An Introduction to Stata

Instructions for Lab 1

Statistics 111 - Probability and Statistical Inference

Lab Objective

To become familiar with the software package Stata.

Lab Procedures

Stata gives us an enormous advantage over people who learned about and performed statistical analyses back in the pre-computer days. It allows us to avoid the drudgery of long, arithmetical calculations in favor of understanding concepts and analyzing data. You may find Stata a little annoying at times (all computer software is), but I suspect that you will be thankful of its existence once we start analyzing data.

Start up Stata. To do this, log in to the terminal with your Duke ID. If you use the Windows interface, click on the lower-left icon to bring up a list of all programs. Among these is a folder labeled "Math and Statistics" (or something similar). Open that, and among the list of packages is Stata. Click on its icon to start the Stata session.

You will first notice the four main windows to Stata: Review, Variables, Command, and Results. In addition, like most Windows-based programs, there are menu choices, including File, Edit, Data, Graphics, Statistics, User, Window, and Help. However, Stata is first and foremost a programming language, and as such all of its functionality can be accomplished from the Command line. Any output will appear in the Results window (the big window). (Type `memory` in the Command line to see an example of this. This shows how current memory is allocated as well as what constraints there are.) The Review panel shows a history of the recent commands that you have entered, and the Variables window lists all the variables currently in memory.

Data analysis tip: Note that almost all Stata commands follow a subset of the following structure, where items in brackets are optional for most commands:

```
[by varlist:]  command [varlist] [=exp] [if exp] [in range] [weight] [, options]
```

In this lab we focus on creating and loading Stata data sets as well as basic exploratory commands.

General features

1. The Working Directory

   It is always important to know where the files that you are using are saved on the computer. This is so both you and Stata can access the correct files. Let's see how this would work for this problem.

   In the section Loading Stata Data Sets is the link to the dataset for this problem set. Download the data and save/put it in the folder you are going to work in (e.g., C:\Users\John\Documents\Stats111\lab1). Now you need to direct Stata to this folder, which will be the folder you will be working from. There are two ways to accomplish this.

   (a) Use the menu bar and navigate to File -> Change Working Directory. This will pull up a window that you can then use to find the folder where you saved the data.

   (b) Issue a command to change the working directory by typing `cd` followed by the filepath to your file, (e.g., `cd C:\Users\John\Documents\Stats111\lab1`).

   Typing `pwd` into the command line well tell you the current directory and can be used to verify that you are in the right place.

2. "Do-files" and comments

   All the commands you enter into the Command Line for the lab can (and should) be put into a "do-file" to allow replication and access at a later date. You should save a "do-file" in the folder where the data is. Let's go through another example.

   Download this sample "do-file" and save/put it in the same folder where you saved the data above. Now go back to Stata, make sure you are in the correct working directory, and run this file through each of the following three ways:

   (a) Type `do example.do` in the command line (after making sure the Present Working Directory (`pwd` is the same one that the "do-file" is in).

   (b) Navigate to File -> Do... and manually select the file.

   (c) Open up the "do-file" editor (Ctrl+8; or Windows -> Do-file Editor), open the "do-file" in the editor, and Execute the job (Tools -> Execute or shortcut key on the toolbar).

   Note the words on lines 7-8 of this "do-file". They are green and are preceded by an asterisk (*). These are comments and are very useful in do-files. Stata ignores these lines when executing a "do-file."

   Further, you can create a record of all the commands given and the Results window output by creating a log file. This is simply a text file that contains the output. At the beginning of this example "do-file", it closes any open logs (line 2), starts up a new log (line 3), closes the new log (line 13), and then finishes the "do-file" (line 14).

   Now you can create a copy of this "do-file" and build off it for the problem set.

3. Fonts

   Different fonts generally mean very specific things in the labs. This is to help you more easily distinguish Stata code from the rest of the text:

   - `Typewriter text` refers to Stata commands.
   - ***Italicized text***, either in *`typewriter`* or ***in regular font***, refers to variable names. Sometimes these are specific variables in a data set (***Deaths***) while at others they are simply a place holder for any variable or name that you may choose (*`varname`*).
   - Red text refers to Data Analysis Tips.
   - Blue text refers to links to datasets, examples or places in the document.

Creating new data sets

Sometimes you need to create your own data sets from scratch (e.g., when analyzing data you have collected). This tutorial familiarizes you with some of the ways of creating Stata data sets. One of these is to use Stata's Data Editor and the other is to import data from a text file using the `insheet` command.

Option 1 - Use Stata's Data Editor

Data analysis tip: This is one of the times when it is easier to use a Stata menu option rather than simply typing the commands into the Command line.

Click on the Data Editor (Edit) button or navigate to Data -> Data Editor -> Data Editor (Edit). This brings up a blank spreadsheet that will become the data set. To keep things simple, we'll use observations (rows) for this part of the lab.

Here are the data for the ten most fatal earthquakes on record:

Country	Deaths
Haiti	92,000
China	242,769
Iran	150,000
China	235,502
Indonesia	230,210
Syria	230,000
China	820,000
Iran	200,000
Turkey	240,000
Japan	142,800

Let's first input the numbers in the first column. Once data is entered, you will note that the column was automatically given a name, ***var1***. This is a meaningless name for a variable, so let's rename it. Double click on the box containing ***var1*** and change the variable name to ***Deaths***.

Data analysis tip: When you create data sets for your own research, give your variables descriptive labels. It is easier to interpret analyses when the output has descriptive labels than when the output has labels like ***var1***, ***var2***, ***var3***, etc. Descriptive labels also make your data set comprehensible to others who may want to use it. Finally, if you use the data set in future analyses, you won't have to spend lots of time trying to decipher uninformative variable names.

Now let's input the countries into the next column, once again renaming it from ***var2*** to ***Country***. Note that the values changed color from black to red. Stata distinguishes between variables containing only numbers (numeric) and variables that contain letters, symbols and other characters (string).

Now that the data has been entered, return to the main Stata window. Note that now we have 2 variables in memory and that the results window has a lot of different commands in it (i.e., `replace var1 = 1 in 8`; `rename var2 country`). Everything you did in the Data Editor could have been done by typing commands into the Command line. However, as mentioned, this is an instance when it is easier to use the menu rather than the command line.

Option 2 - Import Data from a File

Another way to get data into Stata is to read it in from a text file with the **`insheet`** command. This requires two steps in this case: making the dataset, then reading it in.

1. Make the dataset

   - Open up a text editor and copy and paste the above data into it, including the headers.
   - We want to make the file a comma-delimited or comma-separated file, which means that each column of data is separated by a comma. The numbers have a comma as the thousands indicator and we don't want that, so remove the commas from each number. (See tip below).
   - Now replace the space between each country and number with a comma.
   - The first two lines should be:

     ```
     Country,Deaths
     Haiti,92000.
     ```
     [1]
   - Make sure the headers are also separated by a comma, and save the file as "countries.txt" in the same directory that all of your work will be for this lab (e.g., C:\Users\John\Documents\Stats111\lab1). Remember you can always check what the Present Working Directory is by typing **`pwd`**

[1] With comma-delimited files, you can also put items in quotations that you want to be treated as one column if you want to preserve commas. Thus the first line could also be represented as `Haiti,"92,000"` if we wanted the comma to be preserved

Data analysis tip: When creating your own data, you often have to work a lot with text editors/Excel. Using the "Find and Replace" option found in these programs can save time. First replacing the commas in the numbers with blanks, then replacing the spaces with commas would make formatting this dataset very easy.

2. Read in the dataset

   The general syntax to read the data in is:

   `insheet using *filename*, comma names case clear`,

   where ***filename*** is the filename (with extension) of your data. The words after the comma are options, giving Stata more information to successfully import the data. The option `comma` tells Stata that the file is comma-delimited. `names` tells Stata that the first row of data has variable names in it. `case` indicates that Stata should keep the case of the variable names (upper-case, etc). Finally, the `clear` option will clear the data currently in memory.

   Putting this all together gives the following command:

   `insheet using countries.txt, comma names case clear`.

Use both of these methods to input the data. After you input all the data, answer the following questions. You don't have to turn in anything for questions A and B. Their purpose is to get you familiar with Stata.

Questions:

A How many earthquakes killed 200,000 or more people?

With ten cases it's straightforward to look at the data and get an accurate count. But with a longer dataset, counting the incidences of each number "by hand" would be cumbersome. In such settings, you can make life easier by sorting the numbers in increasing order, then count the incidences. Let's do this in Stata just to get familiar with this command.

Not surprisingly, the command is simply `sort`. Type the variable names after the `sort` command in the desired order, in this case `sort Deaths`. The data is now sorted in ascending order by ***Deaths***. Alternatively, you can navigate to Data -> Sort -> Ascending Sort and follow the wizard.

Typing `browse` will open the Data Edtior and let you look at the data. The data can also be seen by typing `list [varname(s)]`, which will show the output in the Results window.

Data analysis tip: Stata is case sensitive. `sort Number` and `sort number` are commands referencing two separate variables, so an error will occur if you do not use the same case. Also, `Sort` is not a command while `sort` is. You can try out the four combinations (`SN, Sn, sN, sn`) and you'll see that only one is valid.

B If you want to sort the data first by country and then by death toll (i.e. have all countries in alphabetical order with fatalities listed in increasing order by country), which command would you use? Try them both to see what happens.

- `sort Deaths Country`
- `sort Country Deaths`

Data analysis tip: There is a more advanced form of sort called `gsort` which allows the user to specify the order (`ascending` vs. `descending`) that each variable should be sorted on. This can be navigated to by Data -> Sort -> Ascending and Descending Sort.

Loading Stata Data Sets

Load in the data set forbes94, which contains the 1994 compensation information for Chief Executive Officers (CEOs) of several large companies. You can download this dataset here. Note the extension is ".dta" This is a Stata-specific extension that for the most part can only be used by Stata.

Loading Stata datasets is the easiest way to import data. All you need is the `use` command: `use forbes94.dta, clear`.

When you get a data set, the first thing to do is to figure out how many variables and how many units of observation you have to play with. This is pretty easy in Stata through the `describe` command. The `describe` command will list information about the dataset and it's size as well as information about each variable. If you just want the information about the dataset, use `describe, short`. Alternatively, if you just want information about a specific variable, type `describe` *`varname`*. There are 800 CEOs in this data set and a mix of string variables (`str2`, `str16`, where the number is how many characters the system reserves for that variable) and numeric variables (`long`, `byte`, `int`, `float`, where the different types again refer to how the data is stored in memory).

Let's get into some data analyses. Compile your answers in a document to be printed out and turned in at the end of lab as your lab report. You're permitted and encouraged to talk about questions with your classmates, but write up your lab report with your own words. Feel free to ask for help from the TA or your classmates if you get stuck.

The TA will be grateful if your report is neat and gracefully written.

Also, remember to save all of your commands in a "do-file" for easy reference and make a "log-file" within it.

Questions:

1. Stata displays missing values with dots, `"."`. True or false: There are more than five CEOs whose values of total compensation are missing in the data file. (Hint: You can do this quickly using the `sort` command, then `browse`ing the data.)

Data analysis tip: Sometimes you want to be able to add notes into your do-file that aren't commands (for example, whether this question is True or False). To create a comment in Stata, simply begin a line of text with an asterisk (*) or a double backslash (\\).

Data analysis tip: It is common for some data to be missing in a file. Unfortunately, there is no universally accepted way of representing missing values. Some software packages, like Stata, use a dot or period. Other packages use an "NA" for not available. Some data producers, often federal agencies, use extreme values of a variable (e.g., -99) to indicate missing values. Using extreme values is bad practice: how does the user know if the value is correct as written or if it is a dummy entry to denote missingness? When you get a data set from someone, learn how they code missing data before doing any further analyses.

2. What is the salary (not total compensation) of the CEO of Duke Power?

   We need to search through the data base for Duke Power, then read off the salary of its CEO. One approach is to look at the company names row by row. For those who find comfort in tedium, this is the preferred approach. Although this is less tedious if you first `sort Company` then scroll through the browsed data to the D's.

   The other option would be to use one of the main features that Stata has to offer, the `if` qualifier. This allows us to subset a command to just "Duke Power". You can either `browse if Company=="Duke Power"`, or you can just `list` the variables you are interested in: `list Company Salary if Company=="Duke Power"` (Quotes are required for string variables).

   Data analysis tip: Note the use of two equal signs, `==`. Equality tests are performed using `==`, assignments of values are performed using `=`. Stata will complain if you use the two when you should use one, and vice versa. Type `help operators` for a complete list of Stata's operator syntax.

   Also, look at "Blockbuster Entertai" and "Walt Disney". It is mildly surprising that Blockbuster is considered a retail—not entertainment—industry and that Disney is a travel industry. Most data sets contain oddities if one looks closely.

3. Of all CEOs, which has the highest total compensation? Which has the lowest total compensation?

4. Which industry type has the highest average CEO total compensation? Be careful not to read the decimals incorrectly when you answer the question.

   There are way too many CEOs to figure this out by hand. Let Stata do all the work.

   First, note that Stata has a very useful command `summarize` that will give you basic summary statistics for a certain variables. `summarize TotalComp` will give you the mean, std dev, min, max, and count of ***TotalComp***. In order to find similar stats for each ***WideIndustry***, it's good to use tables.

   There are 3 basic ways that you can create such a table. Each row in the tables that we'll create here will list the mean, std dev and frequency of TotalComp by ***WideIndustry***.

(a) `tabulate` By default, tabulate calculates frequencies. So typing `tabulate WideIndustry` will tell you how many CEOs there are in each ***WideIndustry*** (try this). In order for it to give you different information (mean, std dev, freq), you simply use the `summarize` option:

```
tabulate WideIndustry, summarize(TotalComp)
```

(b) `table` The table command is very useful in that it gives you great flexibility in deciding how you want your table to be organized and what information you want. Like tabulate, you tell it the variable(s) by which you want the table to be organized, and then tell it which statistics you wish to have calculated for those variables, with the `contents()` option:

```
table WideIndustry, contents(mean TotalComp sd TotalComp N TotalComp)
```

(c) `tabstat` The tabstat command works a little differently. You first tell it the variables for which you are interested in getting summary statistics (***TotalComp***) and then tell it how to break it out (by ***WideIndustry***) and which statistics you want (mean, standard deviation):

```
tabstat TotalComp, by(WideIndustry) stats(mean sd N)
```

You can verify that each one of these three tables produces the same values for the mean and standard deviation.

Data analysis tip: Stata tries to automatically format numbers to give the best information. This is not always what you want, thus there is a `format(%`*`fmt`*`)` option on most commands that display output (not `tabulate`). The default format is `%9.3g`, where % tells Stata it's a format, 9 is the total output width, 3 is the number of digits to appear to the right of the decimal place, and g lets Stata to decide how many of the 3 digits to display, versus f, which tells Stata to show exactly 3. For our purposes, we do not care about cents, so your output may be easier to read/compare if you change the format to `%9.0f`, for example: `table WideIndustry, format(%9.0f) contents(mean TotalComp)`. See `help format` for more information.

Each row in the table reports the value of the statistic aggregated over the industries. For example, there are 62 CEOs in "Food" industries, and their average total compensation equals $2,740,661 with a standard deviation of $2,199,752.

5. How many of these CEOs got their undergraduate degree from Duke?

6. Let's assume all the CEOs from UNC schools graduated from UNC Chapel Hill. Assuming this, there are more CEOs with undergraduate degrees from Carolina ("U of North Caro") than there are from Duke ("Duke U"). Your friends at Carolina use this to argue that their graduates are more likely to be CEOs than Duke graduates. How can you respond to this?

Use the CEO counts to make a statistical argument that Duke does not lag behind Carolina in producing CEOs. Write two or fewer sentences to justify your answer. Hint: To come up with a good argument, you need information about UNC and Duke that is not in this data set. Once you've identified what you need, you can look up the information on-line. There is more than one correct answer (it's an argument, after all), but some answers are clearly wrong. Please be plausible.

7. Highest attained educational degree is in the variable ***GradDegree***. Which degree has the highest total compensation: MBA (business), JD (law), MD (physician), PhD, or no graduate degree? Use highest average total compensation as your criterion, and choose only from these categories.

   Data analysis tip: We cannot say definitively from these data that obtaining one degree results in higher compensation than obtaining other degrees. There are small numbers of people in some degree groups. Statisticians typically hesitate to make strong statements based on only a few observations. Plus, there could be lots of reasons why certain degrees have higher compensations than others; it may not be just the effect of the degree that drives compensation. We'll talk more about these issues throughout the course.

8. Explore the data to answer at least one question that interests you. Report your findings to the TA; you don't have to write anything on your lab sheet for this question. The TA will give you credit for answering this problem when you report to them. Ask your TA for help with Stata if needed.

You may want to begin your list of Stata commands by adding instructions for the methods you used in this lab. We'll use sorting and summarizing by groups for later labs, so it will be helpful to have commands for those data analysis tools handy. (Obviously, don't turn in this list.)

This ends the lab. Remember to turn in your lab sheet to the TA.

Include your name and lab time on the sheet.