Intro to Stata 1

1. **Before We Begin...**

   (a) **Purpose**
   To learn the very basics of Stata. Once you learn the basics, the *help* function within Stata will guide you in your future use of Stata. Also, the Stata manuals are great resources for understanding how to use commands and learning practical techniques of using Stata, because they provide clear examples of how to use commands.

   (b) **Format**
   First, I will provide instructions (approximately 10 pages) of how to use Stata or particular commands. At the end of the instructions, I prepare some problems, so you can check your understanding of what you have learned. I plan to post my answer together with a new Stata tutorial on the next Wednesday. This process will continue several times. After you finish this series of Stata tutorials, I am sure you can use Stata by yourself with reference to the *help* function within Stata.

   (c) **Note to Stata 12**
   As of April 2012, Stata 12 has been available in IUJ. Before Stata 12, Stata 10 was used in IUJ. Basically, there are no changes between Stata 10 and Stata 12 in terms of how to use it. The main differences between them are i) new commands are available in Stata 12 and ii) Stata 12 is more user-friendly than Stata 10. (For example, Stata 12 automatically does some preparatory work for you, and the Stata manual is available inside of Stata 12, which was not the case under Stata 10.) In this series of the tutorials, I will describe differences between Stata 12 and Stata 10 whenever necessary, so that even if only an older version of Stata is available after your graduation, you can use Stata without much trouble.

2. **Prepare for Analyzing Data**

(a) Create/Change Working Directory

First, let's create a working directory. You can create a folder called "stata-tutorial" under the hard drive of your PC (C:\temp\), or your own student folder, or whichever folder you have write-access. Within the directory "stata-tutorial," you can create a folder "tutorial1." We work on this directory throughout tutorial 1. A Stata data set called auto.dta is available under C:\Program Files (×86)\Stata12\. Please copy it into the folder "tutorial1" you have just created. Once you create your working directory, you need to tell Stata which directory is your working directory. To do so, you can type the following in the command line:

cd c:\temp\stata-tutorial\tutorial1

(or the appropriate directory path for your folder after the command *cd*) and press the return key. The cd at the beginning stands for change directory. If the name of your folder contains blank (e.g. my temp), you must use double quotes to tell Stata which directory is your working directory.

cd "c:\my temp\stata-tutorial\tutorial1"

In passing, "copy and paste" works for Stata in many cases. For example, instead of typing

cd c:\temp\stata-tutorial\tutorial1

you can just type cd, and for the directory path, copy and paste it from the directory address of a window.

You can always check your working directory by typing *pwd* in the command line and pressing the return key (Hereafter I do not write "press the return key" after each command, but you always need to press the return key after each command). *pwd* stands for present working directory. Type *pwd* in the command line. Stata will return your working directory. By the way, Stata is case-sensitive. For example, Stata does not recognize *Pwd* and returns an error message.

(b) Specify Data You Are Using

The command *use* is to tell Stata which data set you are using. Type the following in the command line:

use auto

To discard the data set from Stata's memory, use the command

clear.[1]

(c) To End Stata
To end Stata, first *clear* the currently-loaded data. Then type *exit* in the command line. I explained this now, so you can quit Stata whenever you want to.

(d) Five Windows + Data Browser/Editor
I assume that you already loaded auto.dta by the command *use*. You see five windows within Stata: Review window, Variables window, Results window, Command window, and Properties window.

i. Review window
All commands you typed after opening Stata are recorded here. This is the record of what you typed in the command line, including invalid commands. You do not need to type the same command repeatedly because the Review window is clickable. For example, suppose you loaded auto.dta and cleared. If you want to load auto.dta again, all you need to do is to click "use auto" in the Review window.

[1]Under Stata 11 or older, you need to pay attention to the size of your data set. The auto dataset (auto.dta) is a small dataset (only 6KB). You are able to open auto.dta without any difficulty under Stata 11 or older. However, when you try to open a large dataset under Stata 11 or older, Stata may complain, saying that your dataset is too large in comparison with the memory size allocated to Stata. If this happens in Stata 11 or older, check the size of the data set you are trying to use and the memory allocated to Stata. To check the memory allocated to Stata, type the following in the command line.

memory

If the allocated memory size is smaller than the size of the dataset you try to open, Stata complains. To change the allocated memory size, you can use the command *set memory*. For example, if you want to allocate 50MB to Stata, type

set memory 50m

in the command line. *set memory* must be used before you load your dataset. Now you have auto.dta in Stata's memory, you cannot change the memory size. To discard the data set from Stata's memory, use the command *clear*. The max memory you can allocate to Stata is the RAM size of your PC. Because RAM is a scarce resource, I advise you to set memory at a somewhat larger level than your dataset. For example, if your dataset is 50MB in size, 70MB or so is enough for Stata to complete basic tasks. Under Stata 12 (or a newer version of Stata in the future), this memory-management work is automatic, so you do not need to do anything as long as the size of your data set is well below the RAM size of your computer.

ii. Variables window
All variables available in the currently-loaded dataset are shown here. The Variable window is also clickable. Instead of typing a variable name in the Command window, you can click the wavy arrow in front of each variable in the Variables window. (The wavy arrow appears once you point a variable using the mouse.)

iii. Results window
Stata returns results in this window. You can copy the results and paste them into MS Excel, Word, or other software.[2]

iv. Command window
You type commands in this window.

v. Properties window
This window shows the properties of a variable highlighted in the Variables window and the properties of the data currently being used.

vi. Data Browser
Near the top of your screen, you see the icon with a magnifying glass. By clicking this icon, you can see the contents of your data. You see that auto.dta contains 74 observations (74 rows) and 12 variables (12 columns). If you opened the Data Browser, please close it now.[3]

vii. Data Editor
Next to the icon for Data Browser, you see a similar icon with a pencil. This icon is for data editing. By clicking this icon, you get a Data Editor. Data Editor looks like Data Browser. The only difference is that you can edit data in Data Editor but cannot in Data Browser. Data Editor is rarely used. To avoid changing the contents of the data by mistake, I recommend you use Data Browser rather than Data Editor if you do not need to change the contents of the data.

[2]When copying a table, use "Copy Table" under the Edit menu. Then a pasted table is nicely formatted in MS Excel and Word.

[3]In Stata 11 or older, you must close a Data Browser (or a Data Editor to be explained soon) before you do something else in Stata. In Stata 12 (or newer in the future), you can leave a Data Browser or a Data Editor open and continue working with Stata for other commands.

(e) Help Function

Like other software, the Help menu at the top of your screen helps you a lot in your future use of Stata. Under the help menu, you see both "Search" and "Stata Commands." If you have a specific command you want to know how to use, choose "Stata commands." Otherwise, use "Search." Also, the command *findit* is very easy to use and powerful (You need internet connection to be able to use the command *findit*). For example, type the following in the command line:

findit memory

What do you get? You see replies in a pop-up Viewer window. All words in blue in the Viewer window are clickable. Further, what is newly available under Stata 12 (and newer in the future) is the Stata manuals inside of Stata. Under the help menu, choose "PDF Documentation." This gives you access to the Stata manuals which are very useful because of abundant examples of how to use specific commands.

(f) Log File

A log file records the commands you used and the results Stata returned. To start recording, type

log using *filename*, replace

in the command line, where *filename* is any name you can choose. *replace* at the end is optional (You will see the meaning of *replace* soon). Stata records all commands you will use and the results Stata will return until you tell Stata to stop recording. To stop recording, type

log close

in the command line. After you type *log close*, your log file is created in the present working directory, with the file name you chose, plus the extension *smcl*. *smcl* files are Stata specific. You can open *smcl* files with a text editor such as Notepad, but it is not friendly to human eyes. To convert *smcl* files to text files, type

translate *filename*.smcl *filename*.txt, replace

in the command line, where *filename* is the filename you chose. *replace* at the end means that if *filename*.txt already exists in the present working directory, the newly converted text file overwrites the old text file with the same name. If you do not want this, you

can issue the same command without ,(comma) replace. If there already exists a text file with the same name, Stata complains. The option *replace* does the same in the command

log using *filename*, replace

which was used to open a new log file. If the present working directory has already an *smcl* file with the same file name in it, Stata replaces the old file with the same file name by the new file. Finally, you must always close a log before starting a new log. If you try to open a new log file when you already opened a log file, then Stata complains.

(g) Do File

So far, we used Stata interactively. That is, you issue a command, and Stata returns a result. You issue another command, and Stata returns another result. Often it is more convenient to prepare all commands beforehand, and run them all at once. To do so, you need to create a *do* file. Using *do* files is better than running Stata interactively in the following two senses. First, you can reproduce your results whenever you need. Second, you can stop using Stata whenever you like. Imagine you work interactively with Stata and did a lot of data management before running regressions. Before you run regressions, you really need to go home. Then you cannot log off your computer without losing all your data-management work. To create a *do* file, go to the Window menu at the top. Under the Window menu, choose Do-file Editor. Perhaps your *do* file begins with

log using *filename*, replace
use *datafilename*

and ends with

log close

In *do* files, you must write one command in one line, and before starting the next command, you must press the return key (You will learn how to continue one command into another line later, but for the time being, you can understand that each line contains one command and that commands are separated by pressing the return key). Do not forget to press the return key after the last line of your *do* file. Once you finish creating a *do* file, save your *do* file (using the File menu within the Do-file Editor) in your present

working directory, with any name of your choice. To run a *do* file, type

do *filename*

in the command line, where *filename* is the name of the *do* file you chose. To open an existing *do* file, go to the Window menu inside of Stata, then choose Do-file Editor and New Do-file. Inside of the Do-file Editor you have just opened, go to the File menu and Open.

(h) This is all you really need to know before analyzing data. I believe you are ready to use Stata now, and even when you face some difficulties, you can solve them with reference to the Help function or by consulting with someone else who is familiar with Stata. You will quickly become comfortable with Stata as your experience of Stata accumulates.

3. Analyzing Data

(a) We use auto.dta to demonstrate some Stata commands. Please load auto.dta in Stata's memory by the command *use*. This dataset is about automobiles sold in the US market in 1978 (real data, not fictitious). The data set contains the name of the car (make), the price (price), miles per gallon (mpg), a repair record (rep78), weight (weight) and length (length) of the car, whether the car is made by a foreign car-manufacturing company or not (foreign), and other pieces of information (headroom, trunk, turn, displacement, gear_ratio).

(b) *summarize*
summarize returns the number of non-missing observations, mean, standard deviation, minimum, and maximum of the variable. For example, type

summarize price

in the command line. Stata returns relevant information. You can summarize multiple variables at one time, so

summarize price weight

is a valid command. If you just type *summarize* without any variables, Stata summarizes all variables in the currently-used dataset. By the way, you may notice that Stata returns zero non-missing observations if you *summarize* the variable make (the name of the

car). You will learn why this happens later in Tutorial 2 or 3, but this is related to how data are stored in the Stata dataset (make is a so-called *string* variable, while the rest of the variables are *numerical* variables.)

(c) *tabulate*

You can tabulate a variable by using the command *tabulate*. For example, type

```
tabulate price
```

in the command line. You get the frequency, relative frequency, and cumulative relative frequency of price. In passing, you may see "–more–" in blue fonts after you use the command *tabulate*. To look at the rest of the table, just press the return key or the space key, or click "–more–" (blue fonts) on the screen (Blue fonts on screens or in pop-up windows are always clickable in Stata). Or if you hate "–more–", you can type

```
set more off
```

in the command line, and Stata never stops in the middle of a table. To put "–more–" back, you can type

```
set more on
```

in the command line.

You can create a two-way table as well in addition to one-way tables we just described. To do so, you naturally need two variables. For example, type

```
tabulate foreign rep78
```

in the command line. Stata returns the frequency of the number of repairs separately for domestic and foreign cars. For example, there are two domestic cars in the dataset, that were repaired only once in 1978. If you want to create one-way tables for multiple variables at one time, you can type, for example,

```
tab1 foreign rep78 weight
```

in the command line. As I said before, you do not need to type "foreign." Remember that the Variable Window is clickable. Just type tab1 and click variables of your choice in the Variable Window.

(d) If Qualifier

Sometimes you are interested in only a subset of the sample. Suppose you are interested in only reliable cars as candidates for the

next purchase. Suppose also you are an eco-friendly person. You may want to know the mean of miles per gallon for those cars whose repair records are two or less. Then, you can type

summarize mpg if rep78<=2

in the command line. If you are interested in only cars whose repair records are only once. Then, type

summarize mpg if rep78==1

in the command line. Note double equal signs after the *if* qualifier. After an *if* qualifier, you can use <, >, <=, >=, ==, and ~=, where ~= means not equal to.[4] Also, you can use multiple *if* qualifiers at one time. For example, you can type

summarize mpg if rep78==1 | rep78==2

in the command line, meaning that you are interested in only cars whose repair recodes are only once *or* twice. Note that you need " | " after an *if* qualifier to represent "*or*." You also need to know that you can type

summarize mpg if rep78==1 & price<5000

in the command line, meaning that you are interested in only cars whose repair recodes are only once *and* whose prices are less than 5000. Note that you need " & " after an *if* qualifier to represent "*and*."

(e) By-able
Both *summarize* and *tabulate* (including *tab1*) are by-able. Suppose you are interested in the mean miles per gallon separately for foreign and domestic cars. Then you can type

bysort foreign: summarize mpg

in the command line.

(f) We've learned a lot. This is enough for the first tutorial. Below is some problems. You will need to use the *help* function to solve some problems. But you should be able to solve all of them because you have gone through all of the instructions above. Finally, it is better to know this. You can always tell Stata to stop the command you issued, by pressing the red cross icon near the top. For example, your command by mistake requires Stata to produce a huge table (Type "tabulate weight mpg" in the command line), you can always cancel the command by pressing the red cross icon.

[4] != also means not equal to.

4. Tutorial 1 Homework

   (a) Create and run a *do* file that implements the following:

      i. You consider purchasing a car. One problem is that your garage is small. Physically, it is impossible to park a car with the length 200 inches or longer in your garage. How many cars in the dataset qualify this restriction, so they are candidates of your next purchase?

      ii. You are also an eco person. You are interested only in cars whose miles per gallon are 20 or more. How many cars qualify this restriction in addition to the previous restriction?

      iii. How many missing observations does the variable rep78 have? Hint: . (dot) represents missing in Stata. Use the *help* function to look up *tabulate* to learn how you can use the option *missing*.

      iv. Calculate the mean price of economic cars separately for foreign and domestic cars, where economic means miles per gallon are 20 or larger.

      v. You want to know how many percentages of domestic cars have repair records of 3 out of all domestic cars (you can ignore missing observations). Similarly, you want to know how many percentages of foreign cars have repair recodes of 3 out of all foreign cars (you can ignore missing observations, again). Calculate these in one command. Hint: There are two ways to do this. One way is to use the command *tabulate*. Look up *tabulate* to learn how to use the options *column* and *row*.

   (b) If you have questions, stop by one of TA's. Stata Exercise 1 is due at the beginning of the class on April 11 (Wednesday). No late assignment will be accepted. You need to turn in a hard copy of your result file (convert your *smcl* file into a *txt* file). You MUST create your result file from a *do* file. If your result file is created without using a *do* file, you get zero credits from this assignment. Please do not include error commands in your *do* file. After printing out your *txt* file, put your name and student ID by hand. Also, please include the question numbers i. through v. by hand in front of your answers to the corresponding questions.

End of Tutorial 1