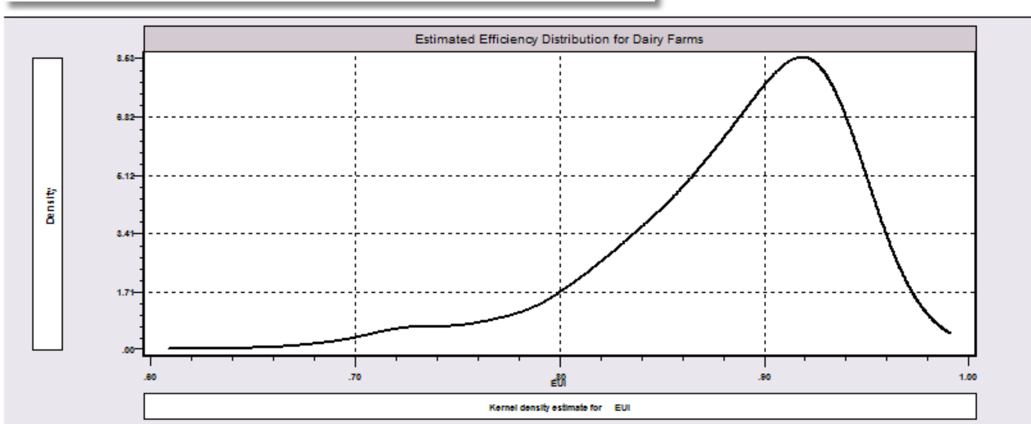


Figure 42. Stochastic Frontier Efficiency Analysis

B. Post Estimation Model Results

After estimation of the model parameters, a variety of computations are used to analyze the model results. Two common calculations are model predictions/simulations and partial effects. There are also standard results computed with model results that are retained so that they can be used in later analyses

1. Predictions

Single equation models can create a new variable that is the predictions for the model using

; Keep = the new variable.

Models differ on what the prediction is. In most cases, it is the expected value of the dependent variable. For a few models, it is also possible to retain residuals with

; Res = the new variable.

For most models, this is not a meaningful result, however. For probability models, such as **PROBIT**, **LOGIT** and **ORDERED**, the predicted probability for the observed outcome is saved with

; Prob = the new variable.

2. Simulations

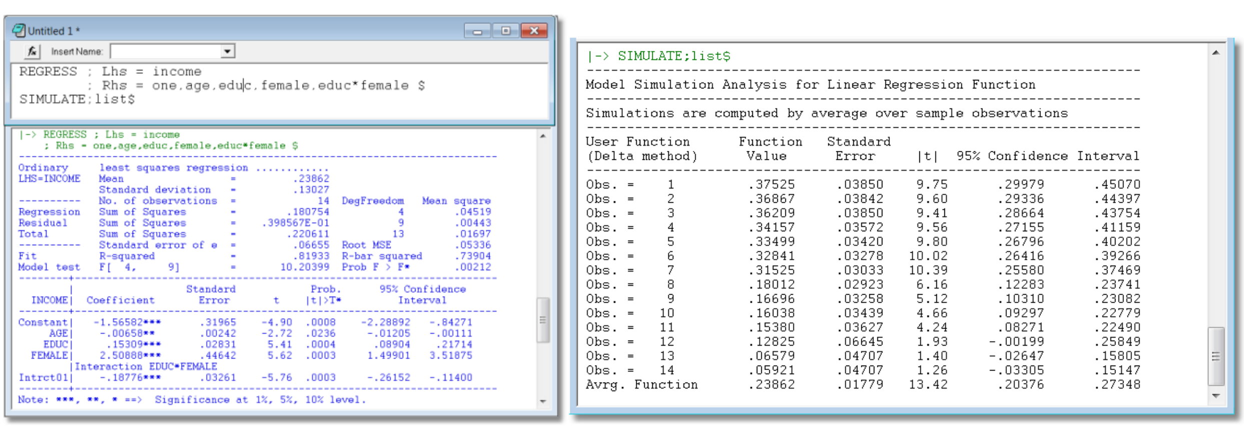
After estimation, model estimation programs store the results for two large processors to use, the simulator and the partial effects program. These use separate post estimation commands. The simple command

SIMULATE $

Produces the average prediction from the model, with an estimated standard error and confidence interval for the mean simulation. Adding **;List** to the **SIMULATE** command produces a listing of the predictions. Adding **;Keep=name** to the command requests that a new variable that contains the simulated values be created in the data set.

A Tip: If you have thousands of observations, it might not be a good idea to use **;List**. If you are trying to produce what looks like a huge list, the program will ask you if you are sure you want to do this.

Figure 43 shows estimation and simulation of a linear regression with an interaction term. The SIMULATE feature accounts for the nonlinearities in the regression. A second example based on a binary choice model appears below in Figure 44.



```
Untitled 1 *
Insert Name:
REGRESS ; Lhs = income
        ; Rhs = one,age,educ,female,educ*female $
SIMULATE;list$
```

```
|-> REGRESS ; Lhs = income
    ; Rhs = one,age,educ,female,educ*female $
-----------------------------------------------------------------------------
Ordinary     least squares regression ............
LHS=INCOME   Mean                   =       .23862
             Standard deviation     =       .13027
----------   No. of observations    =           14  DegFreedom   Mean square
Regression   Sum of Squares         =      .180754           4        .04519
Residual     Sum of Squares         =   .398567E-01          9        .00443
Total        Sum of Squares         =      .220611          13        .01697
----------   Standard error of e    =       .06655  Root MSE          .05336
Fit          R-squared              =       .81933  R-bar squared     .73904
Model test   F[  4,     9]          =     10.20399  Prob F > F*       .00212
--------+--------------------------------------------------------------------
        |                Standard            Prob.      95% Confidence
  INCOME|  Coefficient     Error        t    |t|>T*        Interval
--------+--------------------------------------------------------------------
Constant|   -1.56582***    .31965    -4.90  .0008    -2.28892   -.84271
     AGE|    -.00658**     .00242    -2.72  .0236     -.01205   -.00111
    EDUC|     .15309***    .02831     5.41  .0004      .08904    .21714
  FEMALE|    2.50888***    .44642     5.62  .0003     1.49901   3.51875
        |Interaction EDUC*FEMALE
Intrct01|    -.18776***    .03261    -5.76  .0003     -.26152   -.11400
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
```

```
|-> SIMULATE;list$
---------------------------------------------------------------------
Model Simulation Analysis for Linear Regression Function
---------------------------------------------------------------------
Simulations are computed by average over sample observations
---------------------------------------------------------------------
User Function       Function   Standard
(Delta method)       Value      Error      |t|  95% Confidence Interval
---------------------------------------------------------------------
Obs. =     1        .37525     .03850     9.75     .29979     .45070
Obs. =     2        .36867     .03842     9.60     .29336     .44397
Obs. =     3        .36209     .03850     9.41     .28664     .43754
Obs. =     4        .34157     .03572     9.56     .27155     .41159
Obs. =     5        .33499     .03420     9.80     .26796     .40202
Obs. =     6        .32841     .03278    10.02     .26416     .39266
Obs. =     7        .31525     .03033    10.39     .25580     .37469
Obs. =     8        .18012     .02923     6.16     .12283     .23741
Obs. =     9        .16696     .03258     5.12     .10310     .23082
Obs. =    10        .16038     .03439     4.66     .09297     .22779
Obs. =    11        .15380     .03627     4.24     .08271     .22490
Obs. =    12        .12825     .06645     1.93    -.00199     .25849
Obs. =    13        .06579     .04707     1.40    -.02647     .15805
Obs. =    14        .05921     .04707     1.26    -.03305     .15147
Avrg. Function      .23862     .01779    13.42     .20376     .27348
```

Figure 43. Regression Results and Simulation

3. Partial Effects

Partial effects are an essential part of model estimation. There are several issues to be considered in computing partial effects for a nonlinear model:

- Partial effects are often (correctly) computed as scaled coefficients. However, differences can arise between results computed by using the sample means of the data (PEA, or partial effects at averages) and results computed by averaging the computations across the sample (APE, or average partial effects).

- Partial effects for dummy variables should be computed as discrete differences in predicted values, not scaled coefficients (though the latter is often a surprisingly good approximation to the former).

- When there are nonlinearities in the index function of the model, such as $\boldsymbol{\beta}'\mathbf{x} = \beta_0 + \beta_1 z + \beta_2 z^2$, the program should compute a partial effect for z using the chain rule, not meaningless scaled coefficients for z and separately for z^2.

- When there are interactions in the index function model, such as $\boldsymbol{\beta}'\mathbf{x} = \beta_0 + \beta_1 Ed + \beta_2 Fem + \beta_3 Ed\times Fem$, the partial effects for *Ed* and *Fem* (or the interactions in general) should account for the interaction. There is no separate partial effect for the product term.

- Partial effects for the components of categorical variables can be analyzed in terms of transitions from one level to another, not always strictly between the categories and the base case.

NLOGIT's partial effects estimator, accessed with the command **PARTIALS**, accounts for all of these aspects. The basic command is

PARTIALS ; Effects : variable $

More than one variable can be analyzed by separating the names with slashes (not commas). An example appears in Figure 44. The model command is **PROBIT;Lhs=doctor;Rhs=one,age,educ,female,married,female*educ$**

```
|-> PARTIALS ; Effects : age / educ / female / married ; Summary $
-------------------------------------------------------------------
Partial Effects for Probit Probability Function
Partial Effects Averaged Over Observations
* ==> Partial Effect for a Binary Variable
-------------------------------------------------------------------
                    Partial    Standard
(Delta method)      Effect      Error     |t|  95% Confidence Interval
-------------------------------------------------------------------
      AGE           .00583      .00100    5.85      .00387     .00778
      EDUC        -.01290      .00454    2.84     -.02180    -.00400
   *  FEMALE        .12965      .02119    6.12      .08812     .17117
   *  MARRIED       .04179      .02376    1.76     -.00477     .08836
-------------------------------------------------------------------
```

Figure 44. Partial Effects for a PROBIT Model

Partial effects are computed by averaging across observations (average partial effects). Partial effects are computed at sample means by using **;Means**.

A Tip: In a very large sample, average partial effects can take a very long time to compute. Use **;Means**.

There are several ways to analyze scenarios with the variables in the model. The next example illustrates.

Example: **LOGIT ; Lhs = Doctor**
; Rhs = one,age,educ,income,female,age*female $
SIMULATE ; Scenario : & age=25(2)65 ;plot(ci) $
PARTIALS ; Effects : age & age=25(2)65 ;plot(ci) $

```
-----------------------------------------------------------------------------
Binary Logit Model for Binary Choice
Dependent variable                DOCTOR
Log likelihood function      -1297.88981
Restricted log likelihood    -1346.02091
Chi squared [   5 d.f.]         96.26220
Significance level                .00000
McFadden Pseudo R-squared       .0357581
Estimation based on N =   2039, K =   6
Inf.Cr.AIC  =   2607.8 AIC/N =     1.279
--------+--------------------------------------------------------------------
        |                Standard            Prob.      95% Confidence
  DOCTOR|  Coefficient     Error       z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
Constant|    -.29872       .36477    -.82  .4128    -1.01367     .41622
     AGE|     .03120***    .00605    5.15  .0000      .01933     .04307
    EDUC|    -.06590***    .02055   -3.21  .0013    -.10618    -.02561
  INCOME|    -.07425       .28968    -.26  .7977    -.64200     .49351
  FEMALE|     .98712**     .40428    2.44  .0146      .19473    1.77950
        |Interaction AGE*FEMALE
Intrct01|    -.00971       .00914   -1.06  .2882    -.02761     .00820
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

```
---------------------------------------------------------------------
Model Simulation Analysis for Logit Probability Function
---------------------------------------------------------------------
Simulations are computed by average over sample observations
---------------------------------------------------------------------
User Function         Function    Standard
(Delta method)          Value       Error     |t|   95% Confidence Interval
---------------------------------------------------------------------
Avrg. Function         .62776      .01045   60.07       .60728      .64824
AGE      = 25.00       .51082      .02273   22.48       .46628      .55537
AGE      = 27.00       .52361      .02085   25.11       .48274      .56448
AGE      = 29.00       .53641      .01903   28.19       .49912      .57370
AGE      = 31.00       .54920      .01727   31.80       .51536      .58305
AGE      = 33.00       .56198      .01562   35.99       .53137      .59258
AGE      = 35.00       .57471      .01411   40.73       .54705      .60236
AGE      = 37.00       .58738      .01280   45.87       .56229      .61248
AGE      = 39.00       .59998      .01176   51.02       .57693      .62303
AGE      = 41.00       .61248      .01104   55.46       .59084      .63413
AGE      = 43.00       .62487      .01070   58.40       .60390      .64584
AGE      = 45.00       .63713      .01074   59.30       .61607      .65818
AGE      = 47.00       .64924      .01114   58.27       .62740      .67107
AGE      = 49.00       .66119      .01183   55.90       .63800      .68437
AGE      = 51.00       .67296      .01273   52.88       .64802      .69790
AGE      = 53.00       .68454      .01376   49.74       .65756      .71151
AGE      = 55.00       .69591      .01488   46.77       .66675      .72508
AGE      = 57.00       .70707      .01603   44.10       .67565      .73850
AGE      = 59.00       .71800      .01719   41.78       .68432      .75169
AGE      = 61.00       .72870      .01832   39.78       .69280      .76460
AGE      = 63.00       .73915      .01941   38.09       .70111      .77718
AGE      = 65.00       .74934      .02044   36.66       .70927      .78941
AGE      = 67.00       .75928      .02141   35.46       .71730      .80125
```

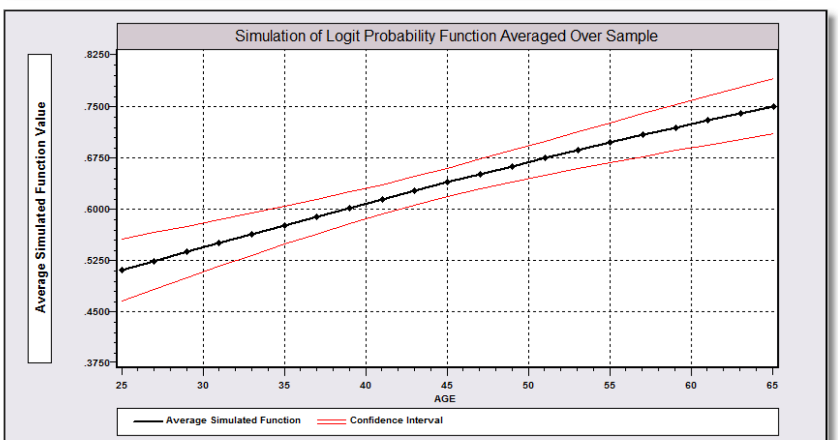


Figure 45. Estimated Binary Logit Model and Simulation

Partial Effects Analysis for Logit Probability Function

Effects on function with respect to AGE
Results are computed by average over sample observations
Partial effects for continuous AGE computed by differentiation
Effect is computed as derivative = df(.)/dx

df/dAGE (Delta method)		Partial Effect	Standard Error	|t|	95% Confidence Interval	
APE. Function		.00601	.00098	6.15	.00410	.00793
AGE	= 25.00	.00639	.00109	5.84	.00424	.00853
AGE	= 27.00	.00640	.00110	5.81	.00424	.00856
AGE	= 29.00	.00640	.00111	5.78	.00423	.00857
AGE	= 31.00	.00639	.00111	5.76	.00422	.00857
AGE	= 33.00	.00638	.00111	5.75	.00421	.00855
AGE	= 35.00	.00635	.00110	5.76	.00419	.00852
AGE	= 37.00	.00632	.00109	5.78	.00418	.00846
AGE	= 39.00	.00628	.00108	5.81	.00416	.00839
AGE	= 41.00	.00622	.00106	5.85	.00414	.00831
AGE	= 43.00	.00616	.00104	5.92	.00412	.00820
AGE	= 45.00	.00609	.00102	5.99	.00410	.00809
AGE	= 47.00	.00602	.00099	6.09	.00408	.00795
AGE	= 49.00	.00593	.00096	6.20	.00406	.00781
AGE	= 51.00	.00584	.00092	6.34	.00403	.00764
AGE	= 53.00	.00574	.00088	6.50	.00401	.00747
AGE	= 55.00	.00563	.00084	6.68	.00398	.00729
AGE	= 57.00	.00552	.00080	6.90	.00395	.00709
AGE	= 59.00	.00541	.00076	7.14	.00392	.00689
AGE	= 61.00	.00529	.00071	7.43	.00389	.00668
AGE	= 63.00	.00516	.00067	7.76	.00386	.00647
AGE	= 65.00	.00503	.00062	8.14	.00382	.00624
AGE	= 67.00	.00490	.00057	8.58	.00378	.00602

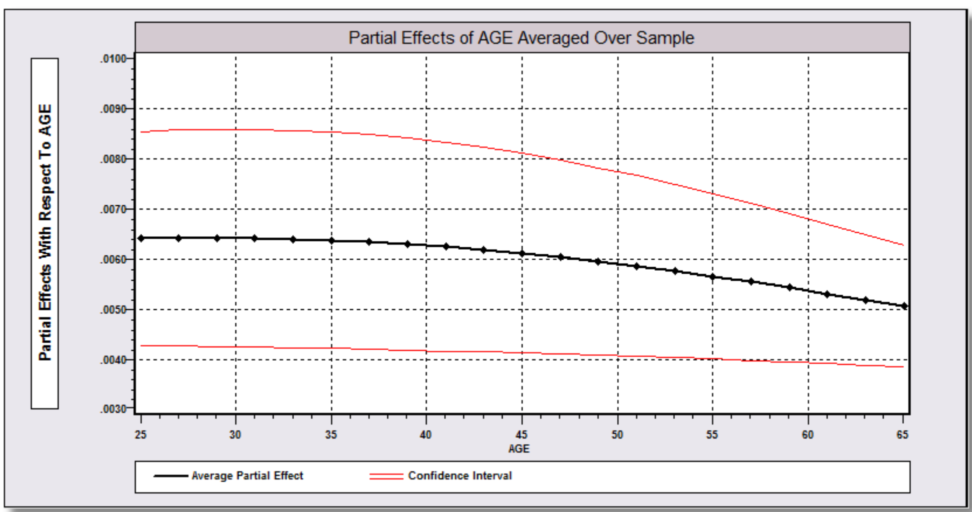


Figure 46. Average Partial Effects over Scenario for Logit Model

4. Retained Results

The **SIMULATE** and **PARTIALS** instructions use the model estimates that are stored by the estimator. Several other results are stored for later use. Matrices B and VARB (the variance of the estimator) are stored as accessible matrices. The updated project window after the probit model in Figure 48 is estimated is shown in Figure 47. Note the appearance of the coefficient vector, the covariance matrix and the scalar log likelihood.in Figre 30. The commands in Figure 31 test the hypothesis that the coefficients in the probit model are all zero using a Wald statistic. The statistic and the critical value from the chi squared table are shown in Figure 32.

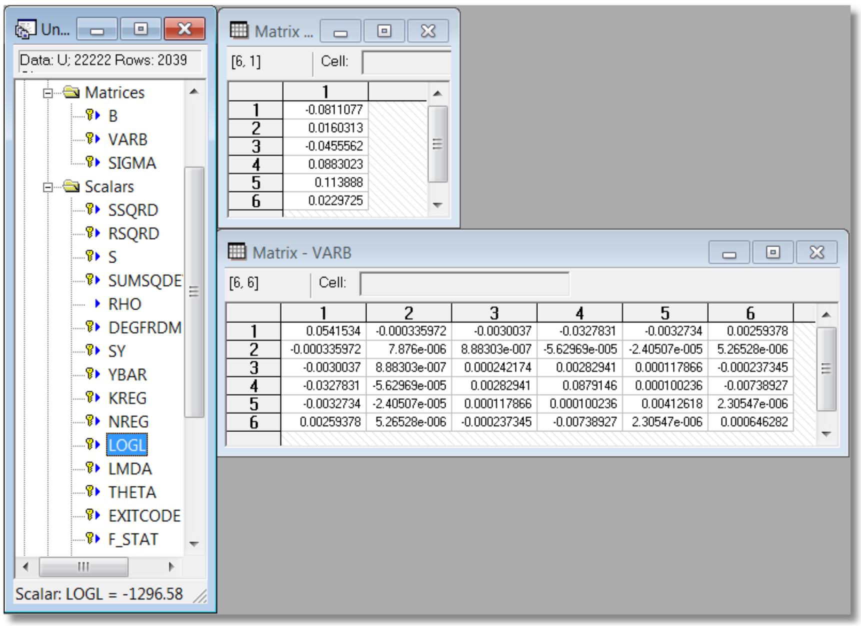


Figure 47. Stored Matrix and Scalar Results

```
NAMELIST ; x = one,age,educ,female,married,female*educ $
PROBIT   ; Lhs = doctor ; Rhs = x ; partials $
PARTIALS ; Effects : age / educ / female / married ; Summary $

MATRIX ; bp=b(2:6)
       ; vp=varb(2:6,2:6)
       ; list ; waldstat=bp'<vp>bp $
CALC   ; list ; ctb(.95,5) $
```

Figure 48. Matrix Manipulation

```
|-> MATRIX ; bp=b(2:6)
    ; vp=varb(2:6,2:6)
    ; list ; waldstat=bp'<vp>bp $

WALDSTAT|            1
--------+--------------
       1|      96.2517

|-> CALC   ; list ; ctb(.95,5) $
[CALC]         =       11.0704978
```

Figure 49. Using MATRIX and CALC to Carry Out a Test

C. Panel Data Forms

Nearly all of the models, such as REGRESS, PROBIT, LOGIT, POISSON, ORDERED, and so on. Generally, these are fixed effects, random effects, random parameters, and latent class models. (The last two of these are also useable with cross sections, but work well and naturally with panel data.) These have a variety of specifications and options all described in the program documentation. We list the basic forms here.

Panel data analysis begins with the **SETPANEL** instruction described in Section VI.E. The data must be arranged in contiguous blocks, by group. If your panel has 5,000 groups and 5 years of data on each group, the first 5 of the 25,000 rows of data are group 1, and so on. For the fixed and random effects models, the linear regression specification is different from all the other nonlinear specifications.

1. Fixed Effects Models

NOGIT's fixed effects estimators are, with the exception of the binary logit model, unconditional estimators. The dummy variable coefficients are all computed. The limit on numbers of groups is hundreds of thousands. The binary logit model may be fit by the conditional (Chamberlain) estimator or the unconditional (Greene) estimator. The linear fixed effects regression is requested with

REGRESS ; Lhs = … ; Rhs = … ; Panel ; FixedEffects $

The general form for nonlinear models is

Model ; Lhs = … ; Rhs = … ; Panel ; FEM $

Two models must be estimated immediately prior in cross section form, FRONTIER and NEGBIN. E.g.,

FRONTIER ; Lhs = … ; Rhs = … $
FRONTIER ; Lhs = … ; Rhs = … ; Panel ; FEM $

The negative binomial looks the same. The other three panel data forms, random effects, random parameters and latent class models for the stochastic frontier and negative binomial models are estimat3ed the same way. There are also a large number of other panel data specification for the stochastic frontier model.

There is a distinction for the logit model.

LOGIT ; … ; Panel ; FEM $ is for the unconditional estimator
LOGIT ; … ; Panel ; FIXED $ requests the conditional (Chamberlain) estimator.

```
|-> REGRESS  ; Lhs = income
    ; Rhs = one,age,educ,married
    ; Panel ; Fixed $
+--------------------------------------------------------------+
| Variable = ___________ Variable Groups    Max    Min   Average |
| TI         Group sizes  ID           550      7      1       3.7 |
+--------------------------------------------------------------+
+--------------------------------------------------+
| Frequency count for group sizes of TI             |
| Group size =   1   Pct =  20.36%   CumPct =  20.36%   |
| Group size =   2   Pct =  12.55%   CumPct =  32.91%   |
| Group size =   3   Pct =  12.36%   CumPct =  45.27%   |
| Group size =   4   Pct =  17.64%   CumPct =  62.91%   |
| Group size =   5   Pct =  14.55%   CumPct =  77.45%   |
| Group size =   6   Pct =  12.91%   CumPct =  90.36%   |
| Group size =   7   Pct =   9.64%   CumPct = 100.00%   |
+--------------------------------------------------+
```

```
-----------------------------------------------------------------------------
Ordinary    least squares regression ............
LHS=INCOME  Mean                   =       .34350
            Standard deviation     =       .16611
----------  No. of observations    =         2039  DegFreedom   Mean square
Regression  Sum of Squares         =      7.98305           3      2.66102
Residual    Sum of Squares         =      48.2492        2035       .02371
Total       Sum of Squares         =      56.2322        2038       .02759
----------  Standard error of e    =       .15398  Root MSE         .15383
Fit         R-squared              =       .14197  R-bar squared    .14070
Model test  F[  3,  2035]          =    112.23349  Prob F > F*      .00000
B-P test    Chi squared [  1]      =   1196.88596  Prob C2 > C2* =  .00000
[High values of  LM favor FEM/REM over base model]
Baltagi-Li form of LM Statistic    =    460.09192  [= BP if balanced panel]
Moulton/Randolph form:SLM N[0,1]   =     34.98263
|------------------------------------------------
Panel Data Analysis of INCOME              [ONE way]
               Unconditional ANOVA (No regressors)
Source         Variation  Deg. Free.   Mean Square
Between         42.94172        549.        .07822
Residual        13.29049       1489.        .00893
Total           56.23221       2038.        .02759
--------+--------------------------------------------------------------------
        |                 Standard            Prob.      95% Confidence
  INCOME| Coefficient       Error         z   |z|>Z*         Interval
--------+--------------------------------------------------------------------
     AGE|    -.00011        .00033      -.35  .7294     -.00076     .00053
    EDUC|     .02055***     .00148     13.91  .0000      .01766     .02345
 MARRIED|     .10630***     .00767     13.85  .0000      .09126     .12134
Constant|     .03630        .02429      1.49  .1351     -.01131     .08391
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

```
-----------------------------------------------------------------------------
LSDV        least squares with fixed effects ....
LHS=INCOME  Mean                   =       .34350
            Standard deviation     =       .16611
----------  No. of observations    =         2039  DegFreedom   Mean square
Regression  Sum of Squares         =      46.2254         552       .08374
Residual    Sum of Squares         =      10.0068        1486       .00673
Total       Sum of Squares         =      56.2322        2038       .02759
----------  Standard error of e    =       .08206  Root MSE         .07005
Fit         R-squared              =       .82205  R-bar squared    .75594
Model test  F[552,  1486]          =     12.43561  Prob F > F*      .00000
Estd. Autocorrelation of e(i,t)    =     -.100262
------------------------------------------------
Panel:Groups Empty       0,     Valid data       550
             Smallest    1,     Largest            7
             Average group size in panel        3.71
Variances    Effects a(i)        Residuals e(i,t)
                .048355                  .006734
Rho squared: Residual variation due to ai  .877760
Within groups variation in INCOME          13.2905
R squared based on within group variation  .247073
Between group variation in INCOME          42.9417
--------+--------------------------------------------------------------------
        |                 Standard            Prob.      95% Confidence
  INCOME| Coefficient       Error         z   |z|>Z*         Interval
--------+--------------------------------------------------------------------
     AGE|     .01326***     .00072     18.53  .0000      .01186     .01467
    EDUC|     .02248***     .00672      3.34  .0008      .00930     .03565
 MARRIED|     .08537***     .00984      8.68  .0000      .06609     .10465
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

Figure 50. Linear Fixed Effects Model

```
-----------------------------------------------------------------------------
FIXED EFFECTS Probit Model
Dependent variable               DOCTOR
Log likelihood function      -659.83028
Estimation based on N =    2039, K = 317
Inf.Cr.AIC  =    1953.7 AIC/N =     .958
Unbalanced panel has     550 individuals
Skipped  236 groups with inestimable ai
PROBIT (normal)  probability model
--------+--------------------------------------------------------------------
        |                  Standard            Prob.      95% Confidence
  DOCTOR|  Coefficient       Error       z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
        |Index function for probability
     AGE|     .07478***      .01533    4.88  .0000      .04474    .10483
    EDUC|    -.14432         .12618   -1.14  .2527    -.39163    .10298
 MARRIED|    -.18558         .20922    -.89  .3751    -.59563    .22448
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

Figure 51. Fixed Effects Probit Model

2. Random Effects Models

Several models, including REGRESS, PROBIT, LOGIT, ORDERED, POISSON and NEGBIN support familiar random effects forms. About 50 models provide a random parameters form, so all of those allow a random effects model in the form of a random constant term model. For the first set, the form of the command is the same for the linear and nonlinear models,

Model ; Lhs = … ; Rhs = … ; Panel ; Random Effects $

3. Random Parameters Models

A random parameters model is defined by defining the model, then defining which parameters are random. The model is estimated by maximum simulated likelihood. Some additional settings may be made to control the simulation.

Model ; Lhs = dependent variable
; Rhs = one,var1,var2,…,varK (list of variables, usually including one)
; RPM ; Panel
; Fcn = var(n), …,var(n) $

where 'var' is a name of a variable that appears in the Rhs list. The simulation can be based on random draws or preferably on Halton sequences which produce better results. An example of a model with six regressors, two random parameters, appears in Figure 35. In the example, the command is

PROBIT ; Lhs = doctor
; Rhs = one,age,educ,married,female,hhkids
; RPM ; Panel
; Fcn = one(n),female(n)
; Halton ; Draws = 50 $

To specify this as a simple random effects model, we would change the function definition to **;Fcn=one(n)**. In the specifications above, the '(n)' indicates a normally distributed parameter. There are 15 other distributions that can be used. An important feature of the RP models is the conditional estimates of the random parameters, E[$\boldsymbol{\beta}_i$|data$_i$]. This is requested with **;Parameters** and creates a matrix named BETA_I that can be further analyzed.

```
SETPANEL ; Group = ID ; Pds = Ti $
PROBIT   ; Lhs = doctor
         ; Rhs = one,age,educ,married,female,hhkids
         ; RPM ; Panel
         ; Fcn = one(n),female(n)
         ; Halton ; Draws = 50 $
```

```
-----------------------------------------------------------------------------
Random Coefficients  Probit   Model
Dependent variable               DOCTOR
Log likelihood function     -1176.65286
Restricted log likelihood   -1288.31976
Chi squared [   2 d.f.]       223.33379
Significance level               .00000
McFadden Pseudo R-squared      .0866764
Estimation based on N =   2039, K =   8
Inf.Cr.AIC  =   2369.3 AIC/N =    1.162
Unbalanced panel has    550 individuals
PROBIT (normal)  probability model
--------+--------------------------------------------------------------------
        |                 Standard            Prob.      95% Confidence
  DOCTOR|  Coefficient      Error       z     |z|>Z*         Interval
--------+--------------------------------------------------------------------
        |Nonrandom parameters
     AGE|      .01955***      .00353     5.53  .0000      .01262     .02647
    EDUC|     -.04438***      .01449    -3.06  .0022     -.07278    -.01597
 MARRIED|      .15720*        .08107     1.94  .0525     -.00168     .31609
  HHKIDS|     -.29947***      .07728    -3.88  .0001     -.45094    -.14800
        |Means for random parameters
Constant|     -.13051         .25112     -.52  .6033     -.62270     .36167
  FEMALE|      .53703***      .06866     7.82  .0000      .40246     .67159
        |Scale parameters for dists. of random parameters
Constant|      .92244***      .04186    22.04  .0000      .84039    1.00449
  FEMALE|      .24588***      .05089     4.83  .0000      .14614     .34563
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

Figure 52. Random Parameters Model Command and Results

4. Latent Class Models

A latent class model is specified with

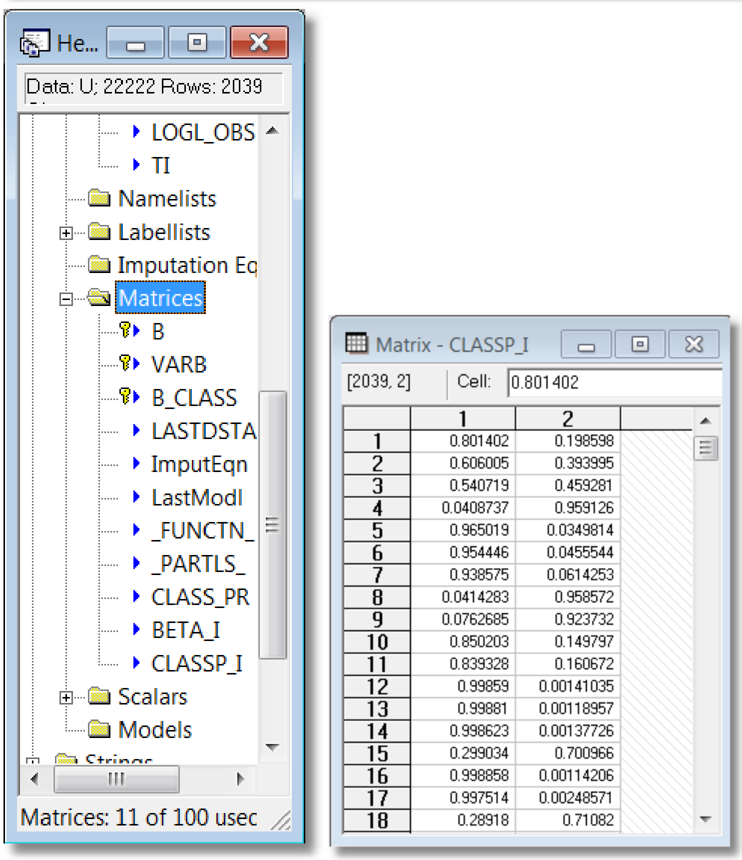
Model ; Lhs = dependent variable
; Rhs = one,var1,var2,…,varK (list of variables, usually including one)
; LCM ; Panel
; Pts = number of classes $

There are a variety of forms of LC models. It is possible to impose constraints across classes to create many different types of models.

Example: **LOGIT ; Lhs = Doctor**
; Rhs = one,age,educ,income,female
; Panel ; LCM ; Pts = 2
; Parameters $

In the example, we fit a two class latent class binary logit model. The ;Parameters requests computation of a matrix of conditional class probabilities. The estimated model, updated project window and 18 of the 550 rows of the class probabilities matrix are displayed in Figure 53.

```
-----------------------------------------------------------------------------
Latent Class / Panel Logit    Model
Dependent variable               DOCTOR
Log likelihood function     -1183.36226
Restricted log likelihood   -1298.45336
Chi squared [   7 d.f.]       230.18221
Significance level               .00000
McFadden Pseudo R-squared      .0886371
Estimation based on N =   2039, K =  11
Inf.Cr.AIC  =   2388.7 AIC/N =    1.172
Unbalanced panel has    550 individuals
LOGIT (Logistic) probability model
--------+--------------------------------------------------------------------
        |                Standard            Prob.      95% Confidence
  DOCTOR|  Coefficient     Error        z    |z|>Z*        Interval
--------+--------------------------------------------------------------------
        |Model parameters for latent class 1
Constant|      .05185       .75004      .07  .9449     -1.41820   1.52191
     AGE|      .04703***    .01037     4.54  .0000       .02671    .06735
    EDUC|     -.04805       .05075     -.95  .3437      -.14752    .05142
  INCOME|     -.44351       .61979     -.72  .4742     -1.65827    .77125
  FEMALE|     1.06581***    .27386     3.89  .0001       .52906   1.60256
        |Model parameters for latent class 2
Constant|    -1.02979       .69737    -1.48  .1398     -2.39660    .33703
     AGE|      .02167**     .00980     2.21  .0270       .00246    .04088
    EDUC|     -.11760***    .04307    -2.73  .0063      -.20202   -.03317
  INCOME|      .98235*      .56350     1.74  .0813      -.12209   2.08678
  FEMALE|      .77644***    .19955     3.89  .0001       .38534   1.16755
        |Estimated prior probabilities for class membership
Class1Pr|      .56120***    .04441    12.64  .0000       .47416    .64824
Class2Pr|      .43880***    .04441     9.88  .0000       .35176    .52584
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```



	1	2
1	0.801402	0.198598
2	0.606005	0.393995
3	0.540719	0.459281
4	0.0408737	0.959126
5	0.965019	0.0349814
6	0.954446	0.0455544
7	0.938575	0.0614253
8	0.0414283	0.958572
9	0.0762685	0.923732
10	0.850203	0.149797
11	0.839328	0.160672
12	0.99859	0.00141035
13	0.99881	0.00118957
14	0.998623	0.00137726
15	0.299034	0.700966
16	0.998858	0.00114206
17	0.997514	0.00248571
18	0.28918	0.71082

Figure 53. Latent Class Binary Logit Model

VIII. Multinomial Logit and Multinomial Choice

NLOGIT contains all of *LIMDEP* plus an additional set of model estimators and analysis tools for multinomial choice models such as the multinomial logit and multinomial probit specifications. The canonical form of the model is illustrated with this example that appears in our sample data set. A model for four models of travel, mode $\in$ (Air, Train, Bus, Car) defines the probability that an individual will choose one of the four. The underlying model is a random utility specification for individual i and modes 1,…,J:

$$U_{i,mode} = \alpha_{mode} + \beta_{time}TIME_{i,mode} + \beta_{cost}\,COST_{i.mode} + \gamma_{mode}INCOME_i + \varepsilon_{i,mode}$$
$$Y_{i,mode} = 1[U_{i,mode} > U_{j,mode} \forall j \neq i] \text{ (} Y_{i,mode} \text{ equals 1 for the mode with maximum utility, 0 else.)}$$
$$\varepsilon_{i,mode} \sim \text{Type I extreme value, independent across i and mode.}$$

The specification implies that

$$\text{Prob}(Y_{i,mode}=1) = \frac{\exp(\alpha_{mode}+\beta_{time}TIME_{i,mode}+\beta_{cost}COST_{i,mode}+\gamma_{mode}INCOME_i)}{\sum_{modes}\exp(\alpha_{mode}+\beta_{time}TIME_{i,mode}+\beta_{cost}COST_{i,mode}+\gamma_{mode}INCOME_i)}$$

This is the basic *multinomial logit model*. (Notice that the specification involves variables (TIME, COST) that vary across choices and a variable (INCOME) that does not vary across the choices. It is not necessary to distinguish. Mathematically, it is necessary to normalize the coefficients so that one of the α_{mode} parameters and one of the γ_{mode} parameters equals zero.) This is the basic model for multinomial choice. *NLOGIT* provides this model, a large number of extensions of the specification, such as the multinomial probit and nested logit models, and a set of analysis tools (similar to **SIMULATE** and **PARTIALS**).

A Tip: The **CLOGIT** command in *LIMDEP* is provided for the basic multinomial logit model. The extensions (as well as **CLOGIT**) are provided by *NLOGIT*.

A. Data

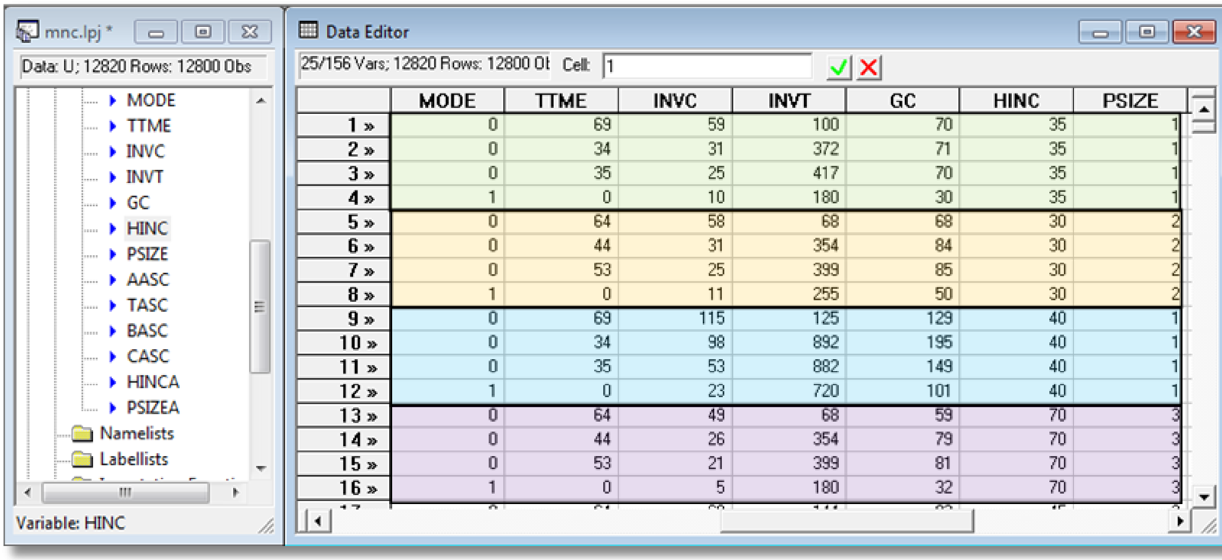
The data for this part of the description of *NLOGIT* are contained in the CSV file, mnc.csv ('mnc' for 'multinomial choice'). To replicate the examples and learn how to fit the models, you should **IMPORT** this file. There is also a project file provided, mnc.lpj, which you can **LOAD** directly. This data file contains 12,800 observations in two data sets. The forst data set contains 12 variables (columns), the second contains 7 – they are arranged side by side in 20 columns. The first 12 are 12,800 observations equal to 400 individuals times 8 repetitions (it is a panel) times 4 choices. The second data set contains 840 observations equal to 210 individuals times 4 choices in each observation. The 840 observations appear in the first 840 rows of their part of the data set. The rows below them (841-12800) contain missing values for these 8 variables. The shorter data set applies to the travel mode example described above.

An Important Tip: When you enter the data for multinomial choice analysis, the **IMPORT** step does not account for the internal structure of the data set. Our file, mnc.csv, is imported simply as 12,800 rows of data.   Like a panel data set, the internal structure of the data is accounted for when the data are used to fit a model.

Data for multinomial choice modeling resemble a panel data set. The data set is arranged in blocks of data for each person for each choice situation. Our examples both describe choices over 4 alternatives. The data are thus arranged with a line of data for each alternative in the choice set for the person. This is indicated in Figure 54.

A Tip: It is possible to work with choice data arranged on a single line – what some other programs call the 'wide form.' This is extremely cumbersome and greatly limits the range of specifications and model sizes. *NLOGIT* does provide a way to use these data, and to convert them to the more accommodating 'long form.'

A Second Tip: *NLOGIT* allows the number of choices in the choice set to vary across individuals. Our first data set is a choice experiment that has 8 choice situations for each person. *NLOGIT* also allows the number of choice situations in a stated choice data set to vary across individuals.



	MODE	TTME	INVC	INVT	GC	HINC	PSIZE
1 »	0	69	59	100	70	35	1
2 »	0	34	31	372	71	35	1
3 »	0	35	25	417	70	35	1
4 »	1	0	10	180	30	35	1
5 »	0	64	58	68	68	30	2
6 »	0	44	31	354	84	30	2
7 »	0	53	25	399	85	30	2
8 »	1	0	11	255	50	30	2
9 »	0	69	115	125	129	40	1
10 »	0	34	98	892	195	40	1
11 »	0	35	53	882	149	40	1
12 »	1	0	23	720	101	40	1
13 »	0	64	49	68	59	70	3
14 »	0	44	26	354	79	70	3
15 »	0	53	21	399	81	70	3
16 »	1	0	5	180	32	70	3

Figure 54. Multinomial Choice Data

The data in Figure 54 are 210 observations on four travel modes, AIR, TRAIN, BUS, CAR in the respective 4 rows. Notice that the first variable, MODE, is $Y_{i,mode}$ in our mathematical example. The first four individuals in the sample all chose AIR, as the 4th row equals one in each case. There are several variables that vary across the choices – they are the attributes: TTME = terminal time (waiting time to begin the journey), INVC = in-vehicle cost, INVT = in vehicle time, GC = a generalized cost measure. There are also two variables that do not vary across choices, HINC = household income and PSIZE = party size These are characteristics of the person (traveler). It is not necessary to expand choice invariant variables. This is done internally as part of the model specification.

B. Basic Multinomial Choice Model and Choice Substitution Elasticities

The essential command for a multinomial logit model is

CLOGIT ; Choices = list of names for the choices
; Lhs = the choice variable
; Rhs = attributes that vary across the choices
; Rh2 = characteristics that do not vary across choices $

Figure 55 illustrates. Note, if you include ONE in your Rhs list, it is automatically moved to the Rh2 list. Models can be specified with either or both Rhs or Rh2 variables. Neither is required. If you do not have an Rh2 list, but you include ONE on your Rhs, the program creates an Rh2 list for you and puts ONE in it. This is not done for any other variables.

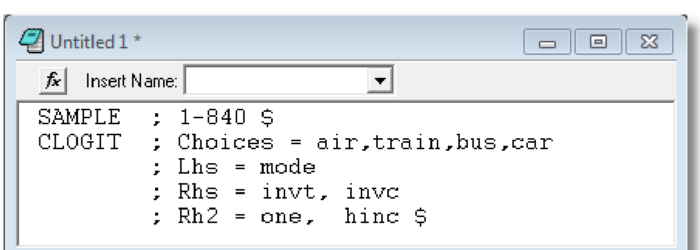


Figure 55. Command for Basic Multinomial Logit Model

Figure 56 shows the estimation results for the commands in Figure 55. This is the standard form of the display for the multinomial logit model. The next section lists some of the different choice models that can be specified. Estimation of every choice model begins with a starting values step at which the basic multinomial logit model is fit.

```
Discrete choice (multinomial logit) model
Dependent variable                Choice
Log likelihood function      -249.25650
Estimation based on N =    210, K =   8
Inf.Cr.AIC  =    514.5 AIC/N =    2.450
R2=1-LogL/LogL* Log-L fncn R-sqrd R2Adj
Constants only   -283.7588  .1216 .1103
Chi-squared[ 5]          =     69.00454
Prob [ chi squared > value ] =   .00000
Response data are given as ind. choices
Number of obs.=    210, skipped    0 obs
```

MODE	Coefficient	Standard Error	z	Prob. |z|>Z*	95% Confidence Interval	
INVT	-.00350***	.00075	-4.69	.0000	-.00496	-.00204
INVC	-.00858	.00626	-1.37	.1707	-.02084	.00369
A_AIR	-1.15318	.70809	-1.63	.1034	-2.54101	.23465
AIR_HIN1	.00243	.01045	.23	.8162	-.01806	.02292
A_TRAIN	2.07165***	.43004	4.82	.0000	1.22879	2.91451
TRA_HIN2	-.05090***	.01207	-4.22	.0000	-.07456	-.02723
A_BUS	.81928	.50127	1.63	.1022	-.16319	1.80176
BUS_HIN3	-.03268**	.01297	-2.52	.0117	-.05810	-.00727

Note: ***, **, * ==> Significance at 1%, 5%, 10% level.

Figure 56. Estimated Multinomial Logit Model

One of the major functions of the estimated choice model is to provide estimates of the impact of changes in relevant variables on the substitution patterns among the alternatives. Choice elasticities are the common device for this computation. The elasticity is defined as

E:Attribute(choice) = The effect on the probabilities of the choices when attribute in a particular choice Changes.

For example, E:cost(air) is the effect of changes in the cost of air on the probabilities of choosing the alternatives. Each attribute in the model produces a full matrix of elasticities. Elasticities are requested with the specification:

; Effects: attribute (alternatives desired)

It is common to request the effect of a change in an attribute in all choices. The following example shows how to do this. The '*' means 'all alternatives.'

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

This produces the table shown in Figure 57. This is the effect of changes in INVC on the probabilities of all all alternatives. If the specification had been invc(air,train), then only the first two rows of the table would be shown.

Elasticity wrt change of X in row choice on Prob[column choice]

INVC	AIR	TRAIN	BUS	CAR
AIR	-.5115	.2196	.2196	.2196
TRAIN	.1040	-.3363	.1040	.1040
BUS	.0392	.0392	-.2477	.0392
CAR	.0437	.0437	.0437	-.1363

Figure 57. Estimated Choice Elasticities

The table of elasticities can be expanded to include much more information by adding

; Full

to the command. This produces the results such as shown in Figure 58.

Average elasticity of prob(alt) wrt INVC in AIR

Choice	Coefficient	Standard Error	z	Prob. |z|>Z*	95% Confidence Interval	
AIR	-.51150***	.01126	-45.45	.0000	-.53356	-.48944
TRAIN	.21960***	.01160	18.93	.0000	.19686	.24234
BUS	.21960***	.01160	18.93	.0000	.19686	.24234
CAR	.21960***	.01160	18.93	.0000	.19686	.24234

Average elasticity of prob(alt) wrt INVC in TRAIN

Choice	Coefficient	Standard Error	z	Prob. |z|>Z*	95% Confidence Interval	
AIR	.10400***	.00440	23.63	.0000	.09537	.11263
TRAIN	-.33626***	.01583	-21.24	.0000	-.36728	-.30524
BUS	.10400***	.00440	23.63	.0000	.09537	.11263
CAR	.10400***	.00440	23.63	.0000	.09537	.11263

Average elasticity of prob(alt) wrt INVC in BUS

Choice	Coefficient	Standard Error	z	Prob. |z|>Z*	95% Confidence Interval	
AIR	.03918***	.00135	29.00	.0000	.03653	.04182
TRAIN	.03918***	.00135	29.00	.0000	.03653	.04182
BUS	-.24774***	.00681	-36.35	.0000	-.26110	-.23438
CAR	.03918***	.00135	29.00	.0000	.03653	.04182

Average elasticity of prob(alt) wrt INVC in CAR

Choice	Coefficient	Standard Error	z	Prob. |z|>Z*	95% Confidence Interval	
AIR	.04371***	.00202	21.63	.0000	.03975	.04767
TRAIN	.04371***	.00202	21.63	.0000	.03975	.04767
BUS	.04371***	.00202	21.63	.0000	.03975	.04767
CAR	-.13634***	.00752	-18.12	.0000	-.15109	-.12159

Figure 58. Full Display of Results for Elasticities

C. Multinomial Choice Models

Most of the extensions of the multinomial logit model are requested by modifying the basic command. The following will list a few of these by way of extending the example in Figures 55 and 56.

1. Multinomial Probit Model

The multinomial probit (MNP) model is an extension of the logit model. The MNP model allows some heteroscedasticity across choices as well as correlation of the utility functions. This is the usual first extension of the MNL model to relax the independence from irrelevant alternatives (IIA) assumptions. The model is requested simply by adding ;MNP to the basic specification. Since it is a simulation based estimator, sometimes it is a good idea to control the number of draws, as shown here.

```
NLOGIT    ; Choices = air,train,bus,car
          ; Lhs = mode
          ; Rhs = invt, invc
          ; Rh2 = one,  hinc
          ; MNP ; Draws = 5 ; Maxit = 5 $
```

(The command has used a very small number of draws and only 5 iterations. This estimator takes a very large amount of time. The results below show the results with 10 draws and allowing it to reach convergence.)

```
-----------------------------------------------------------------------------
Multinomial Probit Model
Dependent variable                    MODE
Log likelihood function      -222.31250
Restricted log likelihood    -291.12182
Chi squared [  13 d.f.]       137.61863
Significance level               .00000
McFadden Pseudo R-squared      .2363592
Estimation based on N =    210, K =  13
Inf.Cr.AIC  =    470.6 AIC/N =    2.241
Model estimated: Apr 21, 2013, 18:42:02
R2=1-LogL/LogL* Log-L fncn R-sqrd R2Adj
No coefficients  -291.1218  .2364 .2203
Constants only   -283.7588  .2165 .2000
At start values  -254.4449  .1263 .1079
Response data are given as ind. choices
Replications for simulated probs. = 100
Used pseudo random draws (Mersenne twister)
Number of obs.=    210, skipped    0 obs
```

```
--------+--------------------------------------------------------------------
        |                  Standard            Prob.      95% Confidence
    MODE|  Coefficient       Error       z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
        |Attributes in the Utility Functions (beta)
    INVT|    -.00861***      .00108    -7.94  .0000     -.01074    -.00649
    INVC|    -.03970***      .00981    -4.05  .0001     -.05893    -.02047
   A_AIR|   -3.27840**      1.32335    -2.48  .0132    -5.87212    -.68468
AIR_HIN1|     .02722         .02175     1.25  .2108     -.01541     .06985
 A_TRAIN|    2.32428***      .44153     5.26  .0000     1.45890    3.18967
TRA_HIN2|    -.02293*        .01201    -1.91  .0562     -.04647     .00060
   A_BUS|    1.55807***      .48400     3.22  .0013      .60945    2.50668
BUS_HIN3|    -.02510         .01627    -1.54  .1230     -.05700     .00680
        |Std. Devs. of the Normal Distribution.
  s[AIR]|    3.80349***     1.00600     3.78  .0002     1.83177    5.77521
s[TRAIN]|     .84104*        .45467     1.85  .0643     -.05010    1.73218
  s[BUS]|      1.0      .....(Fixed Parameter).....
  s[CAR]|      1.0      .....(Fixed Parameter).....
        |Correlations in the Normal Distribution
rAIR,TRA|     .33506         .73301      .46  .6476    -1.10161    1.77173
rAIR,BUS|     .55137         .51104     1.08  .2806     -.45025    1.55298
rTRA,BUS|     .78751***      .19899     3.96  .0001      .39751    1.17752
rAIR,CAR|      0.0      .....(Fixed Parameter).....
rTRA,CAR|      0.0      .....(Fixed Parameter).....
rBUS,CAR|      0.0      .....(Fixed Parameter).....
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
Fixed parameter ... is constrained to equal the value or
had a nonpositive st.error because of an earlier problem.
-----------------------------------------------------------------------------
```

Figure 59. Estimated Multinomial Probit Model

2. Nested Logit Model

NLOGIT allows up to 4 levels in a nested logit model. A nested logit model is specified simply by providing the tree structure in the NLOGIT command.

```
NLOGIT    ; Choices = air,train,bus,car
          ; Lhs = mode
          ; Rhs = invt, invc
          ; Rh2 = one,  hinc
          ; Tree= Private(air,car), public(train,bus)$
```

Tables of elasticities for a nested logit model include a decomposition of the total effect of switching between branches and substitution within a branch.

```
-----------------------------------------------------------------------------
FIML Nested Multinomial Logit Model
Dependent variable                MODE
Log likelihood function      -223.84995
Restricted log likelihood    -291.12182
Chi squared [  10 d.f.]       134.54373
Significance level               .00000
McFadden Pseudo R-squared      .2310781
Estimation based on N =    210, K =  10
Inf.Cr.AIC  =    467.7 AIC/N =    2.227
R2=1-LogL/LogL* Log-L fncn R-sqrd R2Adj
No coefficients  -291.1218  .2311  .2187
Constants only   -283.7588  .2111  .1984
At start values  -249.2565  .1019  .0874
Response data are given as ind. choices
The model has 2 levels.
Nested Logit form:IVparms=Taub|l,r,Sl|r
& Fr.No normalizations imposed a priori
Number of obs.=   210, skipped    0 obs
--------+--------------------------------------------------------------------
        |                Standard            Prob.       95% Confidence
    MODE|  Coefficient     Error       z    |z|>Z*          Interval
--------+--------------------------------------------------------------------
        |Attributes in the Utility Functions (beta)
    INVT|    -.00297***    .00055    -5.39  .0000     -.00404   -.00189
    INVC|    -.00061       .00135     -.45  .6503     -.00326    .00203
   A_AIR|   -1.39463***    .35120    -3.97  .0001    -2.08297   -.70629
AIR_HIN1|     .00172       .00452      .38  .7035     -.00714    .01058
 A_TRAIN|     .02367       .21548      .11  .9125     -.39865    .44600
TRA_HIN3|    -.00800       .00571    -1.40  .1607     -.01919    .00318
   A_BUS|    -.96295***    .31930    -3.02  .0026    -1.58877   -.33712
BUS_HIN4|     .00161       .00845      .19  .8486     -.01494    .01817
        |IV parameters, tau(b|l,r),sigma(l|r),phi(r)
 PRIVATE|    12.5647***   3.27076     3.84  .0001      6.1542   18.9753
  PUBLIC|    7.73382***   1.96806     3.93  .0001     3.87650  11.59114
--------+--------------------------------------------------------------------
```

Figure 58. Estimated Nested Logit Model

```
+----------------------------------------------------------------------------+
| Attribute is INVC       in choice AIR                                       |
|                          Decomposition of Effect if Nest     Total Effect|
|                          Trunk    Limb    Branch   Choice     Mean   St.Dev|
| Trunk=Trunk{1}                                                              |
| Limb=Lmb[1|1]                                                               |
|    Branch=PRIVATE                                                           |
| *      Choice=AIR          .000    .000   -.157    -.025      -.182   .011 |
|        Choice=CAR          .000    .000   -.157     .027      -.130   .011 |
|    Branch=PUBLIC                                                            |
|        Choice=TRAIN        .000    .000    .184     .000       .184   .011 |
|        Choice=BUS          .000    .000    .184     .000       .184   .011 |
+----------------------------------------------------------------------------+

Elasticity wrt change of X in row choice on Prob[column choice]
---------+-----------------------------------
INVC     |    AIR      CAR     TRAIN     BUS
---------+-----------------------------------
      AIR|  -.1821   -.1299    .1840    .1840
      CAR|  -.0267   -.0396    .0391    .0391
    TRAIN|   .0648    .0648   -.1035   -.0721
      BUS|   .0216    .0216   -.0252   -.0457
```

Figure 59. Estimated Elasticities for a Nested Logit Model

3. Mixed (Random Parameters, RP) Logit Model and Willingness to Pay (WTP)

The mixed (random parameters) logit model is the platform for the most recent, advanced formulations of the multinomial choice models in NLOGIT. The RP logit model is specified by providing the definition of the random parameters and, if desired, controls for the simulations.

Willingness to pay (WTP) is often measured in a choice model. The typical calculation is based on

$$WTP_{attribute} = \beta_{attribute} / \beta_{income}$$

Which measures the marginal utility of the attribute divided by the marginal utility of income. When income does not appear in the model, often the negative of a cost coefficient is used as a proxy for the marginal utility of income. When the model has fixed (nonrandom) coefficients, the WTP can be computed simply as the ratio of two coefficients (with a calculator). When parameters are random, WTP will vary across individuals if either of the components does. Figures 60 and 61 show estimation of a random parameters model and examination of the estimates of WTP.

```
RPLOGIT    ; Lhs=mode ; Choices=air,train,bus,car
           ; Rhs=invt,invc
           ; Rh2=one,hinc
           ; Pts=50 ; Halton
           ; Fcn=invt(n)  ? This specifies a single random parameter.
                          ? This can be expanded, e.g., invt(n), invc(n).
           ; Wtp=invt/invc ; Parameters $
KERNEL     ; Rhs=wtp_i
           ; Title=Estimated Distribution of WTP Across Sample $
```

```
-----------------------------------------------------------------------------
Random Parameters Logit Model
Dependent variable                  MODE
Log likelihood function       -196.69869
Restricted log likelihood     -291.12182
Chi squared [    9 d.f.]       188.84624
Significance level                .00000
McFadden Pseudo R-squared       .3243423
Estimation based on N =    210, K =   9
Inf.Cr.AIC  =    411.4 AIC/N =    1.959
Model estimated: Apr 21, 2013, 22:05:33
R2=1-LogL/LogL* Log-L fncn R-sqrd R2Adj
No coefficients  -291.1218  .3243 .3146
Constants only   -283.7588  .3068 .2968
At start values  -249.2565  .2109 .1994
Response data are given as ind. choices
Replications for simulated probs. =  50
Used Halton sequences in simulations.
BHHH estimator used for asymp. variance
Number of obs.=   210, skipped    0 obs
--------+--------------------------------------------------------------------
        |                  Standard            Prob.       95% Confidence
    MODE|  Coefficient       Error       z    |z|>Z*          Interval
--------+--------------------------------------------------------------------
        |Random parameters in utility functions
    INVT|    -.10580***      .00854   -12.39  .0000     -.12253    -.08907
        |Nonrandom parameters in utility functions
    INVC|    -.11409***      .02803    -4.07  .0000     -.16903    -.05916
   A_AIR|   -51.8120***     4.62159   -11.21  .0000    -60.8702   -42.7539
AIR_HIN1|     .23254***      .08504     2.73  .0062      .06587     .39922
 A_TRAIN|    9.21530***     1.96569     4.69  .0000     5.36262   13.06798
TRA_HIN2|    -.08571         .05884    -1.46  .1452     -.20104     .02961
   A_BUS|    5.49669***     1.91604     2.87  .0041     1.74132    9.25206
BUS_HIN3|    -.04898         .05110     -.96  .3378     -.14914     .05118
        |Distns. of RPs. Std.Devs or limits of triangular
  NsINVT|     .10048***      .00879    11.44  .0000      .08326     .11770
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

Figure 60 Estimated Random Parameters Model

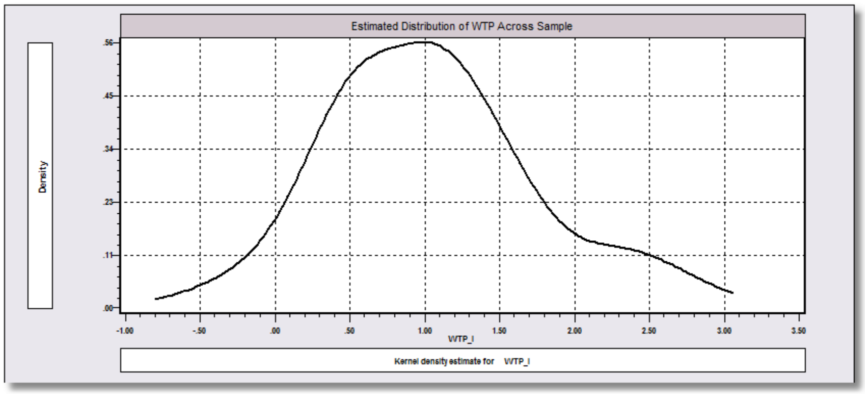


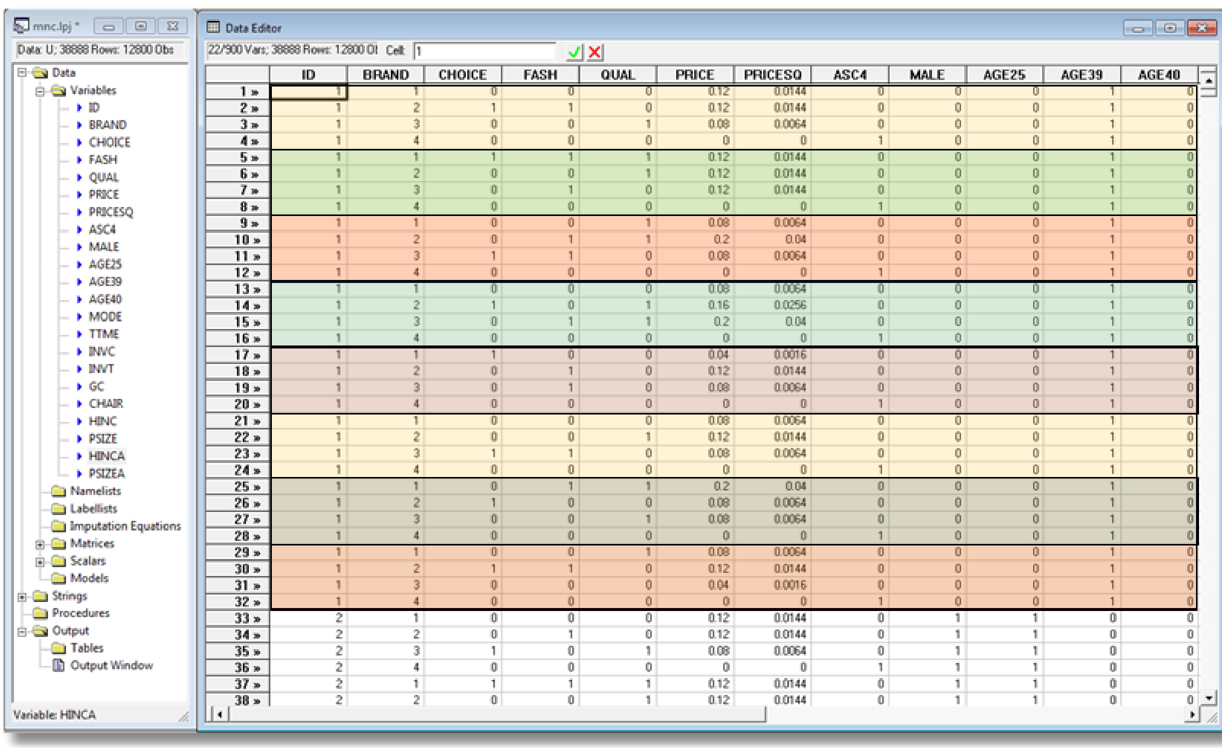
Figure 61 Sample Distribution of Expected Willingness to Pay

D. Stated Choice (Panel) Data

Stated choice experiments are analogous to panel data. The individuals in the sample are observed several times. Our experimental data in mnc.csv consist of 400 individuals each observed making one of four choices, eight times. There are 32 rows of data for each individual. The first individual is shown in Figure 62. For purposes of specifying multinomial choice models that use this structure of the data, this panel has Pds = 8, not 32.

A Tip: Do not use **SETPANEL** to set up stated choice data. The count variable must be constructed appropriately by you. If the number of repetitions is fixed, you will be able to use **;Pds=Nrep** in your command. You will not use **;Panel**. In our models using these data, we will use **;Pds=8**.

A Second Tip: These data are an ‘unlabeled’ choice set. The brands are distinguished only by their position in a list of brands. It is diffucult to interpret substitution patterns in a model for choice with unlabeled alternatives.



	ID	BRAND	CHOICE	FASH	QUAL	PRICE	PRICESQ	ASC4	MALE	AGE25	AGE39	AGE40
1 »	1	1	0	0	0	0.12	0.0144	0	0	0	1	0
2 »	1	2	1	1	0	0.12	0.0144	0	0	0	1	0
3 »	1	3	0	0	1	0.08	0.0064	0	0	0	1	0
4 »	1	4	0	0	0	0	0	1	0	0	1	0
5 »	1	1	1	1	1	0.12	0.0144	0	0	0	1	0
6 »	1	2	0	0	1	0.12	0.0144	0	0	0	1	0
7 »	1	3	0	1	0	0.12	0.0144	0	0	0	1	0
8 »	1	4	0	0	0	0	0	1	0	0	1	0
9 »	1	1	0	0	1	0.08	0.0064	0	0	0	1	0
10 »	1	2	0	1	1	0.2	0.04	0	0	0	1	0
11 »	1	3	1	1	0	0.08	0.0064	0	0	0	1	0
12 »	1	4	0	0	0	0	0	1	0	0	1	0
13 »	1	1	0	0	0	0.08	0.0064	0	0	0	1	0
14 »	1	2	1	0	1	0.16	0.0256	0	0	0	1	0
15 »	1	3	0	1	1	0.2	0.04	0	0	0	1	0
16 »	1	4	0	0	0	0	0	1	0	0	1	0
17 »	1	1	1	0	0	0.04	0.0016	0	0	0	1	0
18 »	1	2	0	1	0	0.12	0.0144	0	0	0	1	0
19 »	1	3	0	1	0	0.08	0.0064	0	0	0	1	0
20 »	1	4	0	0	0	0	0	1	0	0	1	0
21 »	1	1	0	0	0	0.08	0.0064	0	0	0	1	0
22 »	1	2	0	0	1	0.12	0.0144	0	0	0	1	0
23 »	1	3	1	1	0	0.08	0.0064	0	0	0	1	0
24 »	1	4	0	0	0	0	0	1	0	0	1	0
25 »	1	1	0	1	1	0.2	0.04	0	0	0	1	0
26 »	1	2	1	0	0	0.08	0.0064	0	0	0	1	0
27 »	1	3	0	0	1	0.08	0.0064	0	0	0	1	0
28 »	1	4	0	0	0	0	0	1	0	0	1	0
29 »	1	1	0	0	1	0.08	0.0064	0	0	0	1	0
30 »	1	2	1	1	0	0.12	0.0144	0	0	0	1	0
31 »	1	3	0	0	0	0.04	0.0016	0	0	0	1	0
32 »	1	4	0	0	0	0	0	1	0	0	1	0
33 »	2	1	0	0	0	0.12	0.0144	0	1	1	0	0
34 »	2	2	0	1	0	0.12	0.0144	0	1	1	0	0
35 »	2	3	1	0	1	0.08	0.0064	0	1	1	0	0
36 »	2	4	0	0	0	0	0	1	1	1	0	0
37 »	2	1	1	1	1	0.12	0.0144	0	1	1	0	0
38 »	2	2	0	0	1	0.12	0.0144	0	1	1	0	0

Figure 62. Stated Choice Experiment Data

Stated choice data allow the specification of essentially panel data models. The random utility models are variations on the general form

$$U(\text{choice})_{i,t,mode} = \beta_i{}'x_{i,t,mode} + \varepsilon_{i,t,mode} + w_{i,mode}$$
$$\beta_i = \beta + u_i$$

The definition of β_i implies that the parameters are random across individuals, but constant across choice situations. The random terms $w_{i,mode}$ are likewise constant across choice situations, and can be viewed as random effects. The constancy of the random terms in the model allows observations to be correlated within the group, which is the essential feature of panel data. (There are many variations on this model described in the manual.)

The stated choice data consist for each person of 8 repetitions on the choice of one of 3 brands or none of the above. The attributes are 'fashion,' 'quality,' 'price' and 'price^2'. There are also two characteristics, gender coded as male=1 and female=0 and age, coded as a category for three brackets, under 25, 25-39, 40+.

1. Random Parameters Model

Figures 63 – 65 show estimation of a random parameters model with one random coefficient.

```
Untitled 1 *
Insert Name:
SAMPLE ; All $
RPLOGIT ; LHS = Choice ; Choices = Brand1,Brand2,Brand3,None
        ; Rhs = asc4,fash,qual,price
        ; Rh2 = male
        ; Draws = 50 ; Halton ; Pds = 8
        ; Fcn = price(n) $
```

Figure 63 Command for a Mixed Logit Model

```
-----------------------------------------------------------------------------
Start values obtained using MNL model
Dependent variable                Choice
Log likelihood function      -4145.28394
Estimation based on N =    3200, K =   7
Inf.Cr.AIC  =   8304.6 AIC/N =    2.595
Model estimated: Apr 21, 2013, 18:53:22
R2=1-LogL/LogL* Log-L fncn R-sqrd R2Adj
Constants only  -4391.1804  .0560 .0552
Response data are given as ind. choices
Number of obs.=   3200, skipped   0 obs
--------+--------------------------------------------------------------------
        |                Standard            Prob.      95% Confidence
  CHOICE|  Coefficient     Error       z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
   PRICE|   -12.9259***    .91678   -14.10  .0000   -14.7228   -11.1291
    ASC4|    -.09580       .08950    -1.07  .2844    -.27122     .07962
    FASH|    1.52509***    .07336    20.79  .0000    1.38131    1.66888
    QUAL|    1.10398***    .06967    15.85  .0000     .96742    1.24054
BRA_MAL1|    -.35062***    .09727    -3.60  .0003    -.54127    -.15997
BRA_MAL2|     .04184       .09455      .44  .6581    -.14348     .22716
BRA_MAL3|    -.12863       .08901    -1.45  .1484    -.30308     .04582
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

Figure 64. Multinomial Logit Starting Values

```
-----------------------------------------------------------------------------
Random Parameters Logit Model
Dependent variable                CHOICE
Log likelihood function      -4145.27934
Restricted log likelihood    -4436.14196
Chi squared [   8 d.f.]        581.72523
Significance level                .00000
McFadden Pseudo R-squared       .0655666
Estimation based on N =   3200, K =   8
Inf.Cr.AIC  =   8306.6 AIC/N =    2.596
R2=1-LogL/LogL* Log-L fncn R-sqrd R2Adj
No coefficients -4436.1420  .0656 .0648
Constants only  -4391.1804  .0560 .0552
At start values -4145.2839  .0000-.0008
Response data are given as ind. choices
Replications for simulated probs. =  50
Used Halton sequences in simulations.
RPL model with panel has     400 groups
Fixed number of obsrvs./group=        8
Number of obs.=  3200, skipped    0 obs
--------+--------------------------------------------------------------------
        |                  Standard            Prob.       95% Confidence
  CHOICE|  Coefficient       Error       z    |z|>Z*          Interval
--------+--------------------------------------------------------------------
        |Random parameters in utility functions
   PRICE|    -12.9268***     .91686   -14.10  .0000     -14.7238   -11.1298
        |Nonrandom parameters in utility functions
    ASC4|     -.09583        .08951    -1.07  .2843      -.27126     .07960
    FASH|     1.52509***     .07336    20.79  .0000      1.38131    1.66888
    QUAL|     1.10399***     .06967    15.85  .0000       .96743    1.24054
BRA_MAL1|     -.35060***     .09728    -3.60  .0003      -.54126    -.15994
BRA_MAL2|      .04186        .09456      .44  .6580      -.14347     .22719
BRA_MAL3|     -.12861        .08901    -1.44  .1485      -.30307     .04584
        |Distns. of RPs. Std.Devs or limits of triangular
 NsPRICE|      .08072        .84115      .10  .9236     -1.56792    1.72935
```

Figure 65. Estimated Mixed Logit Model

2. Error Components (Random Effects) Logit Model

NLOGIT's Error Components Logit (ECLOGIT) model is equivalent to a random effects model. It is also possible to specify the logical equivalent of a nested logit model. The example below specifies a nested effects model in which one branch contains the three brands and a second contains the outside alternative, none. Figure 66 shows estimates of an error components model. Note how the error components are specified. The (brand1,brand2,brand3) specifies that the same effect appears in all three utility functions.

```
ECLOGIT    ; LHS = Choice ; Choices = Brand1,Brand2,Brand3,None
           ; Rhs = asc4,fash,qual,price
           ; Rh2 = male
           ; Draws = 50 ; Halton ; Pds = 8
           ; Ecm=(brand1,brand2,brand3),(none)$
```

```
--------+-------------------------------------------------------------
        |                  Standard            Prob.      95% Confidence
  CHOICE|  Coefficient       Error       z    |z|>Z*         Interval
--------+-------------------------------------------------------------
        |Nonrandom parameters in utility functions
    ASC4|    -.09712        .09035    -1.07  .2824     -.27420     .07996
    FASH|    1.52529***     .07375    20.68  .0000     1.38074    1.66984
    QUAL|    1.10409***     .06996    15.78  .0000      .96697    1.24120
   PRICE|  -12.9286***      .92705   -13.95  .0000   -14.7456   -11.1116
BRA_MAL1|    -.35028***     .09775    -3.58  .0003     -.54187    -.15870
BRA_MAL2|     .04221        .09493      .44  .6566     -.14384     .22826
BRA_MAL3|    -.12838        .08960    -1.43  .1519     -.30399     .04724
        |Standard deviations of latent random effects
SigmaE01|     .05936**      .02384     2.49  .0128      .01265     .10608
SigmaE02|     .05095**      .02373     2.15  .0318      .00444     .09746
--------+-------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
----------------------------------------------------------------------

 Random Effects Logit Model
 Appearance of Latent Random Effects in Utilities
 Alternative       E01 E02
+--------------+---+---+
|  BRAND1      | * |   |
+--------------+---+---+
|  BRAND2      | * |   |
+--------------+---+---+
|  BRAND3      | * |   |
+--------------+---+---+
|  NONE        |   | * |
+--------------+---+---+
```

Figure 66. Estimated Error Components Logit Model

3. Latent Class Multinomial Logit Model

The data used in this set of examples are experimental, and are carefully generated by an underlying latent class model in which the class probabilities depend on age and sex, and the choices depend on fashion, quality and price, exactly as specified below. The results are shown in Figure 67.

```
LCLOGIT    ; Lhs = Choice ; Choices = Brand1,Brand2,Brand3,None
           ; Rhs = asc4,fash,qual,price,pricesq
           ; Pds = 8
           ; Lcm = male,age25,age39 ; Pts = 3$
```

```
-----------------------------------------------------------------------------
Latent Class Logit Model
Dependent variable               CHOICE
Log likelihood function     -3648.66560
Restricted log likelihood   -4436.14196
Chi squared [  23 d.f.]      1574.95271
Significance level               .00000
McFadden Pseudo R-squared      .1775138
Estimation based on N =   3200, K =  23
Inf.Cr.AIC  =   7343.3 AIC/N =    2.295
R2=1-LogL/LogL* Log-L fncn R-sqrd R2Adj
No coefficients -4436.1420  .1775  .1755
Constants only  -4391.1804  .1691  .1671
At start values -4145.2390  .1198  .1177
Response data are given as ind. choices
Number of latent classes =            3
Average Class Probabilities
     .505  .237  .258
LCM model with panel has      400 groups
Fixed number of obsrvs./group=        8
Number of obs.=   3200, skipped    0 obs
-----------------------------------------------------------------------------
--------+--------------------------------------------------------------------
        |                  Standard            Prob.       95% Confidence
  CHOICE|  Coefficient       Error       z    |z|>Z*          Interval
--------+--------------------------------------------------------------------
        |Utility parameters in latent class -->> 1
 ASC4|1|    1.44175***      .38483      3.75  .0002      .68750    2.19599
 FASH|1|    3.01443***      .14704     20.50  .0000     2.72624    3.30262
 QUAL|1|    -.07457         .12737      -.59  .5582     -.32421     .17507
PRICE|1|   -6.97451        6.48123     -1.08  .2819   -19.67748    5.72847
PRICES|1|  -10.2076       23.79811      -.43  .6680   -56.8510    36.4358
        |Utility parameters in latent class -->> 2
 ASC4|2|    -.74323*        .39686     -1.87  .0611    -1.52106     .03460
 FASH|2|    1.22082***      .16381      7.45  .0000      .89975    1.54188
 QUAL|2|    1.10766***      .16487      6.72  .0000      .78452    1.43080
PRICE|2|  -19.7732***      6.85471     -2.88  .0039   -33.2082    -6.3382
PRICES|2|   22.4120       25.13694       .89  .3726   -26.8555    71.6795
        |Utility parameters in latent class -->> 3
 ASC4|3|     .28994         .41617       .70  .4860     -.52573    1.10561
 FASH|3|    -.16351         .16641      -.98  .3258     -.48966     .16265
 QUAL|3|    2.70297***      .18014     15.00  .0000     2.34990    3.05604
PRICE|3|   -6.95426        7.42439      -.94  .3489   -21.50580    7.59729
PRICES|3|  -7.92518       27.65361      -.29  .7744   -62.12525   46.27489
        |This is THETA(01) in class probability model.
Constant|    -.92984**      .37555     -2.48  .0133    -1.66590    -.19379
 _MALE|1|     .66719*       .36300      1.84  .0661     -.04429    1.37866
 _AGE25|1|   2.13778***     .32185      6.64  .0000     1.50697    2.76859
 _AGE39|1|    .69660        .43521      1.60  .1095     -.15639    1.54960
        |This is THETA(02) in class probability model.
Constant|     .36443        .34484      1.06  .2906     -.31144    1.04029
 _MALE|2|  -2.78223***      .69819     -3.98  .0001    -4.15065   -1.41380
 _AGE25|2|   -.14880        .54764      -.27  .7858    -1.22215     .92455
 _AGE39|2|   1.96741***     .71611      2.75  .0060      .56385    3.37096
        |This is THETA(03) in class probability model.
Constant|       0.0     .....(Fixed Parameter).....
 _MALE|3|       0.0     .....(Fixed Parameter).....
 _AGE25|3|      0.0     .....(Fixed Parameter).....
 _AGE39|3|      0.0     .....(Fixed Parameter).....
--------+--------------------------------------------------------------------
```

Figure 67. Estimated Latent Class Model

IX. Tools

NLOGIT provides a variety of tools that can be used with the model estimation commands or to create new estimators or statistics.

A. Scientific Calculator – The CALC Command

NLOGIT's scientific calculator is an important tool. In the following application we use it to compute the F ratio for a Chow test, then look up the 'p value' for the test by computing a probability from the F distribtion. Note that the named scalars computed with the CALC commands are added to the project, in the scalars list.

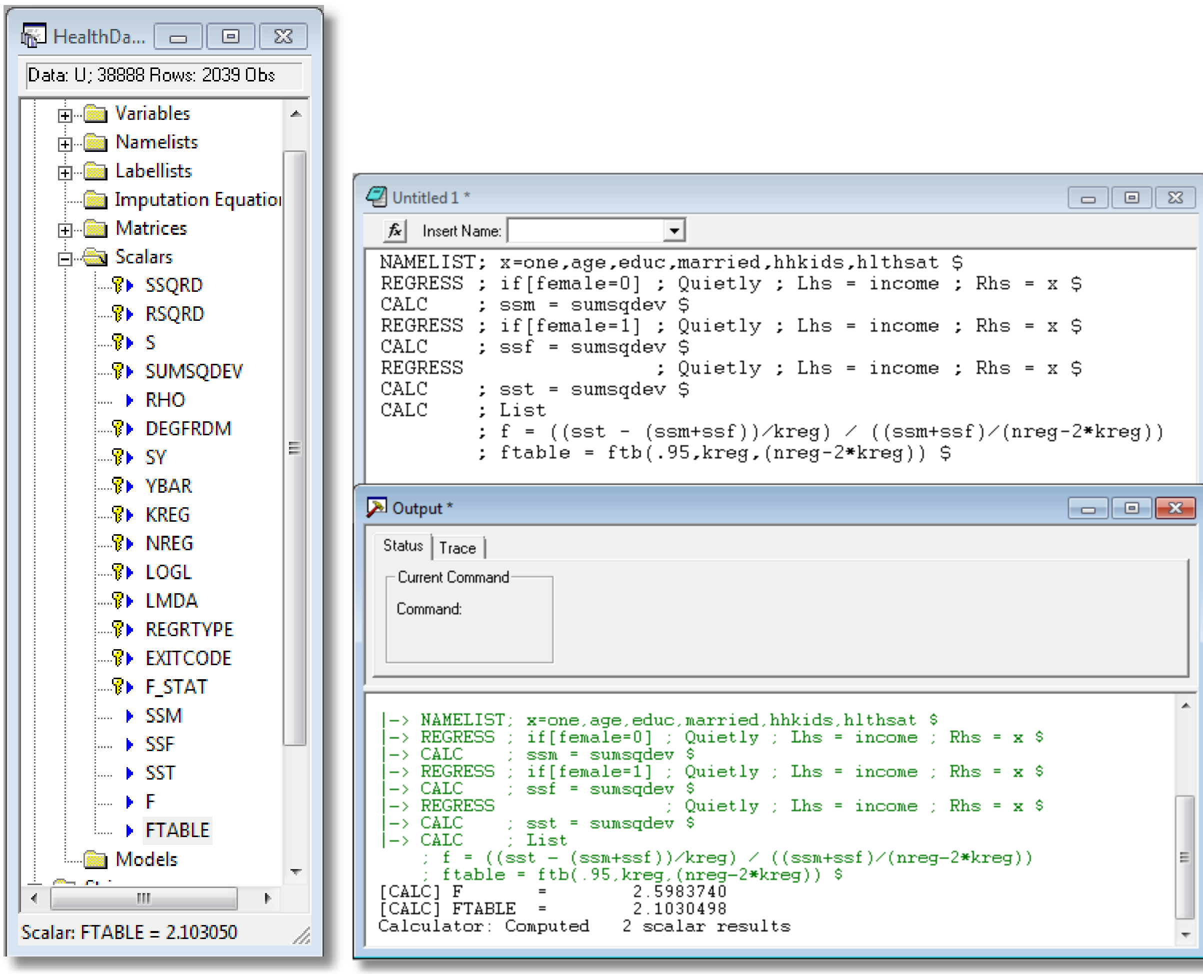


Figure 68. Chow Test Using Calculator

You can invoke the calculator with a **CALC** command that you put on your editing screen, such as **CALC;1+1$**, then highlight and submit with GO, as usual.

A Tip: **CALC** is a programming tool. As such, you will not always want to see the results of **CALC**. The **CALC** commands in the example above that pick up the sums of squares and the one that computes 1+1, do not display the result. If you want to see the result of **CALC**, add the word **;List** to the command, as in **CALC;List;1+1$** and in the commands above that compute the F statistic and the critical value.

The other way you can invoke the calculator is to use Tools→Scalar Calculator to open a calculator window. This would appear like the one below. When you use a calculator window, the results are always listed on the screen.

The example in Figure 69 computes two results, the sum of one and one and the rank of the covariance matrix for the coefficients in the most recently computed regression.

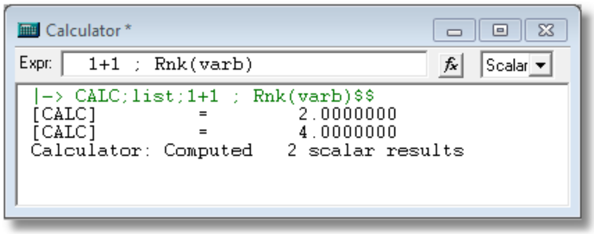


Figure 69. Calculator Window

In addition to the full range of algebra, **CALC** provides approximately 100 functions, such as the familiar ones, log, exp, abs, sqr, and so on, plus functions for looking up table values from the normal, *t*, *F*, and chi-squared distributions, functions for computing integrals (probabilities) from these distrubutions, some matrix functions such as rank and trace, and many other functions.

Any result that you calculate with **CALC** can be given a name, and used later in any other context that uses numbers. Note, for example, in the example in Figure 68, the scalars that are the sums of squares are used in the later command that computes the F statistic. All model commands, such as **REGRESS**, compute named results for the calculator. You can see the full list of these under the heading ‘Scalars’ in the project window shown in Figure 47. After you use **REGRESS** to compute a regression, these additional results are computed and saved for you to use later. Note, once again, the example in Figure 68. Each of the three **REGRESS** commands is followed by a **CALC** command that uses the quantity SUMSQDEV. In each case, this value will equal the sum of squared residuals from the previous regression. That is how we accumulate the three values that we need for the Chow test. Other statistics, YBAR, LOGL, and so on, are also replaced with the appropriate values when you use **REGRESS** or any other model command. The other model commands, such as **PROBIT**, also save some results, but in many cases, not all of them. For example, **PROBIT** does not save a sum of squared deviations, but it does save LOGL and KREG, which is the number of coefficients.

B. Matrix Algebra

The other major tool you will use is the matrix algebra calculator. *NLOGIT* provides a feature that will allow you to do the full range of matrix algebra computations. To see how this works, here is a fairly simple application: The LM statistic for testing the hypothesis that $\sigma_i^2 = f(\mathbf{z}_i'\boldsymbol{\gamma})$ against the null hypothesis that σ_i^2 is constant in a classical regression model is computed as LM = $\frac{1}{2}\mathbf{g}'\mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{g}$ where **g** is a vector of *n* observations on $[e_i^2/(\mathbf{e}'\mathbf{e}/n) - 1]$ with e_i the least squares residual in the regression of **y** on **X**, and **Z** is the set of variables in the variance function. A general set of instructions that could be used to compute this statistic are

```
NAMELIST   ; x = the list of variables ; z = the list of variables $
REGRESS    ; Quietly ; Lhs = y ; Rhs = x ; Res = e $
CREATE     ; g = (e^2/(sumsqdev/n)-1) $
MATRIX     ; list ; lm = .5 * g’z * <z’z> * z’g $
```

The **NAMELIST** command defines the matrices used. **REGRESS** (quietly) computes the residuals and calls them **e**. (There is a matrix command that will do this as well.) **CREATE** uses the regression results to compute the n observations on g_i. Finally, **MATRIX** does the actual calculation. The **MATRIX** command works the same as **CALC**, either in the editor screen or in its own Tools window.

There are only a few things you need to get started using *NLOGIT*’s matrix algebra program. The first is how to define a data matrix, such as **X** in the example above. The columns of a data matrix are variables, so, as you can see in the example, the **NAMELIST** command defines the columns of a data matrix. A single variable defines a data matrix with one column (i.e., a data vector) – note the use of the variable **g** in the example.

- The rows of a data matrix are the observations in the current sample, whatever that happens to be at the time. That means that all data matrices change when you change the sample. For example, **NAMELIST ; x=one,age,educ,income $** for our full healthcare data set defines a 2309×4 data matrix. When it is followed by **SAMPLE;1-500$** , x becomes a 500×4 matrix.

- Data matrices can share columns. For example, with the **x** just defined, we might also have a **NAMELIST;z= one,age,educ,income,married,hhkids $** Thus, x and z share four columns.

- In matrix algebra, the number **1** will represent a column of ones. Thus, if **x** is a variable, you could compute its mean with **MATRIX;List;Meanx=1/n*x'1$**. In defining a data matrix, as we did above, you may include 'one' to carry a column of ones.

There are many matrix operators. The major ones you need to know are

(1) +, -, * for the usual addition, subtraction, and multiplication.
The program will always check conformability. Note, row and column vectors are different.
(2) ' (apostrophe) for transposition
(3) <.> for inversion
(4) [variable] for a diagonal matrix in a quadratic form.

The last of these allows you to compute a result that involves a possibly huge diagonal matrix. For example, in a Poisson regression context, the asymptotic covariance matrix of the MLE is

$$\text{Asy.Var}[\mathbf{b}] = (\mathbf{X}'\mathbf{\Lambda X})^{-1}$$

Where $\mathbf{X}$ is the n×K data matrix and $\mathbf{\Lambda}$ is a diagonal matrix with $\lambda_i = \exp(\boldsymbol{\beta}'\mathbf{x}_i)$ on the diagonal. If you have, say, 1,000,000 observations (you might), then $\mathbf{\Lambda}$ is a 1,000,000×1,000,000 matrix that save for the tiny percentage of values that are on the diagonal, is a matrix of zeros. Obviously, you do not want to create $\mathbf{\Lambda}$ in your computer's memory. But, the syntax above allows you to do that. The matrix result is actually

$$\text{Asy.Var}[\mathbf{b}] = [\Sigma_i \lambda_i \mathbf{x}_i \mathbf{x}_i']^{-1}$$

which is never larger than K×K. *NLOGIT*'s matrix syntax reveals this to the program. The matrix command would be

MATRIX ; AsyVarb = < x' [lambdai] x > $

You could compute this with millions of observations.

When you compute a moment matrix, such as **X′X**, you need not both transpose and multiply. This would involve having a copy of X that is the transpose of X. Again, this is a superfluous waste of space. The command **X'X** means exactly what it looks like. The apostrophe is an operator that dictates how the result is to be computed.

In order to define a matrix with specific values in it, you use

MATRIX ; NAME = [row 1 / row 2 / ...] $

Within a row, values are separated by commas; rows are separated by slashes, and the whole thing is enclosed in square brackets. An example appears below. If the matrix is symmetric, you can define the matrix by its lower triangle – the first row has one element, the second has two elements, and so on.

In the same way that every model command creates some scalar results, every model command also creates at least two matrices, one named **B** which is the coefficient vector estimated, and one called **VARB** which is the estimated covariance matrix. You can use these in your matrix commands just like any other matrix. To compute the Poisson covariance matrix in the example immediately above, you could use

```
NAMELIST     ; x = the list of variables $
POISSON      ; Lhs = y ; Rhs = x ; Keep = lambdai $
MATRIX       ; List ; AsyVarb = <x'[lambdai]x> ; varb $
```

The display would reveal that the matrix we computed, AsyVarb, and the internally computed matrix, varb, are identical.

For another example, here is a way to compute the restricted least squares estimator,

$$\mathbf{b^*} = \mathbf{b} - (\mathbf{X'X})^{-1}\mathbf{R}'[\mathbf{R}(\mathbf{X'X})^{-1}\mathbf{R}']^{-1}(\mathbf{Rb} - \mathbf{q}).$$

For a specific example, suppose we regress y on a constant, x_1, x_2, and x_3, then compute the coefficient vector subject to the restrictions that $b_2 + b_3 = 1$ and $b_4 = 0$. We will also compute the Wald statistic for testing this restriction,

$$W = (\mathbf{Rb\text{-}q})'[\mathbf{R}\, s^2(\mathbf{X'X})^{-1}\mathbf{R}']^{-1}(\mathbf{Rb\text{-}q}).$$

Note that both examples use a shortcut for a quadratic form in an inverse.

```
NAMELIST     ; x = one, x1, x2, x3 $
REGRESS      ; Lhs = y ; Rhs = x $
MATRIX       ; r = [0,1,1,0 / 0,0,0,1] ; q = [1/0] $
MATRIX       ; m   = r*b - q ; d = r* <x'x> * r'
             ; br  = b - <x'x> * r'<d>m
             ; w  = m' <d> m $
```

In addition to the operators and standard features of matrix algebra, there are numerous functions that you might find useful. These include **ROOT**(symmetric matrix), **CXRT**(any matrix) for complex roots, **DTRM**(matrix) for determinant, **SQRT**(matrix) for square root and over 100 others.

C. Procedures

A procedure is a group of commands that you can collect and give a name to. To execute the commands in the procedure, you simply use an **EXECUTE** command. To define a procedure, just place the group of commands in your editor window between **PROCEDURE$** and **ENDPROCEDURE$** commands, then run the whole group of them. They will not be carried out at that point; they are just stored and left ready for you to use later. For example, the application above that computes a restricted regression and reports the results could be made into a procedure as follows:

```
PROCEDURE $
REGRESS      ; Lhs = y ; Rhs = X $
MATRIX       ; r = [0,1,1,0 / 0,0,0,1] ; q = [1/0] $
MATRIX       ; m   = r*b - q ; d = r* <x'x> * r'
             ; br  = b - <x'x> * r'<d>m
             ; w  = m' <d> m $
ENDPROCEDURE $
```

Now, to compute the estimator, we would define **X**, **y**, **r**, and **q**, then use the **EXECUTE** command;

```
NAMELIST     ; X = the set of variables $
CREATE       ; y = the dependent variable $
MATRIX       ; r = the matrix of constraints
             ; q = the vector on the RHS of the constraints $
EXECUTE      $
```

To use a different model, we'd just redefine **X, y**, **R**, and **q**, then execute again.

Since the commands for the procedure are just sitting on the screen waiting for us to Run them with a couple of mouse clicks, this really has not gained us very much. There are several better reasons for using procedures. The **EXECUTE** command can be made to request more than one run of the procedure, procedures can be written with 'adjustable parameter lists,' so that you can make them very general, and can change the procedure very easily. Repetitions of procedures can be used to develop bootstrap estimators of sample statistics.

The following computes a Chow test of structural change based on an **X** matrix, a **y** variable, and a dummy variable, **d**, which separates the sample into two subsets of interest. We'll write this as a 'subroutine' with adjustable parameters. Note that this routine does not actually report the results of the three least squares regressions. To add this to the routine, the CALC commands which obtain sums of squares could be replaced with **REGRESS ;Lhs = y ; Rhs = X $** then **CALC ; ee = sumsqdev $** In this application, we have used a feature of PROC that allows it to accept adjustable parameters.

```
/*        Procedure to carry out a Chow test of structural change.
          Inputs: X = namelist that contains full set of independent variables
                  y = dependent variable
                  d = dummy variable used to partition the sample
          Outputs F = sample F statistic for the Chow test
*/        F95 = 95th percentile from the appropriate F table.
PROC = ChowTest(X,y,d) $
CALC            ; k = Col(X) ; Nfull = N $
INCLUDE         ; New ; D = 1 $
CALC            ; ee1 = Ess(X,y) $
INCLUDE         ; New ; D = 0 $
CALC            ; ee0 = Ess(X,y) $
SAMPLE          ; All $
CALC            ; ee = Ess(X,y) $
CALC            ; List
                ; F = ((ee-(ee1+ee0))/K) / (ee/(Nfull-2*K) ) ; F95 = Ftb(.95,K, (Nfull-2*K)) $
ENDPROC $
```

Now, suppose we wished to carry out the test of whether the labor supply behaviors of men and women are the same. The commands might appear as follows:

```
NAMELIST ; HoursEqn = One,Age,Exper,Kids $
EXECUTE ; Proc = ChowTest(HoursEqn,Hours,Sex) $
```

A Tip: The preceding illustrates a particular calculation using a procedure. The Chow test (or its maximum likelihood equivalent for nonlinear models) can be carried out with a single command, such as

```
REGRESS ; For[(test) female = *,0,1] ; Lhs = y ; Rhs = x $
```

One of the main uses of procedures is to carry out repetitions of instructions. The following example illustrates. The next section extends this idea to bootstrapping estimators. The procedure in the example is applied in Figure 70.

```
/*  The data set consists of G groups.  We wish to estimate a logit  model of y on X
    for each group and arrange the coefficient vectors in the rows of a matrix named BG.
    There is a variable named GROUP that indexes the groups.  We do not know G.
    That is to be determined.
*/
NAMELIST     ; x = the group of variables $
CREATE       ; y = the variable $
CALC         ; g = max(group) ; k = col(x) $  Learn g and k from the model setup.
MATRIX       ; bg = init(g,k,0.0) $ Matrix where where we will stack the coefficients
PROCEDURE $
LOGIT        ; If[group = i] ; Quietly ; Lhs = y ; Rhs = x $
MATRIX       ; bg(i,*) = b' $  Puts i'th coefficient vector in i'th row of matrix.
ENDPROC $
EXECUTE      ; i = 1,g $  Executes for i = 1,2,…,g.
```

In the example below, 'group' is a random discrete uniform(1,10) variable, i.e., **CREATE; group = rnd(10) $**

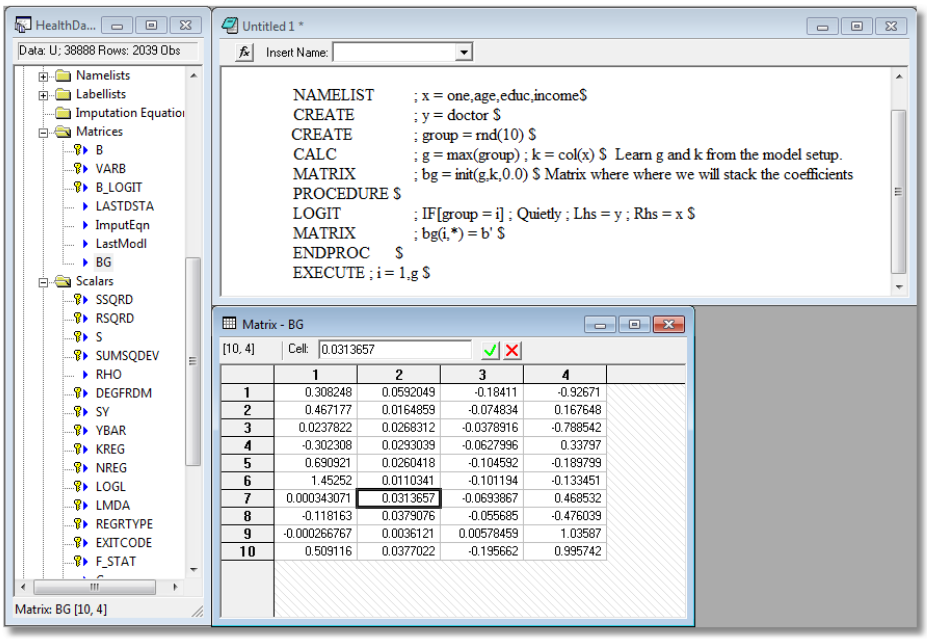


Figure 70. Repeated Execution of a Procedure

D. Bootstrapping

You can use procedures to compute bootstrap results for any scalar or vector that you compute using data. This can be a coefficient vector, a test statistic, or any other result that is computed using a sample of data. The general form of the procedure is as follows:

```
… any preliminary setup
PROCEDURE $
… compute the scalar with CALC or the vector with MATRIX.
… This part of the procedure may contain as many commands and
… calculations as needed.  It needs only to produce the result to be
… examine with a name, to be used later.
ENDPROC $
EXEC ; n = number of bootstrap replications ; Bootstrap = the name $
```

The procedure is actually executed n+1 times, first with the full original sample, then n times with the bootstrap samples. In the following example, we compute the vector of partial effects in a Poisson regression and bootstrap a covariance matrix. (Partial effects for a Poisson regression is a built in procedure in *NLOGIT* – we do this here just to illustrate the method.)

```
NAMELIST    ; x = age,educ,income,hlthsat $
PROCEDURE $
POISSON     ; quietly ; Lhs = docvis ; Rhs = x,one ; keep = lambdai $
CALC        ; apescale  = xbr(lambdai) $
MATRIX      ; ape = apescale * b(1:4) $
ENDPROC $
EXEC        ; n = 50 ; bootstrap = ape $
```

```
|->        NAMELIST   ; x = age,educ,income,hlthsat $
|->        PROCEDURE $
|->        POISSON    ; quietly ; Lhs = docvis ; Rhs = x,one ; keep = lambdai $
|->        CALC                  ; apescale  = xbr(lambdai) $
|->        MATRIX     ; ape = apescale * b(1:4) $
|->        ENDPROC $
|->        EXEC                  ; n = 50 ; bootstrap = ape $
Completed    50 bootstrap iterations.

-----------------------------------------------------------------------------
Results of bootstrap estimation of model.
Model has been reestimated      50 times.
Coefficients shown below are the original
model estimates based on the full sample.
Bootstrap samples have 2039 observations.
Estimated parameter vector is APE      .
Estimated variance matrix saved as VARB.
--------+--------------------------------------------------------------------
        |                 Standard            Prob.       95% Confidence
BootStrp|  Coefficient      Error       z    |z|>Z*          Interval
--------+--------------------------------------------------------------------
  APE001|     .04459**      .01732     2.58  .0100       .01065     .07853
  APE002|    -.28225***     .07141    -3.95  .0001     -.42220    -.14230
  APE003|    -.37537        .72097     -.52  .6026    -1.78844    1.03770
  APE004|   -2.28112***     .19633   -11.62  .0000    -2.66592   -1.89633
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

Figure 71. Results of Bootstrap Iterations

When you compute bootstrap replicates such as those shown in Figure 71, *NLOGIT* also creates a matrix named BOOTSTRP that contains the actual replicates. Figure 72 shows part of the results for the experiment in Figure 71.

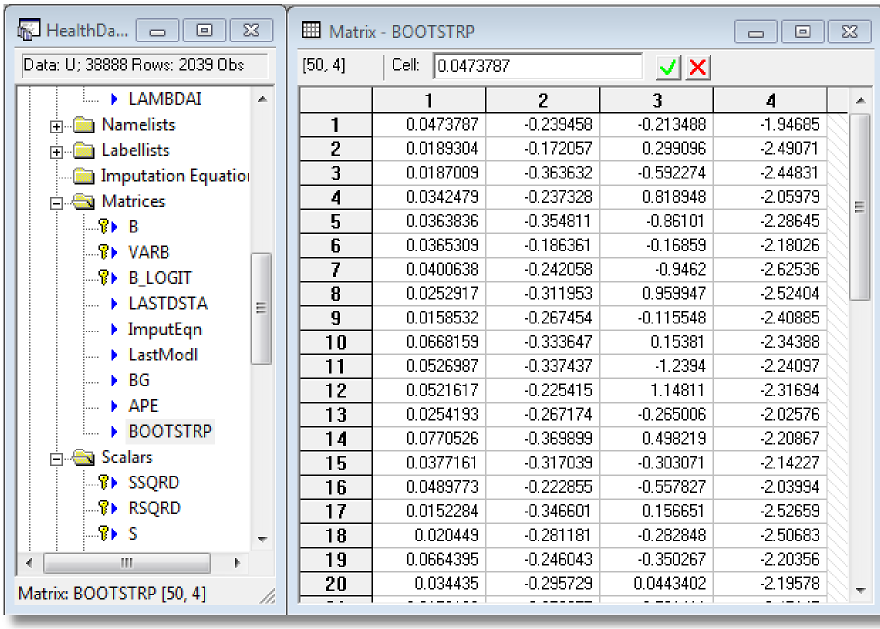


Figure 72. Saved Bootstrap Replicates

E. Displaying Results

NLOGIT provides several ways to display estimation results (and several formats, including export to *Excel* and formatted tables that can be exported to editors such as *Word*). To produce a standard output table for a set of estimates and the estimated covariance matrix, you need the estimates, the matrix, labels for the estimates (optional) and, perhaps, a title. Figure 73 shows how to construct a **DISPLAY** command for our bootstrap results in Figure 71. The command is

DISPLAY	**; parameters = ape**	? the name of the coefficient vector
	; covariance = varb	? the name of the covariance matrix
	; labels = x	? here, x provides a set of names, not the actual data
	; title = Bootstrap … $	? the desired title

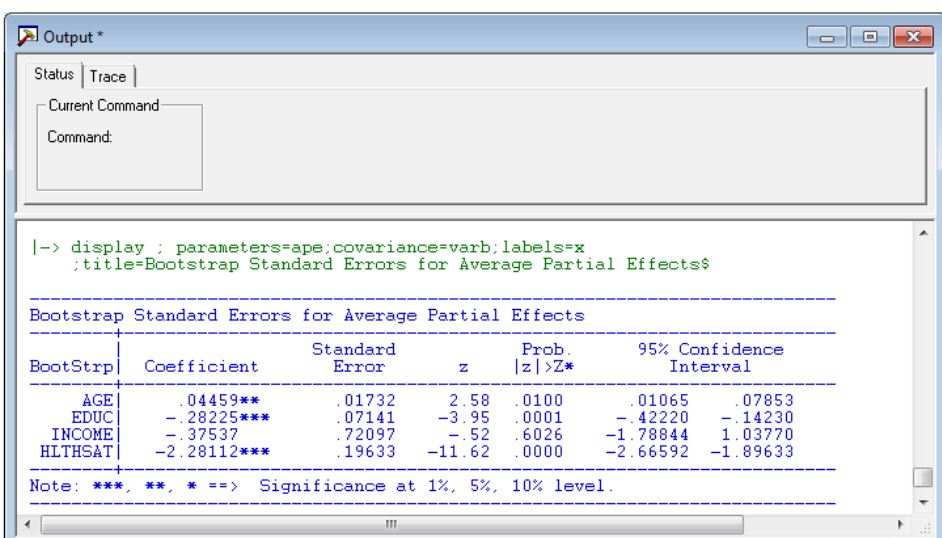


Figure 73. Display of Esstimation Results

E. WALD, SIMULATE and Standard Errors for Nonlinear Functions

Two devices, **WALD** and **SIMULATE** are provided for computing functions of parameters and standard errors for nonlinear functions. Both of them compute linear or nonlinear functions and standard errors usually using the delta method. (The method of Krinsky and Robb is also available.) Functions can be any desired computation using a parameter vector and the data.

1. The WALD Command

WALD is used for computing multiple functions and can be used to test hypotheses about functions of parameters. To illustrate, we manipulate the average partial effects shown in Figures 71 and 73. The WALD command to examine what is actually not a useful function would appear thusly:

```
WALD        ; parameters = ape
            ; covariance = varb
            ; labels = ca,ce,ci,ch
            ; fn1=ca*exp(ca'x) + phi(ca)  $
```

A Tip: In the function definition above, x is a namelist with 4 names that was defined above in part D, **x=age,educ,income,hlthsat**. The parameter vector is (ca,ce,ci,ch). The construction ca'x uses the parameters beginning with ca and x beginning with the first variable to compute the inner product. When one of the two components is shorter than the other, the shorter list is used. Thus, **ce'x** = ce*age+ci*educ+ch*income. If we defined **z=age,educ**, then **ca'z** would equal ca*age+ce*educ.

WALD requires the parameter vector, covariance matrix, labels, and up to 50 function definitions. As seen in the top panel of Figure 74, **WALD** computes the function at the means of the data using the current sample, and uses the delta method to compute standard errors and confidence intervals. By adding ;Average to the command, you can request that the avarage function value be computed, rather than the functions at the averages. This appears in the lower panel of Figure 74. **WALD** also computes the chi squared test of the null hypothesis that all of the functions are jointly zero. Note, in Figure 74, there is one function – the Wald statistic in this case is the square of the z statistic.

```
-----------------------------------------------------------------------------
WALD procedure. Estimates and standard errors
for nonlinear functions and joint test of
nonlinear restrictions.
Wald Statistic              =    5552.00703
Prob. from Chi-squared[ 1] =       .00000
Functions are computed at means of variables
--------+--------------------------------------------------------------------
        |                  Standard            Prob.       95% Confidence
WaldFcns|  Coefficient       Error        z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
 Fncn(1)|     .51783***      .00695     74.51  .0000       .50421     .53145
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
WALD procedure. Estimates and standard errors
for nonlinear functions and joint test of
nonlinear restrictions.
Wald Statistic              =    3626.57026
Prob. from Chi-squared[ 1] =       .00000
Functions of data are averaged over the obs.
--------+--------------------------------------------------------------------
        |                  Standard            Prob.       95% Confidence
WaldFcns|  Coefficient       Error        z    |z|>Z*         Interval
--------+--------------------------------------------------------------------
 Fncn(1)|     .51921***      .00862     60.22  .0000       .50231     .53611
--------+--------------------------------------------------------------------
Note: ***, **, * ==>  Significance at 1%, 5%, 10% level.
-----------------------------------------------------------------------------
```

Figure 74. WALD Command for Analyzing Nonlinear Functions

2. The SIMULATE Command

The **SIMULATE** command shown in Section VII.B.2 can also be used to analyze functions of parameter estimates. The base cases are give the same result as **WALD**, as shown in Figure 75 for this example – note the function analyzed is the same as used in WALD.

```
SIMULATE    ; parameters = ape ; covariance = varb
            ; labels = ca,ce,ci,ch
            ;function=ca*exp(ca'x) + phi(ca) $
```

```
-----------------------------------------------------------------------
Model Simulation Analysis for User Specified Function
-----------------------------------------------------------------------
Simulations are computed by average over sample observations
-----------------------------------------------------------------------
User Function       Function   Standard
(Delta method)       Value      Error      |t|  95% Confidence Interval
-----------------------------------------------------------------------
Avrg. Function       .51921     .00862   60.22       .50231      .53611
```

Figure 75. Analyzing a Function with SIMULATE

SIMULATE computes the average function as opposed to **WALD** which computes the function at the means. As noted, **WALD** will compute the average function if the command contains **;Average**. **SIMULATE** will compute the function at the means if the command contains **;Means**.

3. WALD or SIMULATE - Which Should You Use?

For computing a function and appropriate standard errors, **WALD** and **SIMULATE** give the same answers. They differ as follows:

- **WALD** can be used to compute the chi squared test statistic for testing the hypothesis that the functions are all zero (simultaneously)
- **WALD** can analyze up to 50 functions in the single command.
- **SIMULATE** has many options for analyzing scenarios and simulating a function over a variety of different settings of the variables in the equation.
- **SIMULATE** can plot function values as well as listing them.

An example of a more elaborate use of **SIMULATE** appears in Figure 76. The command is as follows:

```
SIMULATE    ; parameters = ape ; covariance = varb
            ; labels = ca,ce,ci,ch
            ;function=ca*exp(ca'x) + phi(ca)
            ;scenario: & educ=12(1)20
            ;plot $
```

Model Simulation Analysis for User Specified Function

Simulations are computed by average over sample observations

User Function (Delta method)	Function Value	Standard Error	|t|	95% Confidence Interval	
Avrg. Function	.51921	.00862	60.22	.50231	.53611
EDUC = 12.00	.51869	.00799	64.96	.50304	.53434
EDUC = 13.00	.51847	.00771	67.26	.50336	.53358
EDUC = 14.00	.51830	.00750	69.08	.50360	.53301
EDUC = 15.00	.51817	.00735	70.51	.50377	.53258
EDUC = 16.00	.51808	.00723	71.61	.50390	.53226
EDUC = 17.00	.51801	.00715	72.45	.50399	.53202
EDUC = 18.00	.51795	.00709	73.09	.50406	.53184
EDUC = 19.00	.51791	.00704	73.58	.50411	.53171
EDUC = 20.00	.51788	.00700	73.94	.50415	.53161

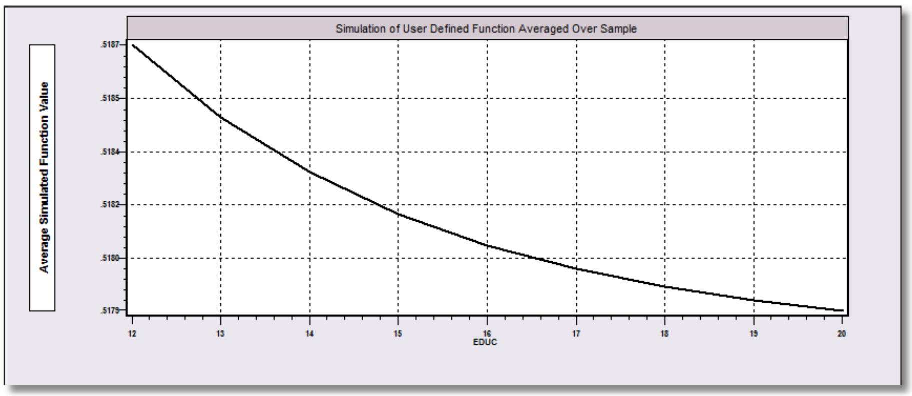


Figure 76. Analyzing a Scenario with SIMULATE