

Introduction to Stata

Enrique Moral-Benito*

January 2009

1 From the very beginning: What is Stata? Why should I use it?

Stata is a general purpose statistical software package. It is command-based software, available for Windows, Macintosh, and Unix systems. Stata provides a highly flexible interactive mode, which makes it easier for beginners to learn and use. Stata also supports features for programming and matrix manipulation.

Stata is advertised as having three major strengths: data manipulation, statistics, and graphics.

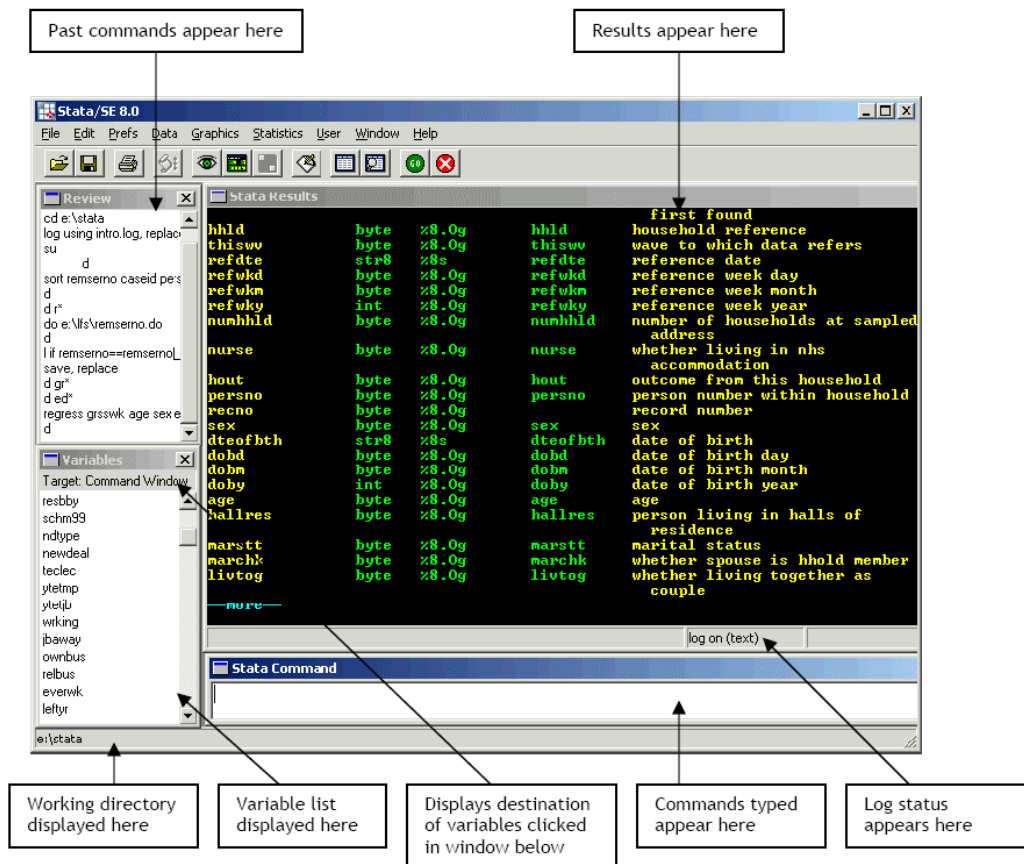
Stata is an excellent tool for data manipulation: moving data from external sources into the program, cleaning it up, generating new variables, generating summary data sets, merging data sets and checking for merge errors, collapsing cross-section time-series data on either of its dimensions, reshaping data sets from “long” to “wide”... In this context, Stata is an excellent program for answering ad hoc questions about any aspect of the data.

In terms of statistics, Stata provides all of the standard univariate, bivariate and multivariate statistical tools, from descriptive statistics, regression, principal components, and the like. It has a very powerful set of techniques for the analysis of limited dependent variables: logit, probit, ordered logit and probit, multinomial logit, and the like. Stata’s regression capabilities are full-featured, including regression diagnostics, prediction, robust estimation of standard errors, instrumental variables / two-stage least squares, generalized method of moments, seemingly unrelated regressions, vector autoregressions and error correction models, etc.

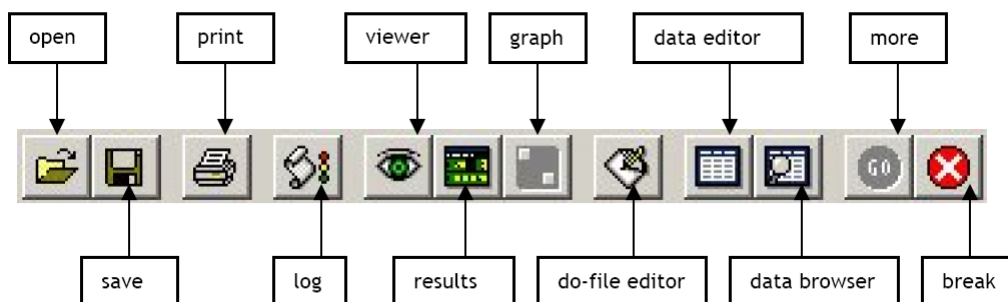
*E-mail: enrique.moral@gmail.com

2 Getting Started

Once you have clicked on Stata 9 icon you will see something like that:



On the other hand, the Stata toolbar is:



open: open a stata dataset. **save:** save a dataset. **print:** print contents of active window. **log:** to start or stop, pause or resume a log file. **viewer:** open viewer window, or bring to the front. **results:** open results window, or bring to the front. **graph:** open graph window, or bring to the front. **do-file editor:** open do-file editor, or bring window to the front. **data editor:** open data editor, or bring window to the front. **data browser:** open data browser, or bring window to the front. **more:** command to continue when paused in long output. **break:** stop the current task.

2.1 The Display

Automatically displayed windows

- Command Window: executes STATA commands; type in commands here and execute with the ENTER key.
- Results Window: displays commands entered and corresponding output/results; for screen display only; contents cannot be edited.
- Review Window: displays commands already entered; to re-run commands without typing them again, "click" the command line; the highlighted command line will appear in the STATA Command Window; ENTER to execute the command.
- Variable Window: Where a list of all the variables contained in the dataset currently loaded appears. This window is empty when you first start a Stata session because a dataset has yet to be loaded into memory. Allows for "point and click" additions to the command line.

Other Windows accessible through the "Windows" menu or by clicking on the respective menu button

- Data Editor: allows for manual entry of data or manual correction of data (use cautiously; however you will be asked to confirm changes upon closing the window)
- Data Browser: allows for viewing but no editing of data (button accessible only, safer than Data Editor)
- Do-File Editor: allows for the creation of saved lists of command "do-files"
- Viewer Window: allows for you to open log files which are text file versions of the results window suitable for editing (also provides access to help files)

2.2 Loading / Opening / Saving Stata-format (.dta) Files

Opening an existing Stata Dataset

Select Open under the File Menu and browse for the Stata files (.dta)

Memory note: By default, Stata starts with 4 megabytes of memory. Often, your dataset will be larger than this and you will need to increase the amount of memory Stata uses. Files larger

than 4,000,000 will not be loaded into Stata unless you increase the memory with the "set memory" command: `set mem 10m`

Entering data manually

Open the data editor as described above and either type in data or copy and paste data from an excel spreadsheet (caution: do not enter "." for empty cells; Stata will generate these markers by itself.)

2.3 Getting Help

To get help on a particular command, type: `help commandname` eg: `help regression`

To obtain all references to a topic, type: `search topic` eg: `search regression`

An easy way to get help, especially if you don't know the command name is to use the drop down menu. Go to help and then choose either stata command, search or contents.

For additional help, stata manuals are available in the statlab and online help is available at: <http://www.stata.com/>

2.4 Log Files / Do Files

It is ALWAYS a good idea to record your Stata session and to save your output for later viewing and/or printing. Logs capture all the text printed in the results window.

To open a log file, in the command window type: `log using c:\mylog`, or use the drop down menu going to File and opening Log. You now have a log file in your y drive called mylog that will record everything you type in the command window and the output that you see on the screen in the results window.

- To turn the log off, you simply type: `log close`
- To write over an existing log file you type `log using c:\mylog, replace`
- And to append to an existing file you type `log using c:\mylog, append`
- To temporarily stop logging: type in the Command window `log off`
- To resume logging: type in the Command window `log on`
- To save time editing, consider suspending (log off) the log when it is not necessary (log on to resume). Logs can be edited in wordpad etc. or in the viewer.

• Stata saves your log file as a .smcl so we need to translate it to text to take a look at it in word: `translate c:\mylog.smcl c:\mylog.txt`

- Alternatively, when you open a log file you can use the command `,text` which automatically

saves everything in a text file.

There is only one thing in life more important than a log file and that is a DO file.

A Do file is a record of operations that you are carrying out just as you would type them in one-by-one during a regular Stata session. Any command you use in Stata can be part of a do file. Do files are an easy way to clean and document your data, to replicate programs later on, to replicate a program with different data, and many other things.

`doedit` opens a text editor which allows you to edit do-files and other text files. You can again use the drop down menu: File -> Do, to open one. Alternatively, you can click on the icon that looks like a pencil on an envelope (see Stata toolbar above)

When using a do file it is a good idea to make notes to yourself: to do that type `*/` in the beginning of a line. Everything you type afterwards will be disregarded by Stata but it will remind you what you did in the program.

To run a do file you highlight the part of your program you want to run and simply click on the icon that looks like a note paper with a downwards arrow next to it (you can also highlight command lines and run them separately).

Note: Stata is case sensitive! All the commands in Stata can only be recognized in lower-case. File names, variable names and label strings can be in either lower or upper case, but you must be consistent.

2.5 Syntax and commands

There are a number of ways to request stata to run commands: typing them in the command window, running them through a do-file, or pointing and clicking in the drop down menus. Drop down menus are useful when you are unsure of commands to run, or options available. However, once you have a working knowledge of the commands, it is easier and faster to run them from the command line or through a do-file. The basic syntax of any Stata command:

```
COMMAND Variable-list Restrictions, options
```

2.5.1 Basic Commands

Stata handles two different types of variables: numeric variables (whose values are only real numbers), and string variables whose values are combinations of alphabetic and/or numeric variables.

Operators in Stata:

+ addition

- subtraction

* multiplication

/ division

^ raise to a power

> greater than

< less than

>= greater than or equal to

<= less than or equal to

== equal to (the relational operator for equality is a pair of equal signs)

~= not equal to

& and

| or

~ not

abs(x) absolute value

exp(x) exponential

ln(x) natural logarithm

log(x) natural logarithm

log10(x) logarithm to base 10

sqrt(x) square root

Note: Stata has a simple calculator function: display or di. For example, di sqrt(2)/2 or di normprob(1.64)

2.6 Working with data

All of the following can be done using commands or the drop down menu.

2.6.1 Descriptive Statistics

describe varlist

list varlist

sum varlist

table varlist

corr varlist

2.6.2 Manipulating Data

gen: generates variables.

Let's say we want to split the "level" variable into high level or low level, and we want to create a dummy variable:

```
gen level2=1 if level>5000
```

But what happens to those that have a level of < 5000? Stata assigns them .

We need to assign them a numeric value, and since we want to create a dummy, let's give them a 0:

```
replace level2=0 if level2==.
```

We can also do this using the drop-down menu. All the commands for data manipulation are found under Data -> Create or Change Variables. As we see, in the output window, after using the drop down menu (which took longer) Stata places the same command we used above in the output window and runs that command. So, if we did not know the command before, we can now copy it and keep it somewhere else (like in our do-file) for later use. Of course, it is also now stored in our log-file for later use.

rename: changes the name of a variable

```
rename initialname newname
```

egen: is used for creating variables with mathematical operators:

```
egen avx = mean(x)
```

other examples:

drop varlist: deletes the variables of varlist

sort varlist: arranges the observations of the current data into ascending order of the values of the variables of varlist.

The sort command is particularly useful when using the **by varlist:** prefix (discussed below) data must be sorted by varlist. For example:

```
sort country
by country: sum income
```

By: The prefix `by varlist:` causes the command that follows to be repeated for each unique set of values of the variables in `varlist`.

Remark: you can use the command `bysort` if you want to do both steps at once. For example, in a panel dataset, for creating a new variable that is the mean of income for each country you type:
`bysort country: egen minc=mean(income)`

2.7 Regression

All regression commands can be found in the drop-down menu by going to Statistics -> and then choosing the appropriate sub-menu.

`regression` or `reg:` runs a simple OLS regression

`regression depvariable, independent variables`

For example:

`regression y x1 x2 x3`

Now, restrict the observations to countries 1 – 98:

`regression y x1 x2 x3, if country<99`

With robust standard errors:

`regression y x1 x2 x3, robust`

To add variables to the command line, you may either type the variable or click on the variable name in the variable window.

2.8 Graphing

The drop down menus in Stata will be most useful for graphing. These are some of the more frequently used graphs, that can be found by typing the following commands, or using the drop down menu:

`hist x` for a histogram of the variable `x`

`scatter y x` for a scatter plot

Use the drop-down menu to insert titles and legends.

To save a graph, right-click on it and choose save as. Or, you can copy and paste your graph directly into word by using the right mouse button.