MPP-C6: Statistics II

Programming with Stata

Max Callaghan

Hertie School of Governance

?? February 2015

Hertie School
of Governance

Outline

- Why Program?
- Reproducible Research
- Programming in Stata to accomplish difficult tasks and simplify repetitive tasks

Why Program?

- Computers can perform calculations faster and more accurately than humans
- When a task is difficult or repetitive, if often makes sense to instruct the computer to do it
- When those instructions are written down, we and others can see exactly what has been done
    - It's easier to repeat the work
    - It's easier to spot errors
    - It's easier to repeat the work, changing just one detail, without doing every subsequent step again.

Reproducible Research

- “The standard of reproducibility calls for the data and the computer code used to analyse the data to be made available to others” [1]
- Literate programming ties together data, code and the actual research output, enhancing reproducibility

Reproducible Research with Stata

- What do we already do that helps to keep our research reproducible?
- How do we ensure that what's in our research output reflects the calculations we report making?

Reproducible Research with Stata

- The ideal for perfect reproducibility would be to have a single document that contains instructions for performing calculations as well as for producing the research output we present.
- This is not possible in Stata without LaTeX but we can at least make the way we include Stata output in Word documents more systematic

Reproducible Research with Stata

[Include instructions to do this, with a simple example of output that changes to reflect a change in a do file]

Programming with Stata

Outline

- Stata basics
- Directory structure
- Reading data
- Transforming and processing data
- Presenting results with Stata

Stata Basics

- Pointing and clicking is fine for exploring data
- The command line is fine for trying out commands
- Anything you want to be able to reproduce, you should put in the do file

Stata Basics

Writing a good do file

A good do file should be readable by humans as well as computers.

- Use comments to explain what each line is doing
- Empty lines are free, space makes your code easier to read
- Use meaningful names when you create them, and write them consistently (variable_name, variableName or VariableName)
- Follow indentation conventions e.g.

```
for i in 1/5 {
  print i
}
```

Stata Basics

Understanding Stata Commands

Stata commands are preprogrammed functions that take information we give to them, do something with the information, then output something. We pass information to commands with *arguments* and *options*.
If you are not sure how to use a command you can get help by typing "help" and then the name of the command. For example,

```
help regress
```

will take you to the regress command's manual

Stata Basics

Reading the Stata manual

The Syntax regress *depvar [indepvars] [if] [in] [weight] [,options]* describes the basic use of the command

- Pay attention to the order of the arguments. This is how Stata knows which arguments are which
- Items in square brackets are optional
- Options come after a comma. Possible options are described in the help file
- You don't always need to set a lot of options, but you should pay atention to what the defaults are

If you don't know the command you want to use, you will have try and describe your problem to google.

Directory Structure

File paths tell the computer where to read and write information. They differ between Windows and Mac/Linux. (\or /)

- Absolute paths start from the top of the tree and specify each subdirectory until the file e.g. "C:bla\bla\data.dta" (or "/bla/bla/data.dta", or even "http://bla.com/data.dta")
- Relative paths start from the current working directory

Directory Structure

If you refer to more than one resource, it makes sense to set the working directory at the top of your do file and use relative paths. You'll want to think about how you structure your directory so that you can access items easily.

- If your do file produces output, think about where you want to save it so that you can access it automatically with another program
- This also allows you to change computers easily. Dropbox is an easy way to carry entire directories between computers. Git/Github is even better as it incorporates version control.

Reading Data

Data doesn't always come in nicely formatted dta files. Sometimes you have a data source or sources in files that aren't set up for stata to read. You often have to do a bit of work to get things into the format you want: the more of that work is recorded the better.

Some principles

- Keep an original copy of the data exactly as you found it, if you make changes, save to a new name
- Check the data has been imported properly before you use it
- Try and make changes with Stata in your do file. If you have to change in Excel, write down what you did

-

-

References