

Exploring Data in Stata Using Plots

Graphing Basics

- Visually inspecting data is always an important first step to data analysis
- Stata allows users to quickly and easily create scatter, line, and bar graphs to explore their datasets
- The main command for graphing is `graph`, followed by a keyword that identifies the type of graph:

```
graph twoway scatter vary varx
```

```
graph box cont_var, over(cat_var)
```

```
graph twoway line vary varx
```

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

Graphing Window

- When the user runs a **graph** command, the Graph Window will automatically pop up
- Stata has a built-in graph editor that is fairly customizable. It can be useful for touching up a graph for publication, but nearly all of the options available there are also available through **graph** command options
- If you close the Graph Window, you can always return to the last created graph by typing a blank **graph** command into the Command Window

graph twoway scatter

```
graph twoway scatter vary varx, title("Title Text")
```

- This command creates a scatterplot with *vary* on the y-axis and *varx* on the x-axis
- The title option adds "Title Text" as a title
- There are many other options, such as *msymbol*, *mcolor*, or *msize* which control the type of plot symbol, the color of symbols, and their size
- Let's try graphing mpg vs. price of automobiles

graph box

```
graph box cont_var, over(cat_var) title("Text")
```

- This command creates box plots of a continuous variable (cont_var) separated by a categorical variable (cat_var)
- Stata will allow you to create group box plots over continuous variables
- We can take a look at mpg across domestic and foreign cars as an example
- There are many other options, which are best explored by calling **help** graph box

graph twoway line

```
graph twoway line vary varx, lpattern(dash)
```

- This command creates a line graph (with a dashed line)
- Data need to be set up properly, but any number of y-variables can be plotted on the same scale by repeating
- We can take a look at `le` over `time` using the `uslifeexp` dataset
- Twoway graph types can be easily layered using the `||` device:

```
graph twoway line vary varx || twoway scatter vary varx
```

graph bar

graph bar (mean) cont_var, over(cat_var)

- This command creates a bar graph of a continuous variable over categorical groups using the mean statistical function
- The user can specify a different statistical function or use `asis` to graph the value of a variable
- The `cont_var` is optional, without it Stata produces a bar graph of percentage of observations in each `cat_var`

graph bar, over(foreign)

graph name

```
sysuse educ99gdp
```

```
graph bar (asis) public, over(country) name(public_graph)
```

```
graph bar (asis) public private, over(country) name(comparison_graph)
```

- The name option will save your graph to Stata's working memory (not as a file someplace on your hard drive)
- To re-display a previously named graph, use **display**
graph graph_name

graph combine

```
graph combine graph1 graph2 ...
```

```
graph combine public_graph comparison_graph
```

- The combine command takes multiple graph names and combines them into one graph
- There are several options for arranging your combined graphs, notably `rows(n)` and `columns(n)` which determine how many rows and/or columns of graphs the combined graph will have

graph save

graph save [graphname] filename

- The graph save command will save the current graph or the graph graphname as a file to disk
- Stata graphs are saved as .gph files, which are Stata-specific file types. When loaded, these files will look different based on individual user settings
- If you want a graph to look *exactly* the way it does on your screen, you should save using the option `asis`

graph export

graph export filename.suffix

- This command exports the current graph as a specific image file (using the appropriate file extension `.suffix`)
- Each file type has a specific set of options (often tied to size of the saved image and their resolution)

graph display comparison_graph

graph export comparison_graph.tif

Exercises (1)

1. Movie Metadata

- A. Create a scatter plot of gross against budget. ***It doesn't look great!*** Scour the graph help files to find out how to use the log scale for both x and y axes, and name this graph `gross_budget_log`.
- B. Create a box plot which compares the imdb scores of movies across the expensive categorical variable. Name it `imdb_expensive`.
- C. Create a bar graph that displays the percentage of movies over the categorical country code variable. Name it `movies_country`.
- D. Combine the three graphs above into one graph. Make one version with one column, make another with one row. Save these graphs as `movies_col.gph` and `movies_row.gph`!

Exercises (2)

1. Auto Data

- A. Write a loop in `auto.do` which produces separate scatter plots for price vs mileage, price vs headroom, and price vs trunk space. Make sure each one has a title! Save each graph as `price_othervariable.gph`.
- B. Create box plots of mileage and headroom across our price category variable. To make things more complex, use the `by` option to create separate graphs across the foreign variable. Give them names.
- C. Combine the graphs from part B and export this combined graph as a `.tif` file named `auto_boxplots.tif`