Excel-Stata Block Teaching Notes

Content by Tommy Morgan
Formatting adapted from Natalie Jensen & Emily Leslie
Find more at Tommy's Lab Hours via tmorg.org

Last Updated June 2022

Note: don't try to have students follow along on their computers with any of this. They'll get behind or come in late and either derail the lecture or zone fully out. They will get to try stuff on the homeworks later, and you can always help out with stuff in lab hours. The exception to this is at the end of Intro to Stata 2, when you end early and can be wandering the lab and helping students as they begin to write their own stuff.

I have recorded examples of each of these three lectures. You can find them here:

Playlist

If that link doesn't work, navigating to labhours.tmorg.org will take you to my YouTube channel. From there, find the playlist called "My Teaching Material".

Nice to Sheet You: Intro to Excel

In this section, `[array]` refers to a set of cells, like A2:A16, while `[cell]` refers to just one cell, like A2. Logical expressions are written in double quotes, like `">4"`.

Data Setup and Summary Statistics

- Open up Excel! (Or Google Sheets)
- Gather age, birth order, and number of kids in the family for 15-30 students.
- Introduce Excel functions and how they work by using `=2022-[cell]` to fill in approximate year of birth. Adjust `2022` to whatever the current year is.
- Show how to use Excel functions by finding the min, max, mean, median, and standard deviation of age. Those functions can go in any nearby cell. They are `=MIN([age array])`, `=MAX([age array])`, `=AVERAGE([age array])`, `=MEDIAN([age array])`, and `=STDEV.S([age array])`, respectively.
- Demonstrate finding the correlation between age and year of birth to get that nice -1 result. Then consider finding correlation between birth order and number of kids to get something more interesting and still probably high, and correlation between age and birth order to get something fun and interesting. The function is `=CORREL([y variable array], [x variable array])`.
- Take the conditional mean of age if number of kids in the family is greater than 4. The function, which can go in any nearby cell, is `=AVERAGEIF([number of kids array], ">4", [age array])`. Compare this conditional mean of age to the original mean of age and also take the opportunity to teach what a logical expression is (i.e. a sentence that can be true or false) and how they're used here.

Graphs and Plots

Make a scatter plot showing number of kids in the family by year of birth.

- Highlight the year and family size columns. Take this time to demonstrate the CTRL-click and SHIFT-click shortcuts.
- After highlighting, go to the Insert tab and click the scatter plot icon.

- Demonstrate graph options by playing around with axis titles, colors, etc.
- Point out the many options for graph type available within Excel or Sheets.

Histogram

Make a histogram of any variable:

- Excel does not come pre-loaded with the ability to build histograms. Show how to install the Toolpak that enables this by navigating to File > Options > Add-ins > Analysis ToolPak + Go > Analysis Toolpak + OK.
- Now, in the Data tab, find a new Data Analysis button on the far right of the upper bar. Then click Data Analysis > Histogram > OK.
- Use the pop-up window to specify the input array, the output range, and the bin array. You'll have to go make a list of bins somewhere on the sheet for this. Check the chart output button at the bottom as well.

File Types

Make a quick mention of the differences between .xlsx and .csv, namely, .xlsx will keep your graphs alive and .csv is very common in data that you get from the internet.

A Blind Date with Big Data: Intro to Stata 1

Prior to the Lecture

- This lecture starts with the story of my WRTG 150 project where I analyzed US county-level data using only Excel. The folder of massive sheets I reference are linked here: Google Drive. If that link doesn't work, it's also pasted in plain text at the bottom of the document.
- I like to treat Stata as a helpful friend when I'm talking about it in this lecture. Little statements like "Hey Stata, can you please `use` the `example` file?" or "Stata never wants you to be angry at it, so it will never do anything you don't exactly tell it to do." I get the sense that humanizing an unfamiliar tool like Stata can help acclimate students to it. You do kinda have to commit to that bit, though, so if you're not into it, just go through the list and read around whatever similar stuff I write into these bullet points.
- Download the example file in the Google Drive folder to your desktop. If you don't or can't, just make a file beforehand called "example.dta" with a column of string data and a column of numbers. Make sure some of the string characters are non-numeric.

Excel is not the Answer

- Show the clean_final data file in the linked folder. Note that it shows signs of somebody hand-combining two enormous spreadsheets of data. Mention how awful that must have been!
- Show the two other files in the folder. Those are the two files that get combined to make the clean_final file. Mention how obnoxious it must have been to make those line up so well in Excel.
- Suggest that there must be a better way!

The Command Window

- Open Stata. Describe the general layout of the command window, taking care to specifically mention the command line at the bottom center, the command history to the left, and the variable list to the upper right (which should currently be empty).

- Specifically avoid mentioning the current directory line at the bottom left. Point out that there are shortcut buttons at the top of the window, but suggest that the students avoid them while learning how to use Stata.

- Run a `use` command to pretend to open the example.dta file, but do it without changing the directory. When the command fails, discuss the idea of a directory. I like to frame it as "Stata can only look in one drawer of the filing cabinet of your computer at a time, and it won't switch drawers without you telling it to because it never wants to make a mistake."

- Run a `cd` command to change the directory to wherever you have "example.dta" stored. Take this opportunity to begin building the idea that Stata commands run in a "command thing, options" framework.

- Click the red `use` command in the command history and show that it will now correctly open the example.dta file.

- Point out that there is now text in the variable list. Use a `browse` command to take a closer look!

The Browse Window

- Point out that the browse window has data organized just like Excel does, with variables in different columns and observations in different rows.

- Switch to edit mode (caution against this) to delete an observation in the numeric variable. Teach that the missing value "." represents an empty cell in numeric variables.

- Talk about the difference between the red string variable and the black numeric variable. I like to frame it as "Stata looks at the black stuff and says 'Oh, yeah, those are numbers. I know exactly what I can do with 24. Then it looks at the red stuff and says 'I have no idea exactly what this is 2 and 4 are, but they are a 2 and a 4 together.'"

- Pretend to try to turn the string variable into a numeric variable with a `destring` command. Don't specify the gen or replace options you'll need. Instead, it's time to go to the help files!

The Help Files

- Run `help destring` to pull up the help file for the destring command.
- Run through the syntax line. Point out that the stuff in square brackets [] are optional and the stuff in the curly brackets {} is required to make these commands run. Now you know that you'll need a gen() or replace option. Pick the replace option first and run it!
- Now return to the help file. Show how in-depth the help files are by clicking the blue `newvarlist` part of the `generate()` option in the syntax line. Play around with a couple of `tostring`s and `destring`s to show how commands and options work in general.
- Now open the help file for `drop` and `keep`. Play around with the `if` and `in` parts of the command. You can use the new variables you made with `tostring` and `destring` to show these off. The goal here is to introduce `if` and `in` and retouch on logical statements in conjunction with `if`, all while reinforcing the "command thing, options" syntax of Stata.
- Once that's all done, run a `destring var, force` command. Point out that if someone ran a `force` option by mistake, they would have no way to undo it! ... unless ...

The Do Editor

- Bring up the do file editor. Demonstrate writing and running code from the file, including highlighting and running only a couple lines.
- Please strongly stress commenting their code via asterisks or double back slashes. PLEASE.
- You might need to open the `use` help file to show the `clear` option. That will help emphasize the "undo button" allure of the do file.

Loose Ends

- Question 6 on the assignment asks the students to finish

Here's the plain text link to the Google Drive folder:

https://drive.google.com/drive/folders/1AzeaMKfHP8mZ9lPzZBxq2UmaMs1wNCJS?usp=sharing

More Data, No Problems: Intro to Stata 2

This lecture is about advancing from the basic understanding of Stata's basic features gained in Intro to Stata 1 to using some of the more advanced but still simple tools like `generate` and `label`. It should be shorter than the first one so that you can use the rest of the time to help them start the corresponding homework assignment, as they may need more help actually writing the stuff for this one than they did with the hand-holding in the first assignment.

Tying Up Every Useful End

- You can only use the `use` command for .dta files. Show `import delimited` and `import excel`, with their corresponding options, syntaxes, and requirements.
- Help them know where to find the file path to a specific directory. You can show clicking the top bar of Windows File Explorer, or the SHIFT + right click to "Copy path" option in windows.
- Demonstrate the `generate` command. It's one of the more fidgety ones. Pay particular attention to the = and ==. Show that you can use a logical statement to generate binary variables.
- Use whatever binary variable you just made to demonstrate labels. Show how to label the variable as a whole with `label variable` and how to label the actual values 1 and 0 with `label define` and `label values`. Point out that labeled data turns blue in the browse window.
- Pretend to not know the `correl` and `summarize` commands. Google things like "correlation command stata" and show how to find stuff that will give them commands they need. They will have to do this on the assignment, as it will require commands like `recode` and `rename` that you won't have (and **should not** have) shown them.

Go for it!

- That pretty much covers everything that bears covering.
- It's finally their turn! Now have them start the assignment!