***Stata Introduction***

***Applied LDA, Fall 2018***

***Meizi Liu***

***Adapted from notes of previous TAs***



The information below will be written for Stata 14. Where I am aware of a difference in commands for earlier versions I will include additional notation.

**Do files**

Do files allow you to re-execute your commands and resume work where you left off when stopping work. It is similar to the R file or R markdown if you have used R before. The file extension for these files is “.do”.

*General notes on do files:*

The .do file allows you to stay organized—you don’t have to go back and annotate and organize the file afterward, when you may have forgotten exactly why you performed a certain command or if a command’s output ended up being what you needed. Within a .do file, you can start a line of text with an asterisk or double slash and that line of text will NOT be executed – this is how you make notes about your commands to yourself or others. Feel free to use this feature liberally such as if you want to explain how/why you used a unique command you prefer, etc. I typically will describe what I am doing with an asterisk and then type in a command or two, then highlight the commands and click on the “Execute” button (far right on the top of the do file editor with the blue play button and horizontal lines) or use the shortcut “Ctrl+D”. That way you can make sure your commands don’t have any errors and give you the output you want.

You can find numerous sources on this common topic online with a Google search.

The following examples are from a modified version of the Framingham data containing only data from period 3 (“session1 frmgham example period 3 only”). Note that the .do file accompanying this tutorial has many of these commands included, as well as others.

**Opening files and reviewing the data**

1. If you have previously been using a different dataset that still shows up in Stata, it is a good idea to clear the memory by typing the following in the command box and hitting enter (be sure to save any changes in the previous file if need be):

**clear**

1. I usually first open the data file being used, and then begin my do-file. This can be done by going to “File -> Open…” and finding the file, or by using the *use* command:

**use "mypath\mydatafile.dta", clear**

1. It is always a good idea to review the data first to see what they look like. This can be done using the “Data Editor (Browse)” button in the main Stata console, but there are also a number of useful commands for this review. These include *list*, which is simply a display of the data in the Results window, and *describe*, which will give a list of the variables and their types (string, float etc). You can simply type *describe* in the command window to list all variables (the list here is so long you will need to click the blue “more” at the bottom of the window (or hit the space bar) to see them all. If you find yourself having to continually click on “more” and want to turn this feature off, you can type, “set more off” in the command line either before or any time during your session.). More commonly, you can describe only the variables in which you are interested by typing their names.

**list**

**describe**

**describe sex age stroke**

1. To see a statistical summary of non-string variables (you cannot run statistical analyses on strings), you can enter the abbreviation *sum* by itself to see a summary of all variables. To avoid further massive output, this is perhaps ill-advised with this large dataset (if you accidentally do this you can type “q” or hit the red stop-sign at the top of Stata to stop the output); the following command instead only summarizes a couple of variables:

**sum age**

**sum age stroke**

You will see the number of observations and the means, standard deviations etc. for these variables. If you include “, detail” after the variable names, you will also see percentiles and some other information such as skewness and kurtosis.

1. You can also created frequency tables of variables with the tabulate or *tab* command:

**tab bpmeds**

**tab sex stroke**

You can list multiple variables after tab to cross-tabulate variables. This is useful for exploration of categorical variables and how they relate.

1. Be aware of missing data.

**tab bpmeds, missing**

**count if totchol==.**

**Generating simple variables**

Creating new variables is integral to data analysis. Where data are in a continuous numerical format, one way to do this is to perform simple mathematical functions in Stata to create a new variable. This new variable will represent the product, quotient, log etc. of the values of one or more existing variables. With continuous variables, this is easy because their raw value for a given observation is the actual value of interest, thus the variable names can be treated as numbers for the purposes of writing code.

Let’s say you want to represent cigpday (cigarettes per day) as packs per day instead. Let’s assume there are 20 cigarettes/packet (although this assumption is debatable)

1. We’ll use the *gen* or *generate* command for this in which we name this new, modified variable *cigpack*. To divide by 20, enter:

**gen cigpack=cigpday/20**

1. As always, you can examine the data produced with *tab* and *sum*:

**sum cigpack**

1. If you decide you prefer to use 25 cigarettes/pack instead of 20, you can use the command replace:

**replace cigpack=cigpday/25**

1. You can also perform simple mathematical computations directly using variable names (e.g., gen quotient= variable1/variable2 will create a new variable that represents the quotient of the values of these variables for the available observations – there is no need to manually enter any numbers for this type of calculation).

**gen chratio=totchol/hdlc**

**sum chratio**

1. Another useful command is egen, which is similar to gen but a bit more complex--used for generating variables that are summarized statistics.

**egen cursmoke\_count=sum(cursmoke)**

1. If at any point you find a variable is no longer needed, doesn’t work properly, etc., you can *drop* it by using the following command. (If you close Stata without saving when it asks, it will automatically drop all newly generated variables, preserving the original data. It will only create your new variables again when specifically commanded, such as when using a do file that does so):

**drop chratio**

If you permanently drop a variable, you will have a problem if any other new variables rely upon it for their calculated values! You will then get error codes indicating bad syntax when you try to run commands with these other, new variables.

1. Now close Stata. IMPORTANT: in general, if Stata gives you the further option of saving data when it closes, typically, you would NOT do so. This would save over the original data set you were using, which may permanently alter the original variables depending on what you did during the session.
2. To resume work, you may execute the .do file after reopening it in Stata. Open up Stata and then open up your .do file using the “File -> Open…” drop-down command. You will end up where you left off. NOTE: if there is an error in any command, Stata will stop at that command and will not execute any subsequent commands in your do file. You’ll see the dreaded red text indicating the problem. Check for code you meant to comment out but forgot to do so or vice versa.

Helpful resources:

1. UCLA Stata Learning Module <http://www.ats.ucla.edu/stat/stata/modules/>
2. In the Stata command: type “help *any keyword*”