ECO375 Tutorial 1
Introduction to Stata

Matt Tudball

University of Toronto Mississauga

September 14, 2017

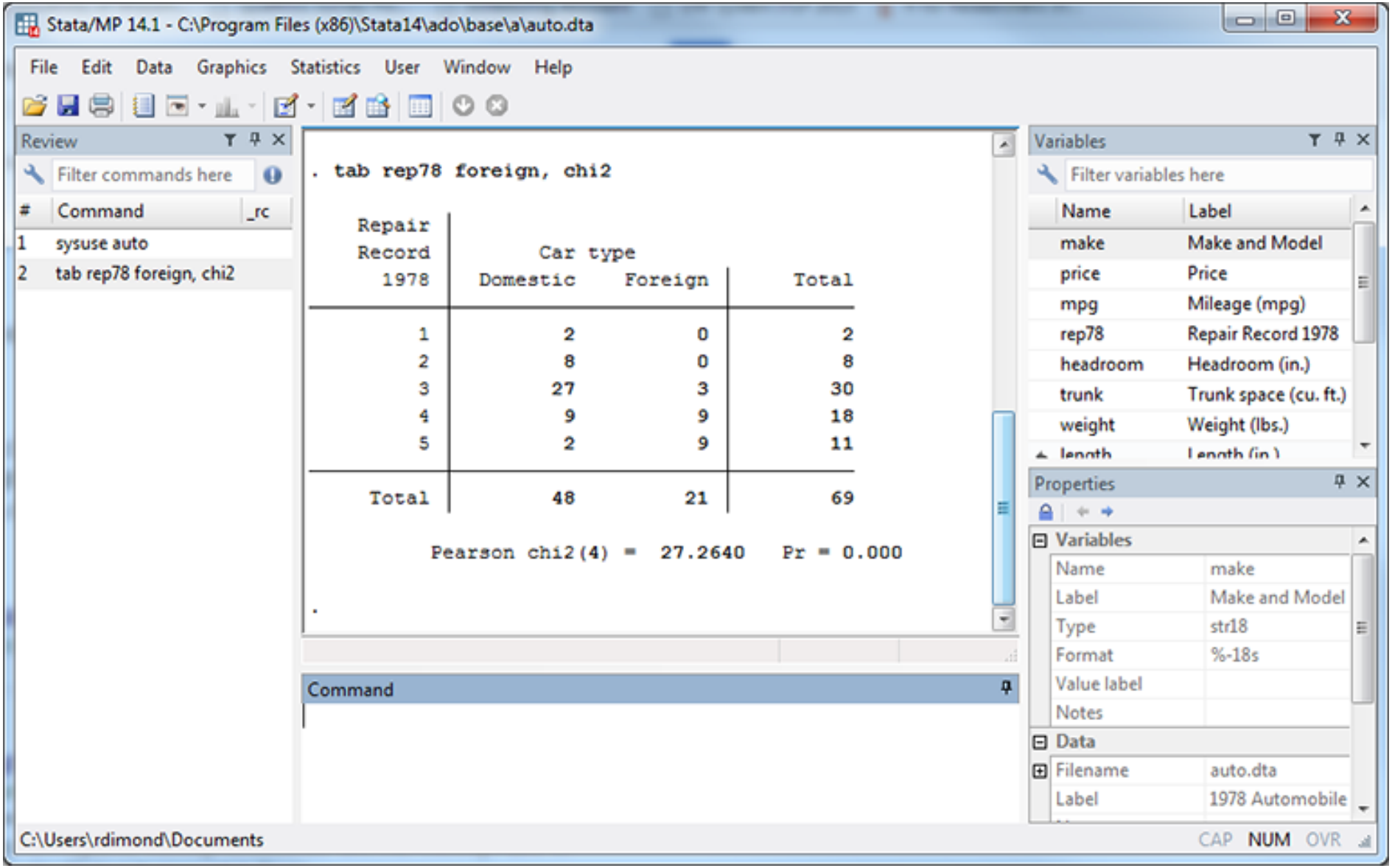
What Is Stata?

- Stata is a powerful statistical programming language available on Windows, Mac OS X and Linux.
- Stata stores the dataset to be analysed in RAM. This provides a speed advantage but it means that your computer must have enough physical RAM to load Stata and also allocate enough memory to load and perform calculations on your dataset. Make sure you check system requirements on stata.com.
- The version of Stata recommended for this class is Stata/IC. This can store a maximum of 2,048 variables and 2.14 billion observations. You will not be asked to analyse datasets larger than this in this class.

Following Along With The Slides

- The examples I use in these slides come from an in-built example dataset containing information on automobiles from 1978, including make, price, miles per galleon, weight, length, etc. It can be loaded into Stata by typing the command `sysuse auto`.
- You are encouraged to open up Stata and follow along with the examples in the slides.

Exploring The Stata Interface



Loading Data In .dta Format

- The first thing you will want to do when loading data into Stata is set up a working directory or file path.
- To set up a file path you can enter the command:
  `global path D:/Dropbox/Personal/ECO375 TA/UTM`
  This will save to memory a global macro called `path`
  containing the string `D:/Dropbox/Personal/ECO375 TA/UTM`
  To use this to load a .dta file (the file type Stata uses for datasets) you can enter the command:
  `use "$path/data.dta"`
- To set up a working directory you can enter the command:
  `cd "D:/Dropbox/Personal/ECO375 TA/UTM"`
  Once you have a working directory, you can simply type the following to load .dta files:
  `use data.dta`

Loading Data In Other Formats

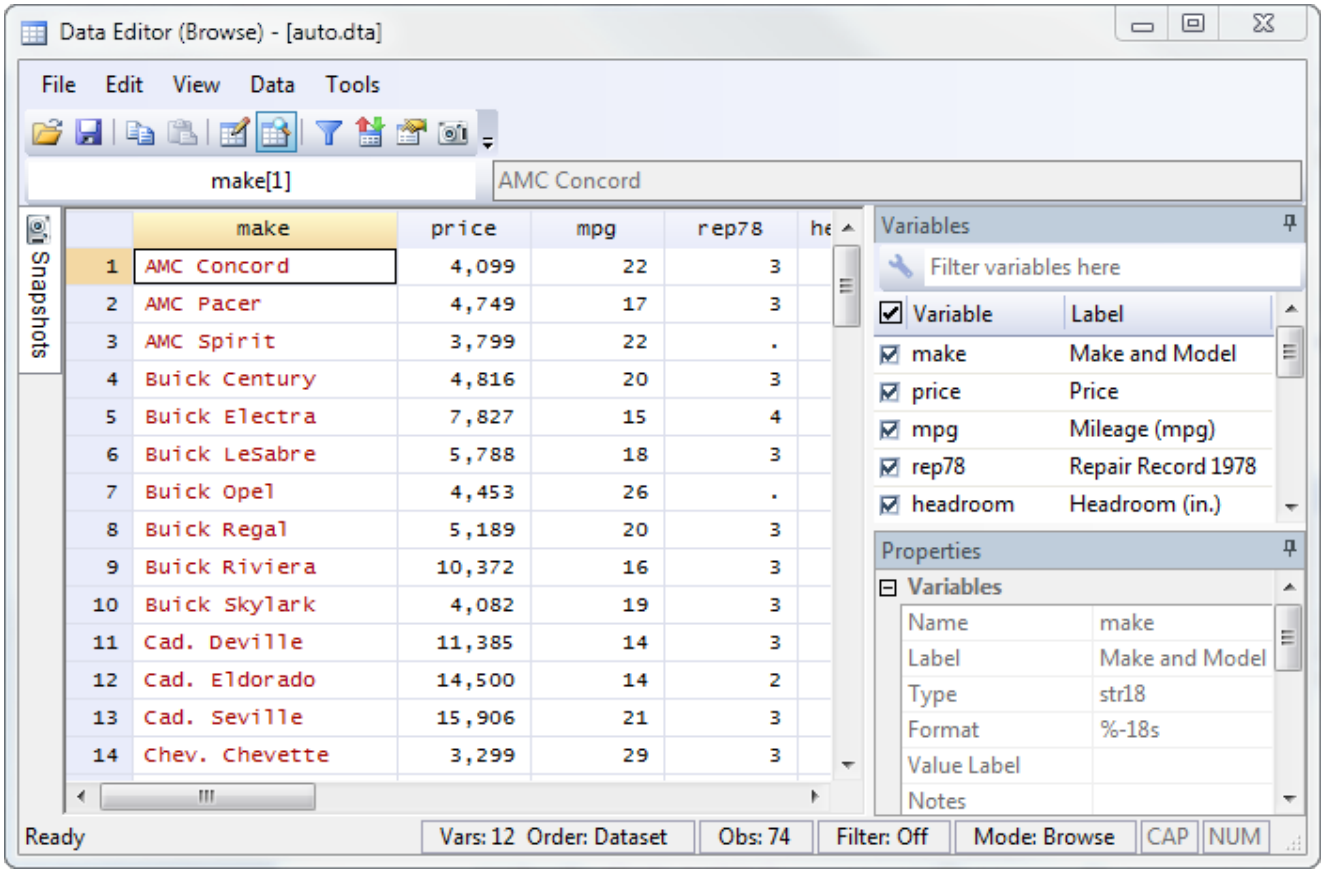
- If you have a text file in which there is one observation per line and variables are separated by some delimiter (usually comma, tab or semicolon) then you can load the file into Stata using the command `import delimited`.
- Suppose for example that you have a CSV (comma-separated values) file, then you would type:
  `import delimited "$path/data.csv", delimiters(comma)`
  The last part is redundant for this file type but it will be useful for others, such as .txt files.
- For other file types, I recommend following File $\rightarrow$ Import in the Stata GUI (Graphical User Interface). This brings up a menu allowing you to choose your file type and import it directly.

Saving Your Data

- If you have loaded your data in from another file type or made adjustments to your dataset, it is usually a good idea to save it as a .dta file. You can do this by typing the command:
  `save "$path/data.dta"`
- If you want to overwrite an existing dataset of that name, simply add to the end of the above command `, replace`

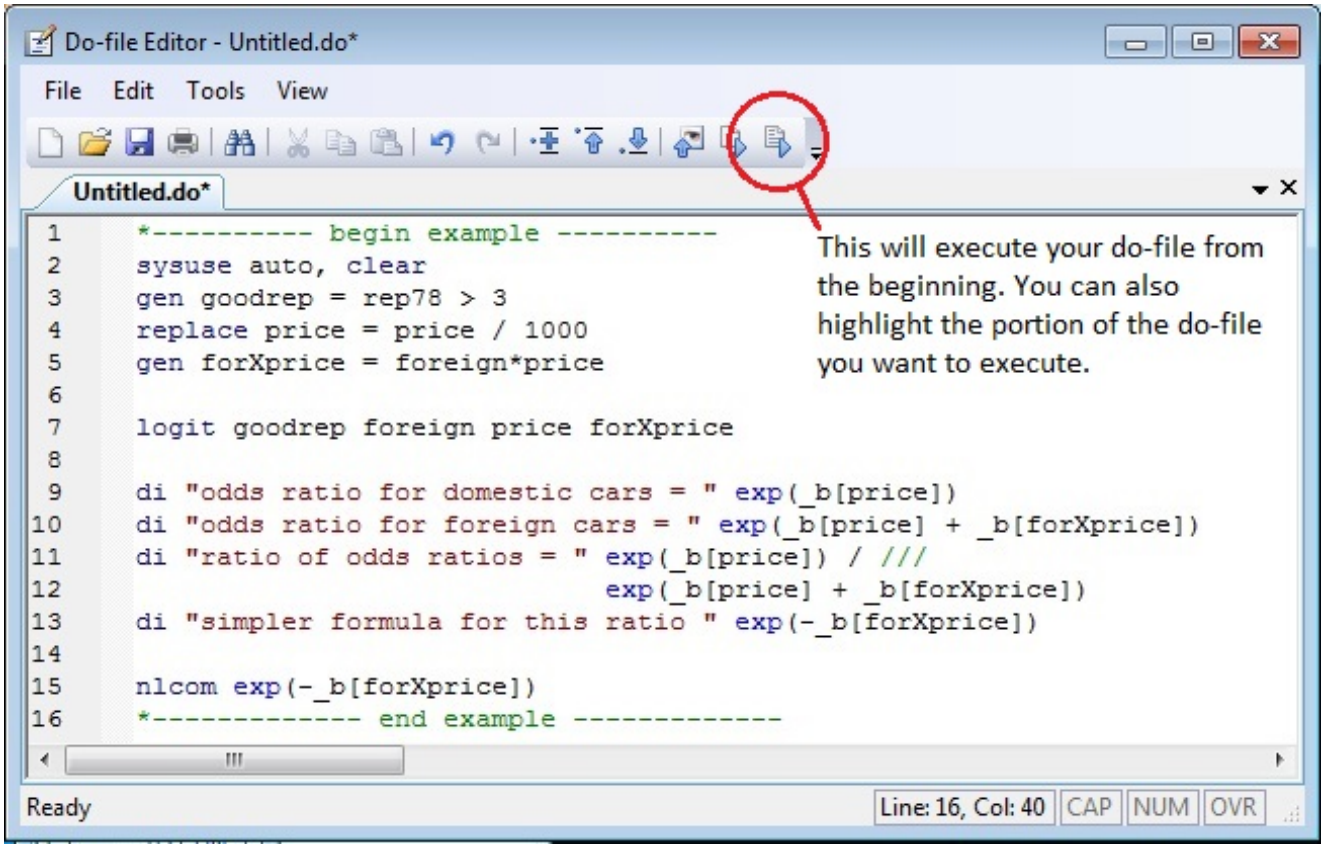
Looking At Your Data

- Once your dataset is loaded into Stata, you may want to take a look at it. You can do this by following Data → Data Editor → Data Editor (Browse) in the Stata GUI.



Setting Up A Do-File

- You can use a do-file to compose, save and execute multiple commands rather than entering them individually into the command line. You can start a new do-file by following Window → Do-file Editor → New Do-file Editor.



Commenting Your Do-File

- It is good practice to leave comments describing your code in your do-file. This will help other people, like me, understand what you are attempting to do with your code.
- If you begin a line in your do-file with * then everything in that line will be treated as a comment.
- If you want to write a comment after executing a command then you can type //.
- If you want to comment out multiple lines in your do-file then you can use /* and */. Anything written after /* and before */ will be commented out.

```
21  * This is a comment
22  summarize price // This is also a comment
23  /*
24  So is this
25  and this
26  and this
27  */
```

Creating A Log File

- A log file is a file containing everything that was sent to the Stata display window (excluding graphs which are in a separate window).
- You can launch a log file by typing the command:
  `log using "$path/log".`
  Stata will now start logging everything in your results window and will prepare to save it in a file called log.smcl.
- SMCL stands for Stata Markup and Control Language and it is the format Stata uses when saving output files.
- You can close and save your log file by typing `log close.`
- You can convert SMCL files to PDF files by using the command (all on one line):
  `translate "$path/log.smcl" "$path/log.pdf", translator(smcl2pdf).`
  This will be important when submitting assignments.

Exploring Your Data: `describe`

- With your data loaded and your do-file set up, you can now begin exploring your data. There are two commands that are very useful for this.
- The first command is `describe`.
  This will give you a quick description of the data stored in memory, including the number of observations and variables, variable names, storage type (string, integer, etc.), value labels and variable labels.

```
Contains data from C:\Program Files (x86)\Stata14\ado\base/a/auto.dta
  obs:            74                          1978 Automobile Data
 vars:            12                          13 Apr 2014 17:45
 size:         3,182                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
make            str18   %-18s                 Make and Model
price           int     %8.0gc                Price
mpg             int     %8.0g                 Mileage (mpg)
rep78           int     %8.0g                 Repair Record 1978
headroom        float   %6.1f                 Headroom (in.)
trunk           int     %8.0g                 Trunk space (cu. ft.)
weight          int     %8.0gc                Weight (lbs.)
length          int     %8.0g                 Length (in.)
turn            int     %8.0g                 Turn Circle (ft.)
displacement    int     %8.0g                 Displacement (cu. in.)
gear_ratio      float   %6.2f                 Gear Ratio
foreign         byte    %8.0g       origin    Car type
-------------------------------------------------------------------------------
Sorted by: foreign
```

Exploring Your Data: `summarize`

- The second command is `summarize`.
  For each variable this will tell you the number of non-missing observations, mean, standard deviation and min and max values.
- For both of the above commands, you can also limit them to a single variable or group of variables. For example, you can type:
  `summarize price`.
  This will only provide a summary of the variable `price`.

Variable	Obs	Mean	Std. Dev.	Min	Max
price	74	6165.257	2949.496	3291	15906

- You can do the same with `describe`.

variable name	storage type	display format	value label	variable label
price	int	%8.0gc		Price

Exploring Your Data: `tabulate`

- For discrete variables (i.e. variables which can only take on a finite number of values), you may want to create frequency tables describing the frequency with which each value of that variable occurs in the dataset. You can do this with the command `tabulate`.
- Suppose you want to see the most common repair record in our dataset, you can type `tabulate rep78`.

Repair Record 1978	Freq.	Percent	Cum.
1	2	2.90	2.90
2	8	11.59	14.49
3	30	43.48	57.97
4	18	26.09	84.06
5	11	15.94	100.00
Total	69	100.00	

- You can produce a two way frequency table with the command `tabulate [variable 1] [variable 2]`.

Searching For Commands

- Sometimes you may want to perform an operation in Stata but you don't know the associated command. Stata has an in-built search tool to help with this which you can access using the command `search`.
- Suppose you want to figure out how to describe your data but you don't know the command `describe`.
  You can type `search data description`.
  This will bring up a list of commands most related to your keywords 'data description'.

```
search for data description                          (manual:  [R] search)

Search of official help files, FAQs, Examples, SJs, and STBs

[D]     codebook . . . . . . . . . . . . . . . . . . . . Describe data contents
        (help codebook)

[D]     describe . . . . . . . . . . . . . . . . Describe data in memory or in file
        (help describe)

[D]     icd . . . . . . . . . . . . . . . . . . . . . . Introduction to ICD commands
        (help icd)
```

Help With Commands

- You may want a description of what a certain command does. Stata also has an in-built feature for this which can be accessed using the command `help`.
- Suppose you want to to know what operation is performed using the command `summarize`.
  You can type `help summarize`.
  This will bring up a description of the command, including syntax, options, examples and stored results.

Title

[R] summarize — Summary statistics

Syntax

summarize [varlist] [if] [in] [weight] [, options]

options	Description
Main	
detail	display additional statistics
meanonly	suppress the display; calculate only the mean; programmer's option
format	use variable's display format
separator(#)	draw separator line after every # variables; default is separator(5)
display_options	control spacing, line width, and base and empty cells

In And If Statements And `sort`

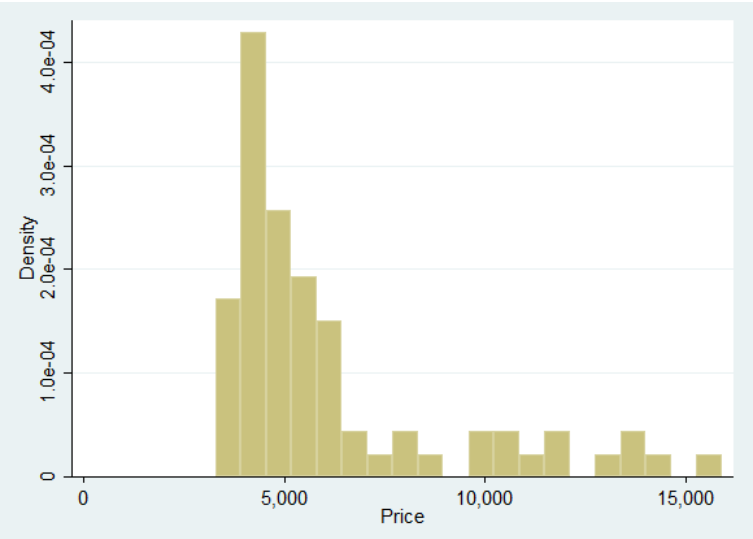
- You can perform operations only on observations that meet a certain criteria using the command `if`.
- Suppose you want to summarise the price of vehicles which do 20 or more miles per gallon, then you would type the command `summarize price if mpg >= 20`.
- You can perform operations over a pre-defined range of observations using the command `in`.
- Suppose you want to summarize the price of the cheapest 20 vehicles in the dataset. Then you would type the following commands into your do-file:

```
sort price
summarize price in 1/20
```

- As seen above, if you want to sort a numeric variable from smallest to largest (or a string variable alphabetically) then you can use the command `sort`.

Data Visualisation: `histogram`

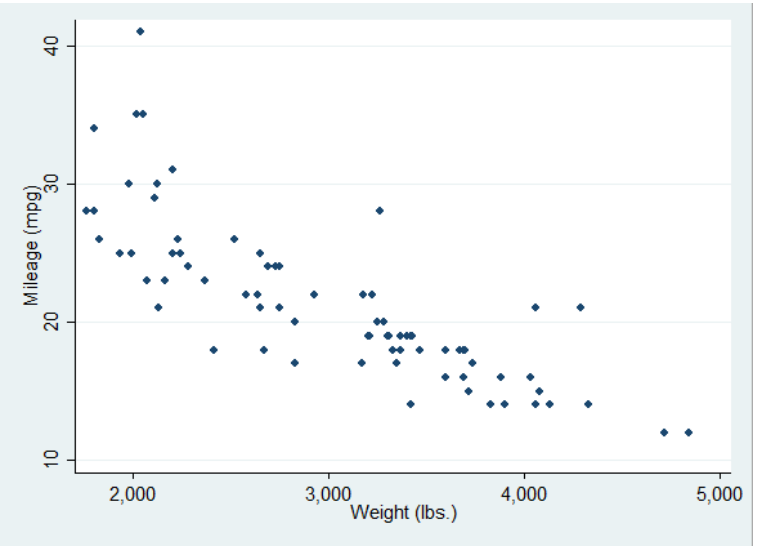
- A histogram is a useful way of visualising the distribution of a given variable over the values that it takes. The command to produce a histogram in Stata is `histogram [variable name]`.
- You can select the number of bins (i.e. the number of rectangles) by adding to the end of the above command `, bin(#)`.
- The histogram below was produced using the command:
  `histogram price, bin(20)`



Data Visualisation: `scatter`

- A scatter plot is a useful way of visualising the covariance between two variables. The command to produce a scatter plot in Stata is `scatter [variable name 1] [variable name 2].`
- For example, we might think that there is a relationship between the weight of a vehicle and its gas mileage. We can produce a scatter plot by typing:

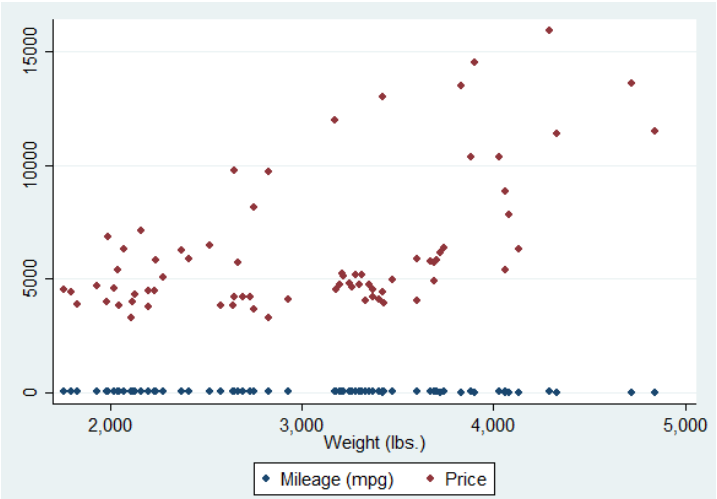
  `scatter mpg weight`



Overlaying Multiple Figures: `twoway`

- Stata classifies all plots which show the relationship between two variables as twoway plots.
- One of the useful features of twoway plots is that multiple twoway plots can be displayed on the same figure.
- Suppose we want to display a scatter plot of miles per galleon against weight and price against weight on the same graph, then we would type the following:

```
twoway (scatter mpg weight) (scatter price weight)
```



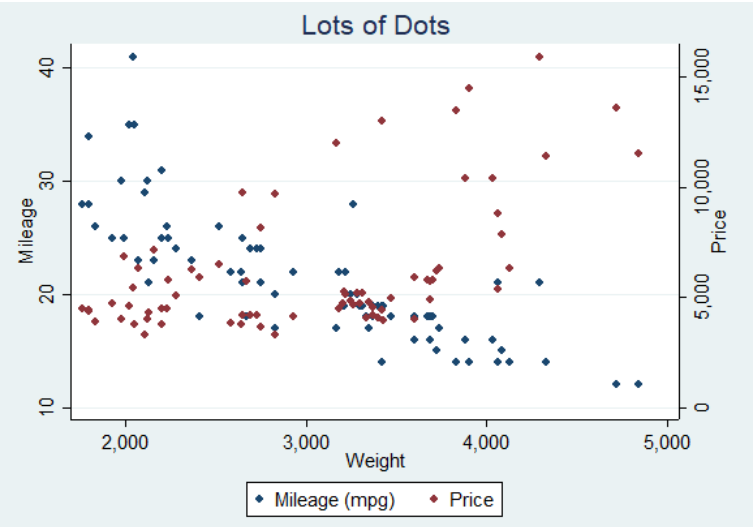
Editing And Cleaning Figures

- Notice that the figure in the previous slide doesn't look very good. It would look much better if mileage were displayed on another axis. We can implement this in Stata by typing:

```
twoway (scatter mpg weight, yaxis(1)) (scatter price
weight, yaxis(2))
```

- We can also give our graph a custom title and custom labels for the axes by adding to the above code:

```
, title("Lots of Dots") xtitle("Weight")
ytitle("Mileage", axis(1)) ytitle("Price", axis(2))
```



Saving Figures

- To save a figure as a Stata .gph file, add to the end of the command generating the figure `, saving("$path/graph.gph").`
- Ex. `scatter mpg weight, saving("$path/graph.gph")`
- You may want to export your figures as a different file type, such as PNG or PDF. This can be done using the command `export`.
- For example, to export as a PDF you can type:

```
scatter mpg weight
graph export "$path/graph.pdf"
```

Generating New Variables

- We often want to generate new variables from our existing variables. This can be done using the command `generate`.
  This command will perform calculations observation-by-observation, including addition, subtraction, division, multiplication, etc.
- Suppose we are interested in the price per pound of a vehicle. We could type: `generate price_per_lb = price/weight`.
  This would generate a new variable in our dataset consisting of the price of each vehicle divided by its weight in pounds.
- Since calculations are performed observation-by-observation, if you put a constant value on the right-hand side, Stata will create a new variable in which all observations take that value.
- You should also explore the command `egen`.
  This stands for 'extensions to generate' and contains many useful in-built functions, such as mean, median and row sum.

Dropping And Replacing Variables And Observations

- We can drop variables and observations using the command `drop`.
- To drop the price and weight variables, we can simply type `drop price weight`.
- To drop observations that meet a certain criteria (for example, miles per galleon less than 20) we can type `drop if mpg <= 20`.
- To drop the first 20 observations, we can type `drop in 1/20`.
- The reverse of this command is `keep`.
  It has the same syntax as `drop`.
- If we want to replace a variable with some function of our existing variables, we can use the command `replace`.
- For example, I may want to represent the length variable in terms of feet rather than inches. Then I would need to type:
  `replace length = length/12`.

In-Class Exercise

- We will now do an in-class exercise. You may work in groups of 2 or 3.
- Download the .csv file titled realestate from my website (matthewtudball.com). This is a dataset of real estate transactions in Sacramento. You will need to load that dataset into Stata and do the following:
  - First drop all observations in which the size of the house (in square feet) is less than 1000.
  - Now find the average price of houses with 2 bedrooms.
  - Find the average price of the 100 cheapest houses.
  - Find the maximum price-per-bedroom.
  - Produce and save a scatter plot titled "Price and Square Feet" with appropriately labelled axes showing the relationship between the price of a house and its size in square feet.