17.874 6/23/2004
TA : Jiyoon Kim

Recitation Note 1

This is intended to walk you through using STATA in an Athena environment. The computer room of political science dept. has STATA on PC machines. But, knowing how to use it on Athena will be helpful, because you can access to machines all over the campus.

First, log on Athena, then at the athena prompt type:

```
athena% mkdir STATA
```

Makes a folder named STATA in your home directory. You can work on STATA within this directory.

```
athena% cd STATA
```

Changes directories. Before, you were in your home directory. By using "cd" command, you can change your directories. If you want to know in which directory you are now, type:

```
athena% pwd
```

Now, you want to read your data. For this exercise (and probably for the whole class), use data files in my public directory. Wanna know how to get data from another person's directory? Public directory is the one which is accessible by everyone without permission. If you want to copy a file from other person's public directory (in this case, mine), first type:

```
athena% attach jiyoon*
athena% cd /mit/jiyoon
```

You are now in my directory. Check it out by using "pwd" command.

In my public directory, I created a subdirectory for this class called "babystat." (isn't it cute?) In this subdirectory, there is a sample text data file called "tv". Let's copy this file to your directory of "STATA."

```
athena% cd Public/babystat
athena% cp tv /mit/your username/STATA
athena% cd /mit/your username/STATA
```

Now you are back into your directory to work on. Let's see if the file was safely copied.

```
athena% ls
```

It will show you the contents of the directory.

* any username of the person whose directory you want to work on.

Let's get to work on data file. Fist, open an emacs file, which is the basic text editor on Unix. (claimed to be the cheapest.) You will see emacs button in the bottom of your window. Simply click it or type:

```
athena% emacs &
```

You will see a buffer window popping up on the screen. Click on the "files" tab and open the file "tv." Then, now you will see some data points.

```
3    3.5
2    3.4
6    3.0
14   2.3
5    3.6
8    2.8
10   1.9
11   2.0
4    3.3
3    3.7
```

I simply made up the data. The first column is hours of watching TV and the second column is GPA. Save this file with the extension of .raw, which saves it as ASCII file.

Move on to STATA. Come back to athena prompt and type:

```
athena% add stata
athena% xstata
```

These two commands load and run STATA. You will be in STATA now. Version 7 features windows-like interface, which is nicer than the old unix interface. You will find a command line at the bottom, readout at the top, command summary box at the top left and variable list box at the bottom left. You can call up a variable or old command by clicking on it. Also, you can use tabs at the top to play with commands without typing them.

Talking about STATA, there are four kinds of files you can create.

- .do : with lots of command lines. By creating this file, you don't need to repeat writing all the commands when you made mistakes.
- .log : it contains all the commands and outputs that you see on the screen. You probably will hand in this file in the class.
- .dta : data file
- .gph :graph file

Having log file is very important since it keeps all the commands and outputs. Without it, once you log out of STATA, you would be empty handed. Warning! Since log file keeps all the commands, of course it will keep all the mistakes you made and ugly error messages. Not quite beautiful, huh. That's why you need to write do file. Write a do file without any mistaken command, and simply run it. You will have a clean log file, then. I am attaching a sample do-file for you.

Let's learn how to use infile command. This command will read ASCII data in text format into STATA data file format with .dta extension.

· infile tvhours grade using tv.raw

Note that the infile command works by looking at the first number or letter, assigning it to the first observation of the first variable. It then looks at the next letter or number, and assigns it the first observation of the second variable. And so forth.

· list

You will see the data in the data editor.

Two other different command to read datasets are "infix" and "insheet." "infix" is used for the fixed field data. i.e. ICPSR data with code book. "insheet" is used for .cvs file that you can create through Excel. We will cover them later, but now you need to know infile command to do some fun exercise.

Well, let's try :

· summarize

· describe

See what you've got there.

Once you created data set with infile command, try "compress." It will save memories.

Now, let's work on REAL statistics. The basic but most important commands that you should know are :

gen (replace)
egen
regress ***dependent variable independent variable***
predict ***some new variable name (under which predicted values of dependent variable will be stored.)***
graph ***dependent variable(y-axis) independent variable(x-axis)***

"gen" creates variables. Once you created a variable and want to change the value of the variable, you have to use "replace" instead of "gen". "egen" is a special kind of generating command. When you need a particular value of your observations, such as mean, maximum and minimum (or there are all different kinds of functions that you can operate with egen), egen is the one that you should work with. "regress" – quite self-explanatory as well as "graph." "predict" will generate a new variable which has values produced by your regression model. We'll get back to that later.

Let's try these commands.

· reg grade tvhour

· graph grade tvhour

You can see tvhours has a negative effect on grade. How surprising. If you want to save the graph, simply add ", saving(tv)." Or you can use any graph file name you want instead of tv. In order to create a new variable, use gen. Suppose you want to create gender variable.

· gen gender = 1

This is a discrete variable with two classifications – male or female. I put "1" for male and "0" for female. First, you simply generate the variable gender with values of all 1. Now, you need to change some observations, because not all of them are male. Let's assume that all those whose grades are greater than 3.0 are female. (no offense, but, after all, I AM a TA.)

· replace gender = 0 if grade >= 3

See, you have to use "replace" not "gen", gender variable is already created.

· egen average = mean(grade)

The above command will give you the mean value of grade of the sample.

· reg grade gender

· graph grade gender

You will probably see some funky graph with two vertical lines. This is because gender is a discrete variable.

Now, let's get back to our previous regression of grade on tvhours. Do you remember your graph file you saved? In order to print out this graph, you have to translate it into postscript file. The command line goes:

· translate tv.gph tv.ps

Then, go back to athena prompt:

```
athena% lpr –Pprintername tv.ps
```

Now you have graph printed.

Let's get out of STATA and enjoy the rest of Friday. First, save the data file into the STATA format.

· save tv
· clear
· exit

Wait! you did not created a log file to record your STATA session so that you can take a look at your research. Man, let's go back to the STATA.

```
athena% xstata
```

· use tv

This will bring your data. (note that you can simply type "tv" if you are within the directory. If the data is outside your current directory, type the whole path such as /mit/*your usename*/*directory name*/tv.)

· log using tv.log

You just opened a log file. Now, retype all the commands you did right before. (see, do-file does good to you.) When you finished typing and doing stuffs, type this to close log and get ready to get out.

· log close

This closes your log and saves it in your STATA directory as tv.log.

· exit

If you want to see your log file, type :

```
athena% lpr –Pprintername tv.log
```

Good job!

```
------------------------------------------------------------------------------
       log:  /afs/athena.mit.edu/user/j/i/jiyoon/17.842/tv.log
  log type:  text
 opened on:   7 Sep 2001, 13:23:26

. infile tvhours grade using tv.raw;
(10 observations read)

. compress;
tvhours was float now byte

. summarize;

    Variable |     Obs        Mean   Std. Dev.       Min        Max
-------------+-----------------------------------------------------
     tvhours |      10         6.6   4.005552          2         14
       grade |      10        2.95   .6720615        1.9        3.7

. describe;

Contains data
  obs:            10
 vars:             2
 size:            90 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
tvhours         byte   %9.0g
grade           float  %9.0g
-------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved

. gen gender = 1;

. replace gender = 0  if grade >= 3;
(6 real changes made)

. reg grade tvhours;

      Source |       SS       df       MS              Number of obs =      10
-------------+------------------------------           F(  1,     8) =   29.65
       Model |  3.20117753     1  3.20117753           Prob > F      =  0.0006
    Residual |  .863822593     8  .107977824           R-squared     =  0.7875
-------------+------------------------------           Adj R-squared =  0.7609
       Total |  4.06500013     9  .451666681           Root MSE      =   .3286

------------------------------------------------------------------------------
       grade |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     tvhours |   -.148892   .0273454    -5.44   0.001    -.2119505   -.0858335
       _cons |   3.932687   .2082561    18.88   0.000     3.452448    4.412926
------------------------------------------------------------------------------
```

```
. graph grade tvhours, saving(tv, replace);

. translate tv.gph tv.ps, replace;
(file tv.ps written in .ps format)

. save tv, replace;
file tv.dta saved

. log close;
       log:  /afs/athena.mit.edu/user/j/i/jiyoon/17.842/tv.log
  log type:  text
 closed on:   7 Sep 2001, 13:23:27
-------------------------------------------------------------------------------
```

STATA Do-file

```
#delimit;

clear;

log using tv.log, replace;

infile tvhours grade using tv.raw;
compress;

summarize;
describe;

gen gender = 1;
replace gender = 0  if grade >= 3;
reg grade tvhours;
graph grade tvhours, saving(tv, replace);
translate tv.gph tv.ps, replace;

save tv, replace;
log close;
```