B203 Quantitative Economics and Econometrics

Compact Introduction to Intercooled Stata 7.0 on Unix

Cristina Santos
c.santos@ucl.ac.uk

7th October 2003

Abstract

These notes aim at giving you an overview of what you will need of Stata to follow this course. They are somehow summarising and updating previous notes that are already available in Prof. Meghir's course web page. Hence, a lot of the comments will be a repetition from those initial notes but do watch out for the differences! In the first section, I will guide you through how to log into Stata, the layout of the main screen, the format of the files that Stata recognises, and what you need to know before you actually get started. It will further describe you good practices you should always use to keep a safe record of your work. The second section will go through the language and functionals themselves.

1 Before you get started

1.1 Logging into Stata

To access this version of Stata from the cluster rooms, after you have logged in: press the *Start* button (bottom left corner of your screen) drag your mouse to *Programs*, then to *Unix Applications*, and click on *Stata on Unix*. Write down your username and UNIX password (note that this one might be different from the one you used to log into your WTS room) and you're in!

1.2 Main Stata Window - understanding its layout

The screen you will see will be composed of 4 windows. Read them now to recognise them.

The Review window - it will record all the commands you have written in the current Stata session, even the ones that gave an error message. In practise, after you press Enter at the end of an instruction, the instruction will be saved in this window and the results the instruction generates will appear in the Stata Results window.
Stata Results window - guess (!), it shows the results of the instructions you have typed in the nameless window below.
Nameless window - is the window where you write the commands you want Stata to execute. Make sure this window is selected and what you are writing showing up there. Otherwise, you will just have to write the last command again...
The Variables window - it will show you the name of all the variables of the data file you have opened at each moment. If it is empty, it means you don't have any file open.

1.3 Logging out

To log out, there is the good and the bad way. The BAD way: clicking on the cross on the top right corner. This will keep your Stata session running for as long as the server needs to understand that you are not there anymore. Given the limited number of Stata licenses, this basically means nobody can use the license you were using for a while. The GOOD way is to type *exit, clear* in the Nameless window followed by the enter key and you're gone. Alternatively, click on the *File* button (top left corner of the Stata window), then on *Exit* and it will do the same.

1.4 Inputing a data file

Stata will only accept as data files the ones with extension *dta*, so only files with a name such as "xxxx.dta" will be recognised. If you are familiar for example, with Excel, you will know it only accepts files with extension *xls* and Word will only accept files with extension *doc* and so on. Now there are 2 ways of opening a data set in Stata, the manual way and the right way.

The manual way: this is how we will be creating a file in exercise sheet 1. Once in Stata, we will edit the numbers directly. To do this, we can either use buttons or write down a command. You can click on *Window* (the before the last button on the top row of the screen), then on *Data Editor*. This will open a new window that looks like an Excel worksheet. The command you could have used to obtain exactly the same result would have been *edit*, followed by enter. Inputing data

is just writing down the numbers in the corresponding cells. Once you are done with it, leave the data editor window by clicking on the top right corner cross. Now, back in the Main Stata window, if you have typed any numbers in the Data Editor, you should see something on the Variables window. If you have inputed the numbers available in exercise sheet 1, you should see something like "var1" and "var2".

The right way: Disadvantages of the previous procedure are countless. First, it is ok if you just have 8 observations and 2 variables but it is not hard to find data sets with 1000000 observations or 400 variables and writing all entries down can be cumbersome... Second, it is not a good practice to use the Data Editor directly, it is nice to know it is there but changing its content is usually done through commands, not through a manual check, to avoid mistakes. Then, if you need to use the values again, you can't unless you save them. So the right way boils down to opening *dta* files already created outside Stata. In Prof. Meghir's course web page, there are 2 of these files that we will need throughout the year, you should think about saving them into your drive (I will tell you how later). So, to open a *dta* file onto Stata, click on *File* (top left corner of the screen), then *Open*. This will by default open your R: drive folder and you just have to click on the file you want to open, then press *Open*. If you look at the Stata Results window now, the corresponding instruction is *use filename.dta, clear*. The clear at the end is to tell Stata that it is ok to exit from the previously opened data file, in case there was one already open.

1.5 Saving a Stata file

Either because you have created a new file by editing it directly in the Data Editor window or because you modified an existing one, saving a Stata data file is done by using the command *save* or alternatively the buttons. Click on *File*, then on *Save as*. If you want to keep the same filename and in the same directory, just click on *Save*. Stata will ask you if you are sure you want to replace the existing file with the new one and you say *OK*. Done! Looking at the Stata Results window, you now know what you could have written to generate exactly the same result (just *save filename.dta, replace*). Please do this after typing the numbers for exercise sheet 1 because you will need them again for exercise sheet 2...

1.6 How to save a file into your R: drive

Let's use one of the files you will need for your homeworks as an example. Go to http://www.homepages.ucl.ac.uk/~uctpcac/b203.htm to access the course material you need. Click on *Term 1* to access the online course material needed for this term. To save any of the Stata files into your R: drive, put the mouse over the file of interest and right click on it (click with the right button). Choose *Save Target as*. Make sure the R: drive is selected and then choose the folder where you want to keep it. You can always create a folder called B203 and save it there (to create this folder, right click on the screen with the existing folders; select *New*, then *Folder*; now call it something nicer than "New Folder", call it "B203"). Double click on the folder of choice and click *Save*. Done! Repeat the same process for all other files of interest.

1.7 Record Keeping

This section explains a few good practice tips you should follow to keep a record of your work.

1.7.1 The log File

As you have realised by now, the commands you have typed show up in one window, the output in another but how can you actually save the results you are producing? This is what you can achieve with a log file. This file basically saves both the commands you have typed plus the corresponding output (including the error messages) that show up in the STATA RESULTS window. The reason why I have stressed this last point is that if a command results in a new window being open, the corresponding output will not appear in the log file (an example of this would be the *edit* command because it opens the Data Editor window or, as you will see shortly, the *graph* command that opens the Graph window).

How to open and close a log file? In the main Stata window, click on *File*, then drag your mouse to *Log* and click on *Begin*. Choose the folder where you want to save this file and give it a name. Again do remember that, by default, the first folders you will see will be the ones in your R: drive. Give it a name and, before pressing save, save as type Stata log. This file MUST have a *log* extension. In the Stata Results window, you will again see the corresponding instruction and the output. Do your things and to close the log file, you can either do the same as before (*File* - *Log* - now choose *Close*) or simply type *log close* followed by enter.

How to print the log file? Just remember where you have saved it, right click on it and select *Open with*. Then choose *Microsoft Word* to open the file. There, now just print it for a hard copy of it to hand me in.

1.7.2 The do-File

Even though you can get away in this course without ever needing to use these files, they are very useful if you want to run a lot of instructions. For those familiar with programming languages, there are usually 2 ways of running your instructions, the interactive and the batch mode. The interactive mode is when things happened as you type them in (pretty much what we have done so far). The batch mode is when you write down everything in a separate file and ask Stata to run it altogether. Advantages: easier to detect mistakes, easier to track down exactly what you have asked Stata to do; allows for more complex instructions that need more than a line to run (something like a loop for those that know what I am talking about, don't panick if you don't, we will not need them in this course.)

How to create and run a do-file? To create a do-file is simply to type the instructions in a Notepad file as long as you remember to save it with a *do* extension. How to do that? Click on *File*, then on *Save as*. Then change Save as Type from *txt* to *all files* and write the name of you file as "xxxx.do". Running the file is just going back to the Main Stata window, click on *File*, then *Do*, search for the file with the *do* extension you want to run and press *Open*. Now wait for error messages or for the final output.

Up to now, we managed everything with buttons but to actually make Stata do something, we need to write down the instructions in the way Stata understands them. My trick: all we need to know is the basic commands. For a full syntax description, we just type *help command* and read what it tells you. For a full description of the basic commands we will need in this course, wait for the next section.

1.8 Online Help

Fine, now what does Stata do? You might want to look at the online help Stata has available. There are also manuals but, as long as you are working in the cluster room or in a place with Internet, there isn't much a manual can give you that you can't find just in front of your computer. To familiarise yourself with

what Stata can do, you can run the tutorials it has available. By typing *help tutorial*, you are given a list of the tutorials that Stata offers. You can run them to get a feeling for the software but if you are getting bored... click on the red circle with a white cross on top of the Stata window, it will abort the tutorial or any long instruction you regret having asked Stata to do.

2 Basic Commands

To be able to use a software, you don't need to know all the commands or exactly how to write down an instruction. Here is a list of the most common commands you will be using this year. To know how to write down the whole instruction, just type *help command* and analyse the syntax structure, the options available and decide for yourself.

Instead of getting help on a particular command, you might need to wander around to know the commands associated with a particular topic. Then use *search whatever topic* or *lookup whatever topic*.

2.1 How to interpret Stata notation when it explains the syntax of a command

When you type in, e. g. *help regress* followed by enter, you see the syntax Stata allows. Bold white words are the words Stata wants users to write exactly as they appear or their shorthand words (you find these shortcuts by looking at the letters that Stata has underlined in each of these words - for example, one can either write regress or just reg to obtain the same result). Then everything that shows up between square brackets is optional, meaning you do not have to write it down but, if you do, then obey the syntax provided. What is in italic green means you need to write the corresponding object down, not the word or symbol itself. Let's keep analising the regress command. "depvar" is in italic green, meaning that instead of "depvar", you should write down the name of the dependent variable of your study (if by coincidence it is called "depvar", fine, but it rarely is...). "[varlist]" is also in green but in square brackets, meaning it is optional but if you do need to write the name of the variables in, do so instead of "varlist". Attention: because the square brackets are in green, it means you never write them down! These rules follow through easily to all commands.

2.2 Basic Commands to modify or create Variables

rename - if you think var1 is not a nice name and you might soon forget what it is actually in there, rename it to something pretty like Price!
drop - this command modifies both variables and observations, check the syntax. Just means delete. You say drop var1 and var1 disappears!
generate - this allows you to create a variable based or not on existing ones. Example: you want to create a variable which is the logarithm of an existing one. What is the Stata way of writing logarithm? You might at some point need to type *help language* and follow from there to know what are the available operators and functions Stata recognises.
label variable - if you are doing graphs, you might want to label variables so that the legend has the information you want.

2.3 Basic Commands to modify observations

replace - replaces the content of the observations that satisfy the conditions you describe.
sort - changes the order of the observations according to some criterion you specify.
drop
keep - opposite of *drop*

2.4 Basic Descriptive Statistics commands

There are tons and tons of them. Just a few:

summarize, *tabulate*, *list*, *describe*, *tabstat*, *count*, *correlate*, ... you can check these for yourselves.

2.5 Basic Regression Analysis commands

This list is short but that is because there are so many options and post-estimation commands related to the ones below that all you need is

regress - by typing *help regress*, you will see how many things you are allowed to do with this command (OLS, Instrumental Variables, Two Stage Least Squares, you will soon know what these are...). I will nevertheless draw your attention to a few of the most important related commands:

test - for hypothesis testing
predict - to create new variables based on the regression estimates.

2.6 Basic Graphing commands

There are mainly 2 commands for this: *plot* and *graph*.

plot - produces ugly with terrible interface graphs. Advantage? It shows up in the Stata Results window and can thus appear in a log file.
graph - pretty graphs, terrible to retrieve them back, lots and lots of options (type *help graph* to know more). The easiest way to save the graphs you produce is by copying them into a word file.

2.6.1 Saving Graphs

After you have typed in the graph command, a Stata Graph window will appear. In this window, right click on the light blue cross on the top left corner, drag mouse to *Edit*, then *Copy all*, then click on *To Clipboard*. Click on the word icon on the bottom left corner of your computer, which will open a new word file for you. There, click on *Edit*, then on *Paste* and there you are, a graph in a word file. Problems with this: background is black, let's change it. In the Stata Main window, before you make the graph, click on *Prefs* (meaning Preferences), choose *Graph Preferences*, then change the *color scheme* from black background to white background, then press OK. Done! Do it again with a pretty and less ink-wasteful graph.