Stata How-to:
Basic Graphs

Patrick Blanchenay

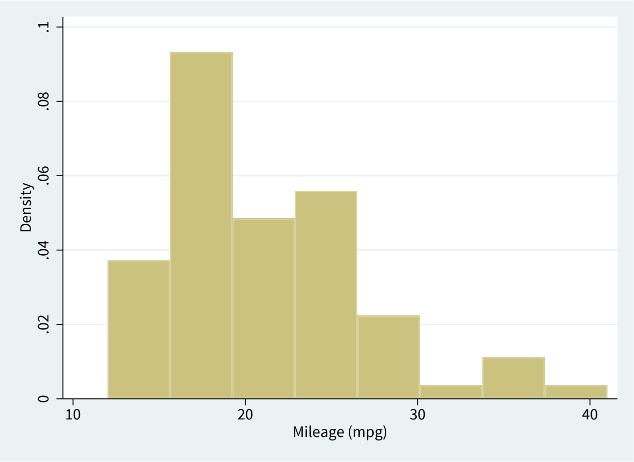
2021-12-27

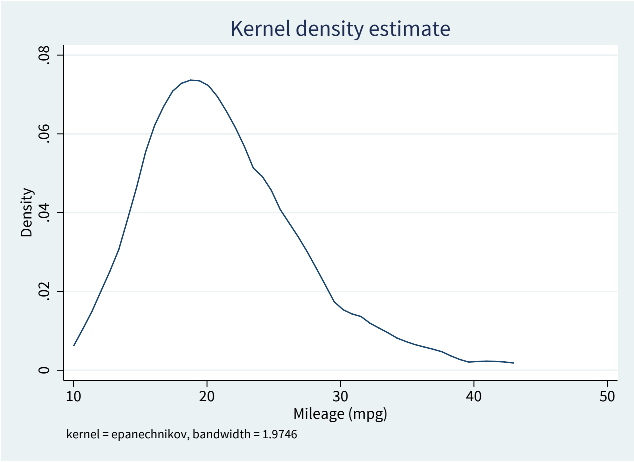
Contents

1. Univariate Graphs

The **histogram** command can be used to produce a histogram of a variable's values (graph on the left); to plot an estimate of the density distribution of variable, use **kdensity** (right). For example, using the Stata-provided `auto.dta` dataset:

```
sysuse auto
histogram mpg
kdensity mpg
```





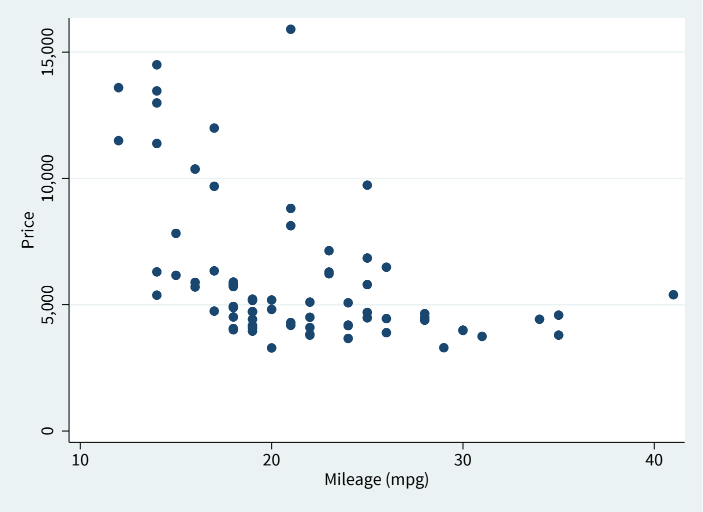
Type **help histogram** and **help kdensity** for more details and options, in particular on how to choose the width of the bars on the histogram.

2. Bivariate graphs

2.1. Scatter plots

Stata also has a large number of “twoway" plots, including scatterplots, line graphs, graphs of functions, lines of best fit, etc. We can create a scatterplot using the **`twoway scatter`** command:

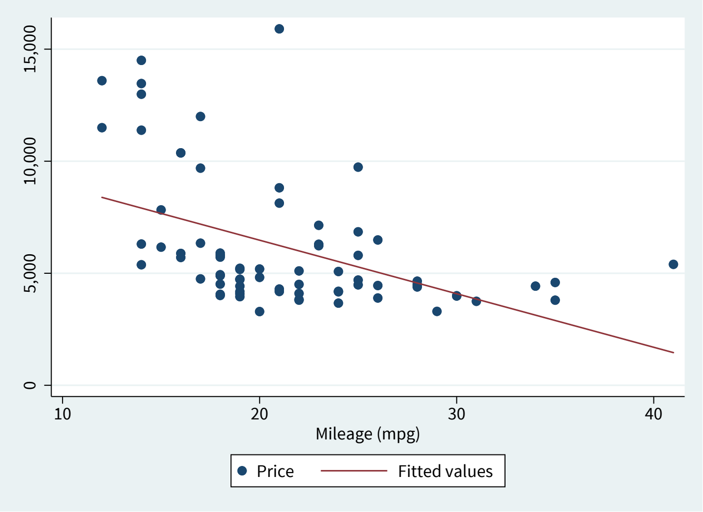
```
twoway scatter price mpg
```



In the command, note that the first variable listed is the variable from the *vertical* axis, followed by the variable on the *horizontal* axis.

We can also combine the scatter plot with an OLS line of best fit using the **`lfit`** command. We do this by enclosing both the **`scatter`** and **`lfit`** subcommands in brackets:

```
twoway (scatter price mpg) (lfit price mpg)
```



Type **`help twoway`** for more details and options.

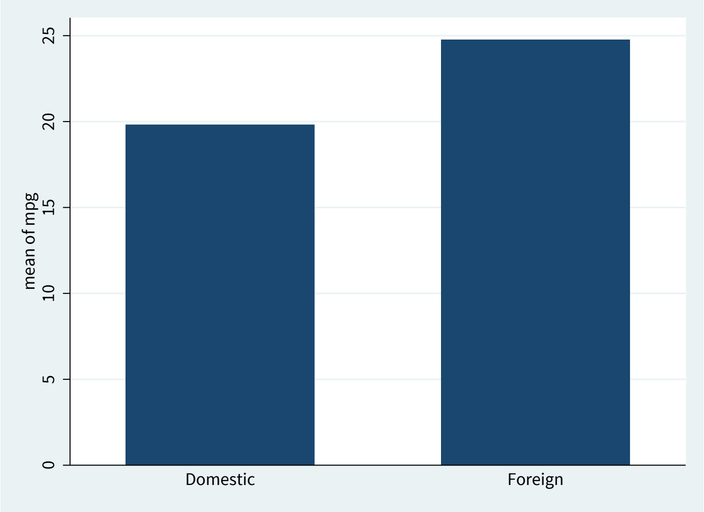
2.2. Bar charts

Bar charts are commonly used to compare, among other things, average values of a variable in different groups. For instance, if you have a treatment and control group, it is useful to visually plot the outcome, or some other characteristics, in the treatment versus control. The general syntax to do that is

```
graph bar (mean) numeric_var, over(cat_var)
```

where you are trying to compare the value of the variable **numeric_var** in groups defined by the variable **cat_var** (which must be categorical variable). For instance, in the Stata-provided auto.dta dataset, to compare fuel consumption between US and foreign cars, one would run:

```
graph bar (mean) mpg, over(foreign)
```



Type **help graph bar** for more details and options.

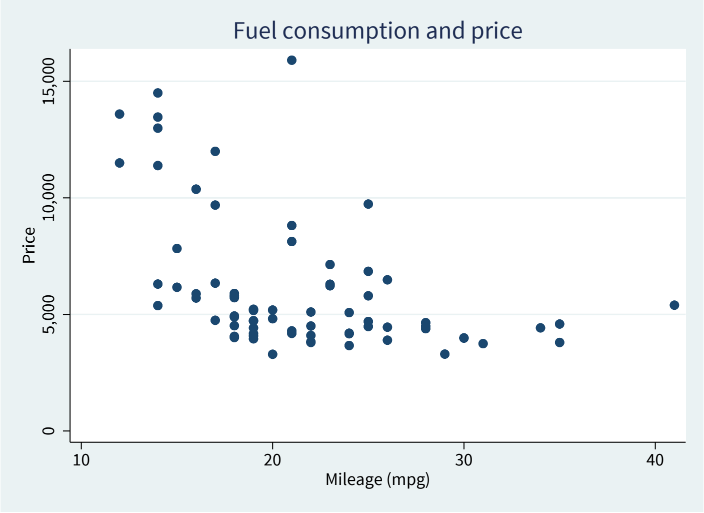
3. Well-labelled graphs

All graphs must be accurate and easy to understand. Graph commands can take numerous options.[1] I only detail here the most important; at the minimum, a title and axis titles should be provided, and the units should be specified.

Add a title using the title option:

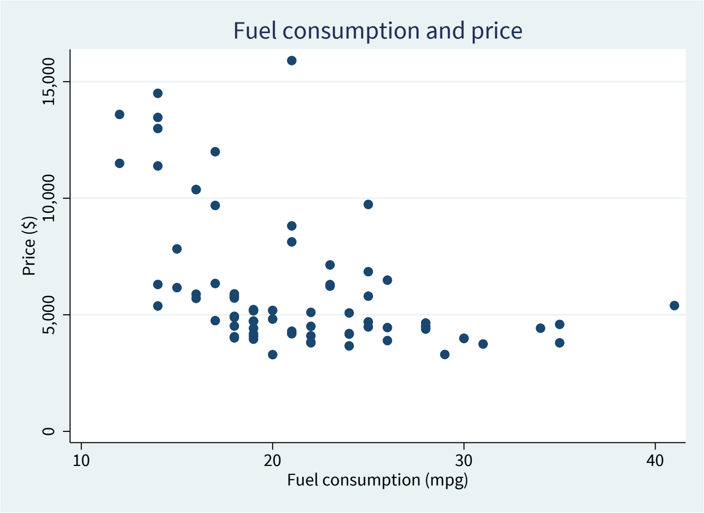
```
twoway (scatter price mpg), title("Fuel consumption and price")
```

[1] It is often easier to split them on several lines using triple slashes ///.



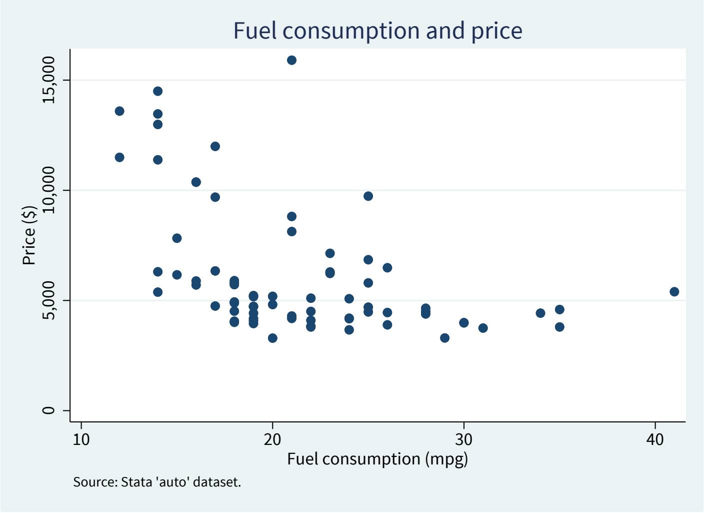
By default, axes titles are set to the corresponding variable label (if defined) or the variable name if not. These can often be profitably overriden by using the **xtitle()** and **ytitle()** options:

```
twoway (scatter price mpg), title("Fuel consumption and price") ///
        xtitle("Fuel consumption in miles-per-gallon") ytitle("Price ($)")
```



Notes, such as sources or explanations, can also be included at the bottom of the graph using the **note()** option:

```
twoway (scatter price mpg), title("Fuel consumption and price") ///
        xtitle("Fuel consumption in miles-per-gallon") ytitle("Price ($)") ///
        note("Source: Stata 'auto' dataset.")
```



4. Saving graphs to use in other applications

Graph commands open a new window and generate a Graph in the Stata graph format `.gph` that most applications cannot use directly. We can save the graph as a PNG file by following up the previous command with the **`graph export`** command:

```
graph export "mpg histogram.png", replace
```

Warning

By default, PNG graphs are saved at the size of the window Stata opened. This can be inconsistent from computer to computer. If you want to manually specify the dimension of the PNG, use the **`width()`** option:

```
graph export "mpg histogram.png", replace width(1280) // the image will be 1280 pixels wide
```

PNG format is useful for quick inclusion in Word/Powerpoint. If you want a scalable format, you can export to PDF or SVG:

```
graph export "mpg histogram.svg", replace
```

Warning

Unless you specify a file path, graphs are exported into your working directory. If you want to export to a subfolder called `graphs`, you must first create that subfolder, and then specify it in the **`graph export`** command:

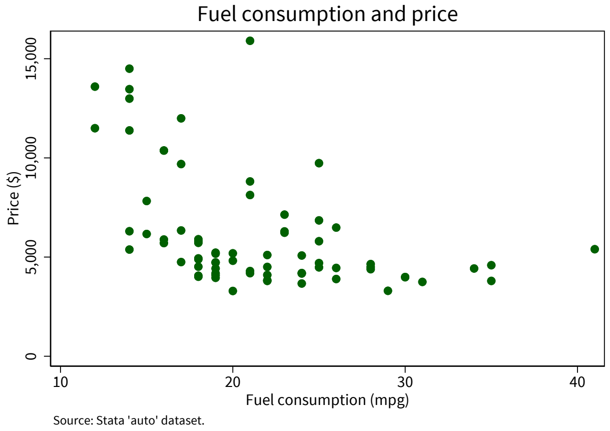
```
graph export "graphs/mpg histogram.pdf", replace // export to graphs subfolder
```

5. Using a scheme to change the graph appearance

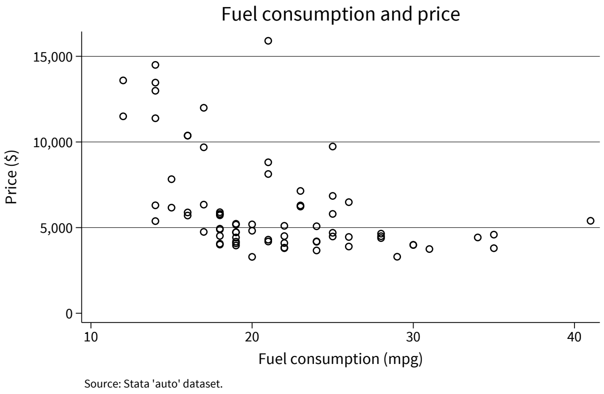
Stata’s default graph appearance is not appealing.[2] Graph schemes can be easily used to modify the overall appearance of the graph. Setting a graph schemes is done with **set scheme**, and should occur only once in the do-file, before any other graphing command:

```
set scheme s1color
set scheme lean2
```

Some schemes are provided in Stata; I recommend `s1color` (left) and`lean2` (right).



Source: Stata 'auto' dataset.



Source: Stata 'auto' dataset.

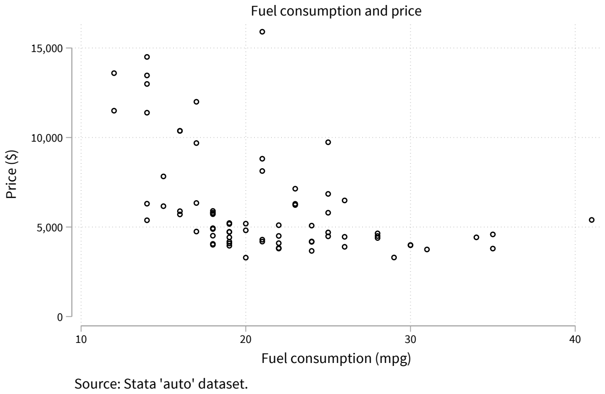
You can also find some online. I recommend schemes by Daniel Bischof; which you can install by:

```
  ssc install blindschemes, replace all
```

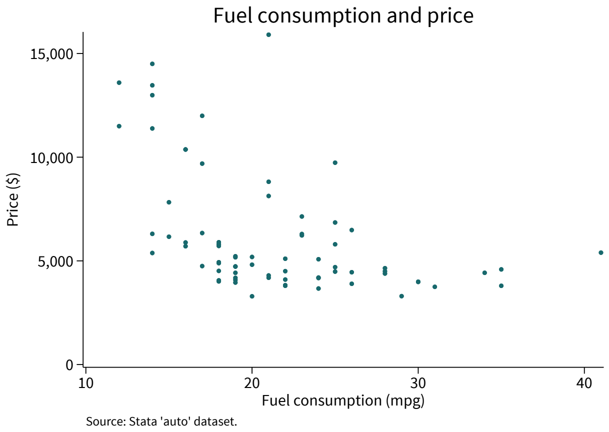
You only need to install them once. Then at the beginning of your do-file, tell Stata to use the scheme `plotplainblind` by inserting:

```
  set scheme plotplainblind
```

Your graphs will have a more modern appearance (left). You can also check Gray Kimbrough’s uncluttered scheme (right):



Source: Stata 'auto' dataset.



Source: Stata 'auto' dataset.

[2] Unless you like early 1990s nostalgia.