ECON 3350/7350: Applied Econometrics for Macroeconomics and Finance

Tutorial 1: Introduction to Stata

1. Getting Started with Stata[1]

The point of this discussion section is to get you started using the statistical software package Stata.

- Starting from the ASCII dataset `cps98.csv`, load the data into Stata. This file contains the data on average hourly earnings, education, gender and age of individuals for a sample of workers in year 1998.
  - A quick way to do this is to save the Excel file and then load it into Stata using the *import* command.
- Use the *sum* command to summarize the variables in the dataset. What is the average hourly earnings and their standard deviation in the sample? How many male workers took the survey? What is the average, minimum and maximum age of the respondents?
- Use the *tab age* command to look at the distribution of age in the sample. What is the mode of this distribution? What is the median (approximately)?
- Use the *hist ahe* command to plot the histogram of the average hourly earnings (i.e., ahe). What can you say about the shape of its distribution?
- Use the *sum ahe if female == 1* command to calculate the average earnings for the females. What is the average earnings for the males? Do you find the difference economically significant?
- Use the *sum if ahe* $>$ *40* command to see who are the top earners – their age, gender, education level. How about those with hourly earnings less than 3?
- Use the *gen ahe2 = ahe*ahe* command to generate a new variable $ahe2$ equal to hourly wages squared.
- Use the *scatter ahe2 ahe, title("Ahe2")* command to graph the relationship between ahe and ahe^2
  - Now try adding the , *xlabel(0(2)50) ylabel(0(1000)3000)* option to see how to change the axes in your graph
- Use the *pwcorr* command to calculate the sample correlation between average hourly earnings and age. Does this correlation change for top and bottom earners?

[1] Questions 1 and 2 are from the Stata tutorial for ECON7310 taught in 2018.

2. Statistics using Stata

At the Famous Fulton Fish Market in New York city, sales of whiting (a type of fish) vary from day to day. Over a period of several months, daily quantities sold (in pounds) were observed. These data are in the Stata data file `fultonfish.dta`.

- Load the data to Stata (use *use fultonfish, clear* command), generate a data description (use *des* command), and obtain summary statistics of variables *lprice*, *quan* and *lquan* (may use *codebook* command with corresponding variable names).
- Use the *sum* command to compute the sample mean and standard deviation of *quan* (the quantity sold).
- Test the null hypothesis that the mean of *quan* is equal to 7,200 pounds per day at the 5% level of significance. Be sure to state (1) the null and alternative hypothesis, (2) the decision rule, (3) the test statistic, (4) the decision, and (5) your statistical conclusion.
- Construct the 95% confidence interval for the test above.
- Plot *lprice* against *lquan* (use commands *scatter*, *lfit*, and *qfit*). Comment on the nature of the relationship between these two variables.
- Use the *export excel* command to save this data to any folder on any drive as an Excel spreadsheet.

3. Time Series Data

The file `gld.csv` contains the time series data (daily from November 18, 2004 to November 23, 2018) of gold prices downloaded from Yahoo!Finance, which is an important source of financial and macroeconomic data. This question aims to introduce some Stata commands essential for processing time series data.

- Load the data to Stata using the *import* command. In what follows, you will only use variables *date* and *adjclose* (i.e., adjusted closing prices), so drop all other variables from the dataset using either *keep* or *drop* command. Change the variable name of *adjclose* to *goldprice* using the *rename* command.
- The first step of analyzing time series data is often to extract the datetime information, whose original values are usually stored in a string variable. You can convert the strings into the integers that Stata can understand and store those values in a date variable. One option is to use the *date* function and the *format %td* (for daily data) command. Create a date variable *t* using this method.
- Once you have the date variable that Stata can understand, you will need to declare the data as time series in order to use Stata's various time series commands/functions/operators. This can be done by using the *tsset* command. With time series set up, you can use two time series commands: $tin(t_1, t_2)$ (means "times in", from t_1 to t_2) and $twithin(t_1, t_2)$ (means "times within", between t_1 and t_2, i.e., excludes t_1 and t_2). List gold prices from December 1, 2004 to December 10, 2004. List gold prices within December 1, 2004 - December 10, 2004.

- One problem of the method proposed above is about the gaps in the time series data, e.g., gold prices are not available in non-trading days (weekends, public holidays, etc.). This may complicate the analysis using lag/lead/difference operators (see below) for those missing dates. In this case, it is more convenient to create a continuous time trend using the *encode* command. Create a date variable *time* using this method and declare time series with variable *time*. List gold prices from December 1, 2004 to December 10, 2004. List gold prices within December 1, 2004 - December 10, 2004. Hint: The *tin* and *twithin* commands are no longer useful, you can consider using the td function, which builds a one-to-one mapping between values of a date variable (no earlier than January 2, 1960) and positive integers. So then, you can use logical operators such as $>$, $<$, $\geq$, and $\leq$ among others.

- In time series analysis, it is common to study the impact of previous values on current ones. For example, you are interested in estimating how the values of variable Y evaluated at time t, denoted as Y_t, are affected by Y_{t-p}, $p = 1, 2, ...$ To do this, you will need to generate variables with past values Y_{t-p}, called Y lagged p periods, using the lag ("L") operator. Generate $goldprice_{t-1}$ and $goldprice_{t-2}$.

- Similarly, you can also generate future values of Y_t, Y_{t+p}, $p = 1, 2, ...$, using the lead ("F") operator. Create new variables $goldprice_{t+1}$ and $goldprice_{t+2}$.

- In many applications, it is the increments rather than absolute values that are of researchers' interests (e.g., returns vs. prices). You can use the difference ("D") operator to calculate the difference between current and past values. For example, applying $D1$ to Y_t yields $\Delta Y_t = Y_t - Y_{t-1}$, and applying $D2$ to Y_t leads to $\Delta_2 Y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2})$. Generate new variables $\Delta goldprice_t$ and $\Delta_2 goldprice_t$.

- Draw the time series plot of *goldprice* against *time* using the *line* command.