UN3412 Introduction to Econometrics

Recitation 1

Suanna Oh*

Spring 2020

This week's recitation introduces STATA, a powerful statistical software package that facilitates both management and analysis of data. This document will cover some STATA basics and then guides one through some simple analysis. Future recitations will apply the econometric concepts taught in lectures to real world data via STATA.

Contents

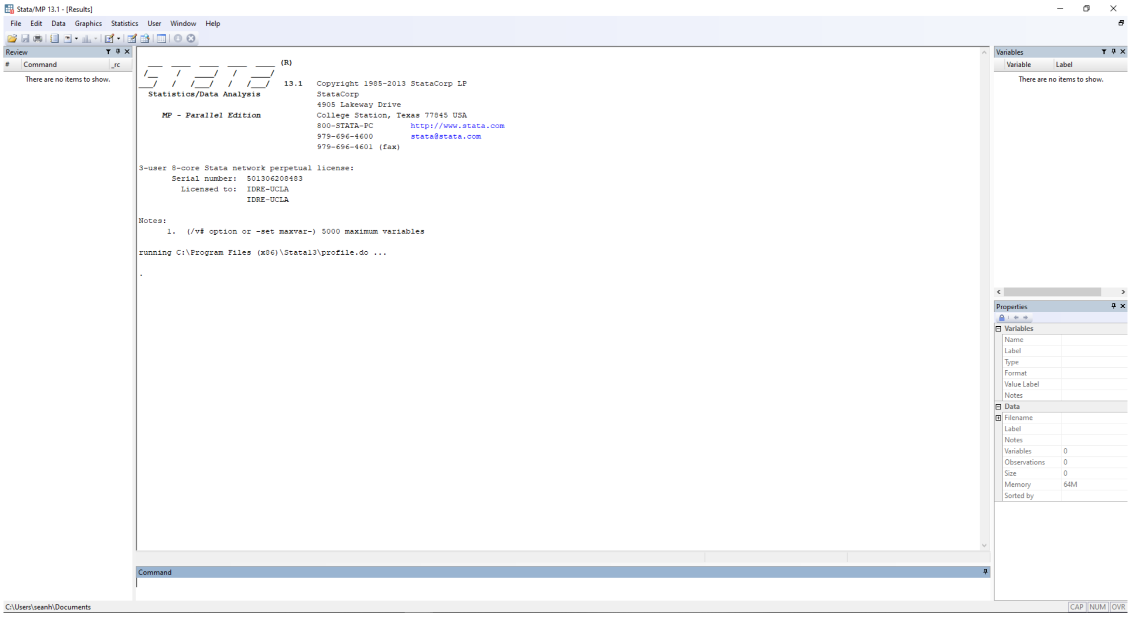
*Revisions by Sean Hyland; errors are likely mine.

1 A Brief Introduction to STATA

1.1 Opening STATA

Mac: In the CU computer labs, open STATA using `Spotlight Search > StataSE 14`.
Windows: In the CU computer labs, go to `Start > All Programs > StataMP 13`.
When STATA starts up, you will see a menu bar, a set of buttons, and usually 4 or 5 windows:

Figure 1: Stata Interface



```
  ___  ____  ____  ____  ____ (R)
 /__    /   ____/   /   ____/
___/   /   /___/   /   /___/   13.1   Copyright 1985-2013 StataCorp LP
  Statistics/Data Analysis            StataCorp
                                      4905 Lakeway Drive
     MP - Parallel Edition            College Station, Texas 77845 USA
                                      800-STATA-PC        http://www.stata.com
                                      979-696-4600        stata@stata.com
                                      979-696-4601 (fax)

3-user 8-core Stata network perpetual license:
       Serial number:  501306208483
         Licensed to:  IDRE-UCLA
                       IDRE-UCLA

Notes:
      1.  (/v# option or -set maxvar-) 5000 maximum variables

running C:\Program Files (x86)\Stata13\profile.do ...

.
```

Variables 0
Observations 0
Size 0
Memory 64M

1. The results window (top center) will show the output of every procedure you run.
2. The variables window (top right) will list the variables (and their labels) in your dataset.
3. The command window (bottom center) is where you can type your STATA code. Whilst it is possible to run most (all?) core procedures via the drop down menus we shall utilize the command line in this course.
4. The review window (upper left) will show the history of the commands submitted during a STATA session.

1.2 Help

To search for a concept, go to `Help > Search` and enter some keywords or type something like: `search linear regression`

in the command window.
Alternatively, `findit <keywords>` searches both STATA's help and the internet.
If you know the command you need help with, type: `help <commandname>`.

1.3 Command Syntax

Most STATA commands have the following syntax:
`command` *`variable(s)`* `[if expression] [in obs. range] [weights] [using` *`filename`*`], [options]`
The `command` is performed on the *`variable(s)`*. If applicable, a particular subset of observations may be selected by the `if` and `in` modifiers, and the specific ways the command should behave are controlled by `options`.

1.4 Typing Commands

In the Command window, type: `display 2+2`
The `display` command causes STATA to show the results in the Results window.
Now try: `Display 2+2`
You will get a red error message since STATA does not recognize this command; commands are case-sensitive.

1.5 Directories

Before opening a STATA file you should always define the location of your working folder. This is where you will keep the data and output for a given project.
To check the current working directory type: `pwd`
The current working directory will be reported in the results window.
To change the current working directory type: `cd` *`filepath`*
For example, create a sub-folder called "Recitation 1" in your documents folder, and then type: `cd "/Users/`*`username`*`/Documents/Recitation 1"`
replacing *`username`* with your Windows username - in the CU labs this will be your UNI.
This will change the directory to the sub-folder you just created.
Note: the quotation marks are required if there are spaces in the filepath.

It is often desirable to import/export files from/to multiple different locations. This could be achieved by specifying the full filepath when required, or through multiple working directory changes. Instead, I prefer to set a shortcut for each key folder via a global.

We define a global as follows:

```
global globalname filepath
```

For example, we could define the ‘rec1’ global as follows:

```
global rec1 "/Users/username/Documents/Recitation 1"
```

After defining a global instead of typing out the full file path we reference the directory (and a potential subdirectory) as follows: `${globalname}subdirectory`.

For example: `cd ${rec1}`

1.6 Log Files

I recommend opening a log file for each STATA session; a log is a text file which records both the commands you type and the corresponding output as it appears in the results window. This allows you to examine your results without STATA, and provides a permanent record of your analysis.

To open a log file named ‘example1’, type: `log using example1.log, replace`

This creates the file `example1.log` in the working directory, which can be read with any text editor such as Notepad. The `replace` option means that the log file will be overwritten if it already exists.

We can close the log file by typing: `log close`

1.7 Loading Data

A dataset named `auto.dta`, where the suffix indicates the file is a STATA dataset, has been saved to Courseworks; navigate to `Recitations > Recitation 1` and download the data to the working directory you defined earlier.

You can then load the data into STATA by typing the command: `use auto.dta, clear`

The `clear` option tells STATA to clear it’s memory before loading the data; if you already have data in memory and do not specify the `clear` option you will get an error message.

Alternatively, we can also import data from the web. The following command imports the same dataset from stata.com:

```
use "http://www.stata-press.com/data/r9/auto.dta", clear
```

Note that the `use` command will load a STATA dataset only.

To import a csv file, type: `insheet [varlist] using filename [, options]`

To import an excel file, type: `import excel [varlist] using filename [, options]`

1.8 Data Management

Some useful data manipulation commands include:

Command	Description
`drop x y`	Drop variables named x and y
`keep x y`	Drop every variable except x and y
`replace x=y`	Replace variable x with expression y
`rename x y`	Rename variable x "y"

1.9 Saving Datasets

To save your dataset type: `save` *`filename`*`, replace`
Note that the `replace` option will overwrite the *`filename`* dataset, so be careful with using this option.

1.10 Close the Log File and Exit STATA

You can close your log file by typing: `log close`
You can exit STATA by typing: `exit, STATA`. The command will return an error message if the dataset features unsaved changes. However, you will be prompted to save your do-file before exiting if you have made unsaved changes.

1.11 Do-files

Instead of typing your commands one by one into the command line, you can write a ".do" file, a text file which can be executed by STATA, and run a batch of commands at once. This ensures either you or others can reproduce your analysis in the future. We will return to this idea after first working through the next section interactively.

1.12 Miscellaneous

The following commands may be of use in future weeks.

- `capture` *`command`*: `capture` executes `capture command`, suppressing all its output including error messages. This is useful as an error in a do-file will cause the terminate when an error is discovered.
- `mkdir "filepath/folder"`: creates the *folder* folder in the *filepath* directory, provided the latter exists.

- You can interrupt any STATA command, which may be desirable if the code is running for too long, by pressing `q` or hitting the Break button Break.

2 Data Analysis and Working Example

In this section we will perform some elementary analysis on the `auto.dta` dataset; if you have not already done so, load the data by following the steps in subsection 1.7

2.1 Describe the data

Let's first describe the data. Type: `desc`
STATA will display the number of observations, as well as the number of variables, their names and labels.
We can also describe a subset of variables by typing: `desc [varlist]`

2.2 Inspect the Data

Let us now examine the make of the first 5 observations, sorted alphabetically by make. Type the following two lines sequentially:

```
sort make
list make in 1/5
```

Note that the '`in`' qualifier allows you to specify the observations to list.

You can also view the data by using `Data > Data Editor > Data Editor (Browse)`, or hitting the data browser button Data Browser.
Alternatively, you can both view and manually edit your data in the Data Editor (`Data > Data Editor > Data Editor (Edit)`, or Data Editor). However, manually editing data is not recommended.

2.3 Summarizing the Data

To obtain summary statistics of the data, type:

```
sum
```

STATA reports (i) the number of observations, (ii) the means, (iii) standard deviations, and (iv) minimum and (v) maximum values of all the variables in the sample (or the subset of variables if that was specified). For example, we see that the mean *mpg* is 21.3.

Note that the table suggests that *make* says has 0 observations, even though we observed the data in teh previous subsection. This is because STATA is reporting the number of non-missing numeric observations, and *make* is a 'string' variable. Similarly, the variable *rep78* has only 69 observations because the values of *rep78* are missing for some observations. Stata denotes missing values with a "." for numeric variables and " " for string variables.

In order to obtain a more detailed summary of the variable *mpg*, type: `sum mpg, detail`
The `detail` option tells STATA to also report the focal percentiles, as well as the skewness and kurtosis of the variables specified. Since the skewness is positive, we know that mpg has a long right tail.

To summarize *mpg* for only foreign cars, type:

```
sum mpg if foreign==1
```

Note that '`if`' is followed by a logical statement: foreign==1. As such, the summary values of *mpg* pertain only to foreign made cars.
The symbols for logical statements include:

Symbol	Meaning
>	Strictly greater
<	Strictly less
>=	Greater or equal
<=	Less or equal
==	Equal
!= OR ~=	Not equal

Thus one way to summarize *mpg* for domestic cars is to type: `sum mpg if foreign!=1`
Alternatively, we could have generated summary statistics by group as follows:

```
sort foreign
by foreign:  sum mpg
```

To obtain more detailed summary statistics for foreign cars, type:

```
sum mpg if foreign==1, detail
```

Note how the STATA command structure allows you to use an if statement followed by the option detail.

2.4 Correlations and Covariances

You can quickly obtain the correlation between a set of variables with the following command: `corr [varlist]`
For example, type: `corr mpg weight`
The output is a lower traingular matrix with ones on the diagonal, implying a variable is perfectly correlated with itself, and a value of -0.8072 on the off-diagonal, which is the sample correlation coefficient between the two series.

Instead, we might be interested in the covariance between a set of variables. This is obtained by the following command: `corr [varlist], cov`
Again the output will be a lower triangular matrix but now with variances on the diagonal, and covariances in the off-diagonal cells.

2.5 Graphs

To see the empirical distribution of *mpg*, type: `histogram mpg`
The graph shows us that *mpg* has a long right tail, consistent with the positive skewness we observed earlier.
You can save this graph by typing: `graph export histogram_mpg.png, replace`

We can examine the bivariate association between *mpg* and *weight* by typing:

```
scatter mpg weight
```

The scatterplot shows heavier cars in our sample tend to have lower mpg.

For more on graphs, see `help graph`.

2.6 Generating Variables

The variable *weight* is in pounds. Let's generate a variable for *weight* in 1000s of pounds.

```
gen weight_1000 = weight/1000
```

Now summarize the variables weight and weight_1000 to see that the variable was created correctly: `sum weight weight_1000`

Other useful mathematical expressions in STATA include:

Symbol	Meaning
$+$	Add
$-$	Subtract
$*$	Multiply
$/$	Divide
ˆ	Power
`sqrt(expression)`	Square root
`exp(expression)`	Exponential
`ln(expression)`	Natural logarithm

The command egen, which is an extension to the generate command (see help egen), is used to create variables that require some additional function in order to be generated. For example, we can generate a variable equal to the sample mean of *mpg* as follows:

```
egen mpg_bar = mean(mpg)
```

2.7 Testing the Difference Between Two Means

Suppose you wanted to test whether foreign and domestic cars have the same mean mpg.
The null hypothesis is: $H_0 : \mu_{domestic} - \mu_{foreign} = 0$.
The alternative hypothesis is: $H_1 : \mu_{domestic} - \mu_{foreign} \neq 0$.
We can test this hypothesis using the Student's t-test in STATA as follows:

```
ttest mpg, by(foreign)
```

We see that the average *mpg* of domestic made cars in our sample is 19.82, whereas the average *mpg* among foreign made cars 24.77. The table also reports the difference between the two means, the standard error of the difference, and the 95% confidence interval and p-value of associated with the difference; the small p-value leads one to reject the null hypothesis that foreign and domestic cars have the same mean mpg at all conventional significance levels.

2.8 Linear Regression

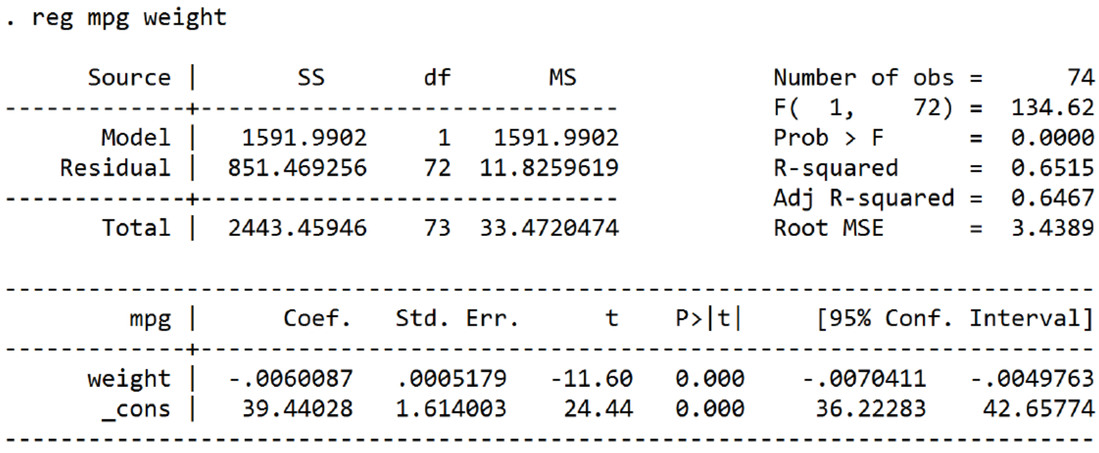
(copied/modified from http://www.ats.ucla.edu/stat/stata/output/reg_output.htm)

In subsection 2.5 we examined the bivariate association between *mpg* and *weight* through a scatterplot; let us now estimate the linear regression of *mpg* and *weight*. To do so, type:

```
regress mpg weight
```

You should see the following output:

Figure 2: Regression Table for *mpg* on *weight*



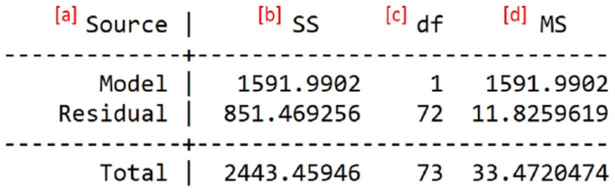
```
. reg mpg weight

      Source |       SS       df       MS              Number of obs =      74
-------------+------------------------------           F(  1,    72) =  134.62
       Model |   1591.9902     1   1591.9902           Prob > F      =  0.0000
    Residual |  851.469256    72  11.8259619           R-squared     =  0.6515
-------------+------------------------------           Adj R-squared =  0.6467
       Total |  2443.45946    73  33.4720474           Root MSE      =  3.4389

------------------------------------------------------------------------------
         mpg |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |  -.0060087   .0005179   -11.60   0.000    -.0070411   -.0049763
       _cons |   39.44028   1.614003    24.44   0.000     36.22283    42.65774
------------------------------------------------------------------------------
```

Note that you can now obtain the fitted values for *mpg*: `predict mpghat`
and the residuals of the regression: `predict ehat, resid`
which will be stored in the new variables mpghat and ehat, respectively.

Here's what the regression output means:

- In the top left hand corner, there is the Analysis of Variance (ANOVA) Table:

Figure 3: ANOVA Table for Regression of *mpg* on *weight*



```
  [a] Source |   [b] SS    [c] df   [d] MS
-------------+------------------------------
       Model |   1591.9902     1   1591.9902
    Residual |  851.469256    72  11.8259619
-------------+------------------------------
       Total |  2443.45946    73  33.4720474
```

[a] Source - Looking at the breakdown of variance in the outcome variable, these are the categories we will examine: Model, Residual, and Total.
The Total variance is partitioned into the variance that can be explained by the independent variables (Model) and the variance that is not explained by the independent variables (the Residual, sometimes called Error).

[b] SS - These are the Sum of Squares associated with the three sources of variance: Total (TSS), Model (ESS), and Residual (SSR).

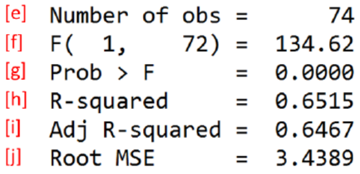
[c] df - These are the degrees of freedom associated with the sources of variance. The total variance has $N - 1$ degrees of freedom. The Model degrees of freedom

corresponds to the number of coefficients estimated minus 1. Including the intercept, there are 2 estimated coefficients in this example, so the model has 2-1 = 1 degree of freedom. The Residual degrees of freedom is equal to the number of observations, less the number of estimated coefficients: 74-2=72.

[d] MS - These are the Mean Squares, the Sum of Squares divided by their respective df.

- In the top right hand corner, there is the table for overall model fit.

Figure 4: Model Fit of Regression of *mpg* on *weight*



```
[e] Number of obs =      74
[f] F(  1,    72) =  134.62
[g] Prob > F      =  0.0000
[h] R-squared     =  0.6515
[i] Adj R-squared =  0.6467
[j] Root MSE      =  3.4389
```

[e] Number of obs - This is the number of observations used in the regression analysis.

[f] F(1, 72) - The F-statistic is the Mean Square of the Model (1591.9902) divided by the Mean Square of the Residual (11.8259619), yielding F=134.62. The numbers in parentheses are the degrees of freedom for the Model and Residual from the ANOVA table.

[g] Prob > F - This is the p-value associated with the above F-statistic. It is used in testing the null hypothesis that all of the model coefficients are equal to 0.

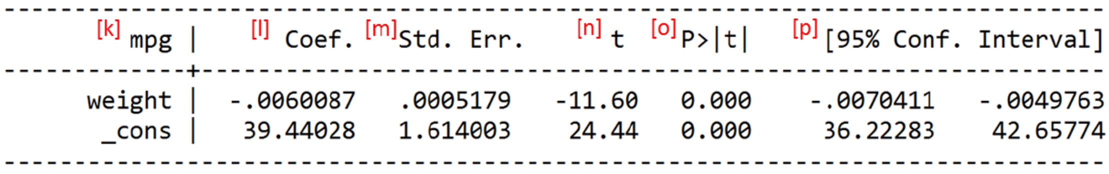
[h] R-squared - This is the proportion of the variance in the dependent variable which can be explained by the independent variable. This is an overall measure of the strength of association.

[i] Adj R-squared - This is a modified version of the R-squared which introduces a penalty for the addition of extraneous predictors to the model. Adjusted R-squared is defined as: $\bar{R}^2 = 1 - (1 - R^2)\frac{N-1}{N-k-1}$, where R^2 is the unadjustred R-squared, N is the number of observations, and k is the number of predictors.

[j] Root MSE - Root MSE is the standard deviation of the error term, and is equal to the square root of the Mean Squares of the Residual ($\sqrt{11.83} = 3.4389$).

- In the bottom, there is the table of regression coefficients.

Figure 5: Coefficients from Regression of *mpg* on *weight*



```
------------------------------------------------------------------------------
      [k] mpg |   [l] Coef. [m]Std. Err.    [n] t  [o]P>|t|   [p][95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |  -.0060087   .0005179   -11.60   0.000    -.0070411   -.0049763
       _cons |   39.44028   1.614003    24.44   0.000     36.22283    42.65774
------------------------------------------------------------------------------
```

[k] mpg - This column shows the dependent variable at the top (mpg) with the predictor variables below it (weight and _cons). The last variable (_cons) represents the constant or intercept.

[l] Coef. - These are the estimated coefficients from the regression equation. The regression equation can be presented in many different ways, for example $Y_{predicted} = b_0 + b_1 \times x_1$; this column of provides the estimated values of b_1 and b_0. The coefficient on weight is -0.0060087 - this tells us that for every one unit increase in weight (remember weight was measured in pounds), mpg is predicted to increase by -0.0060087 units, holding all other variables constant.

[m] Std. Err. - These are the standard errors associated with the coefficients.

[n] t - These are the t-statistics used in testing whether a given coefficient is significantly different from zero.

[o] $P > |t|$ - This column shows the 2-tailed p-values used in testing the null hypothesis that the coefficient (parameter) is 0.

[p] [95% Conf. Interval] - These are the boundary points of the 95% confidence intervals for the coefficients. The confidence intervals are related to the p-values such that the coefficient will not be statistically significantly different from zero at the 0.05 level if the confidence interval includes zero.

2.9 .do File

A .do file that corresponds to the above working example is provided on Courseworks (example1.do), and also appears in section 4 - if you have not already done so, download this file. Then open the do-file editor by going to `Window > Do-file editor > New do-file Editor` (for Mac users: Top do-file), or hitting Do-file Editor.

From the editor, open the file and review it. Note that it is good practice to provide some details about the program at the beginnning, and to annotate throughout the file.

You can execute the file from with the do-file editor window by pressing `Ctrl+D`, or from the command line by typing: `do example1.do`

3 Additional Resources for Learning STATA

This document should facilitate a basic understanding of STATA, and future recitations will introduce additional capabilities. However, the following resources can provide further help or promote a deeper understanding.

1. If you are unsure how to use a specific STATA command use the local help files. They can be obtained by typing the following command into the command line `help` *command_name* where *command_name* is the name of a STATA command, or by clicking `Help > STATA Command... > ` *`command_name`*
2. There is an introductory tutorial on the companion website for the course textbook. See `https://wps.pearsoned.com/wps/media/objects/11422/11696965/tutorials/stata_tutorial_10.pdf`
3. Oscar Torres-Reyna of the Data and Statistical Services at Princeton has written an excellent tutorial. See `http://www.princeton.edu/~otorres/STATA/`
4. The Institute for Digital Research and Education at UCLA has many useful learning modules and analytical examples. See `https://stats.idre.ucla.edu/stata/`
5. STATAlist is an active forum for STATA users of all abilities, see `https://www.statalist.org/`.
6. Finally Googling ”STATA” + [key words] will often point you in the right direction if you can choose the right key words.

4 STATA Code for Working Example

```
********************************************************************************
* Program: Recitation 1, Working Example
* Author: Mariesa Herrmann
* Created: Fall 2009
* Last Updated: Spring 2020
********************************************************************************

// Note that you can insert comments in the program for explanation
// You can enclose comments with "/*" and "*/" as shown above
// Stata will also ignore text that comes after the "//"

*Preliminaries

version 13.0 // Stata commands may differ by version, this ensures consistency
clear // Clear Memory
set mem 100m // Set Memory
set more off // Prevents Stata from pausing when the results window is filled up
capture log close // Close a log if one is open
 // More generally, ’capture’ tells stata to ignore any errors from the command

*Set directories for log files, data, and output using globals
// Here I assume you saved the ’auto.dta’ file in the following directory.
// You will need to change the following line if you saved it elsewhere.
// All output will also be sent there; alternatively, introduce a second global.
global dir "/Users/‘c(username)’/Documents/Recitation 1/"

// Open a log file. The "${logdir}" calls the filepath from the global above.
log using "${dir}example1.log", replace

********************************************************************************

/* Use Data */
use "${dir}auto.dta", clear

/* Describe Data */
desc

/* List make of first 5 observations */
sort make
```

```
list make in 1/5

/* Summarize data */
sum
sum mpg, detail
sum mpg if foreign==1
sum mpg if foreign!=1
sort foreign
by foreign: sum mpg
sum mpg if foreign==1, detail

/* Correlations and Covariances */
corr mpg weight
corr mpg weight, cov

/* Graphs */
histogram mpg // Histogram of mpg
gr export "${dir}histogram_mpg.png", replace
scatter mpg weight // Scatterplot of mpg and weight
gr export "${dir}scatterplot_mpg_weight.png", replace

/* Generate Variables */
gen weight_1000 = weight/1000
sum weight weight_1000
egen mpg_bar=mean(mpg)

/* Test Hypothesis that Domestic and Foreign Cars have same mpg */
ttest mpg, by(foreign)

/* Linear Regression */
reg mpg weight
predict mpghat // Predict the fitted values of mpg
predict ehat, resid //Predict the residuals of mpg

/* Save Changes to Data in Data called "auto2" */
save "${dir}auto2.dta", replace

/* Close the Log File */
log close

********************************************************************************
```