Stata Brush Up Course

Session 3

Sergi Quintana

IDEA

September 4, 2023

Plan for the session

We will see:

- Stata graphics

For reference visit the Stata graph manual here.

Syntaxt

To create a Stata graph we will most likely use the graph command followed by twoway. What distinguishes a twoway graph is that it fits onto numeric y and x axes. Twoway are a family of plots, all of them can be found here. We will analyze six families:

- Scatter plots
- Fit plots
- Line plots
- Area plots
- Range plots
- Distribution plots

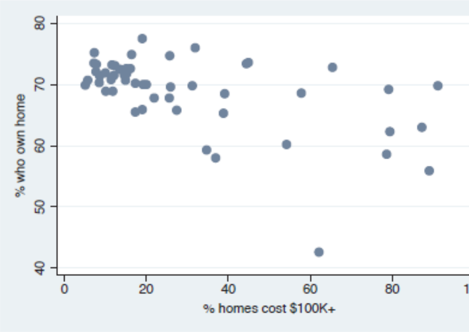
The type of plot will depend on the variable that we want to visualize.

Scatter plot

```
graph twoway scatter ownhome propval100
```

Here is a basic scatterplot. Note that this command starts with `graph twoway`, which indicates that this is a `twoway` graph. `scatter` indicates that we are creating a twoway scatterplot. These are followed by the variable to be placed on the y-axis and then the variable for the x-axis.

Uses allstates.dta & scheme vg_s2c

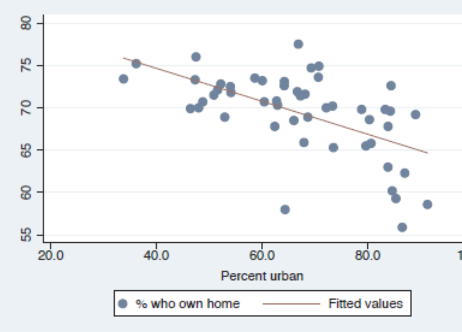


Fitted Line

```
twoway (scatter ownhome pcturban80) (lfit ownhome pcturban80)
```

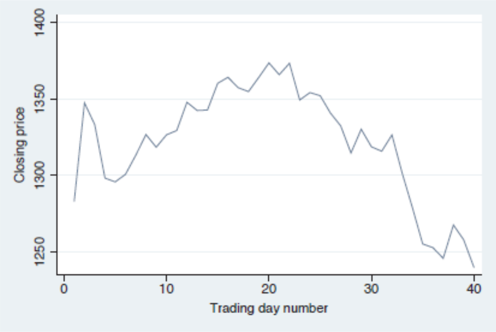
Here, we show a scatterplot of ownhome by pcturban80. In addition, we overlay a linear fit lfit predicting ownhome from pcturban80. See Twoway : Overlaying (87) if you would like more information about overlaying twoway graphs.

Uses allstatesdc.dta & scheme vg_s2c



Line plot

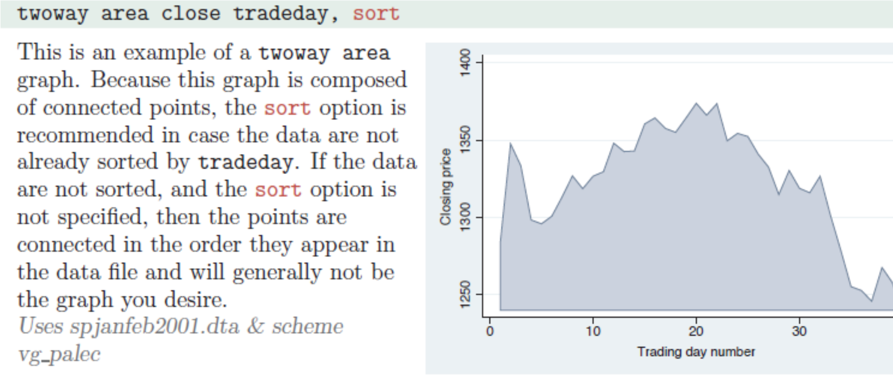
```
twoway line close tradeday, sort
```



Here, we show an example using `twoway line` showing the closing pri across trading days. Note the inclusi of the `sort` option, which is recommended when you have points connected in a Stata graph.

*Uses spjanfeb2001.dta & scheme vg_*

Area plot



```
twoway area close tradeday, sort
```

This is an example of a `twoway area` graph. Because this graph is composed of connected points, the `sort` option is recommended in case the data are not already sorted by `tradeday`. If the data are not sorted, and the `sort` option is not specified, then the points are connected in the order they appear in the data file and will generally not be the graph you desire.

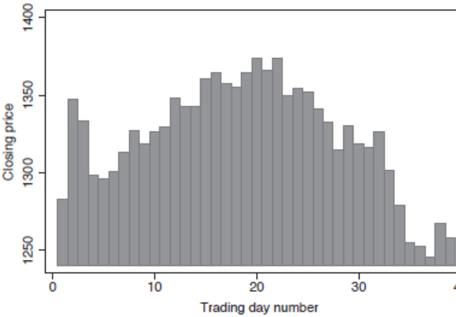
Uses spjanfeb2001.dta & scheme vg_palec

Bar Plots

```
twoway bar close tradeday
```

Consider this bar chart, which shows the closing prices of the S&P 500 broken down by the trading day of the year.

Uses spjanfeb2001.dta & scheme vg_s1m

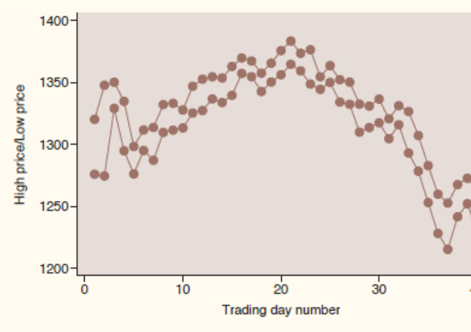


Range Plots

```
twoway rconnected high low tradeday, sort
```

The rconnected (range connected) graph shows the high and low prices by `tradeday`, the number of days stocks have been traded in the year. The `rconnected` plot shows a separate line for the high and low prices, and a marker appears for each x-value. The `sort` option is recommended because the points are connected by lines and is needed if the data were not already sorted on `tradeday`.

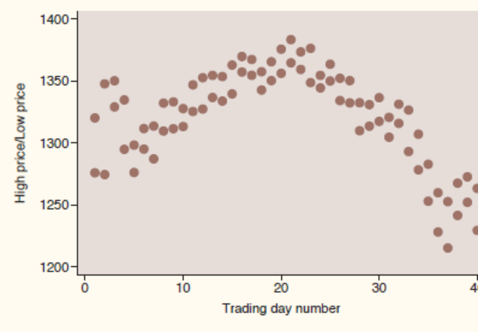
Uses spjanfeb2001.dta & scheme vg_rose



Range Plots

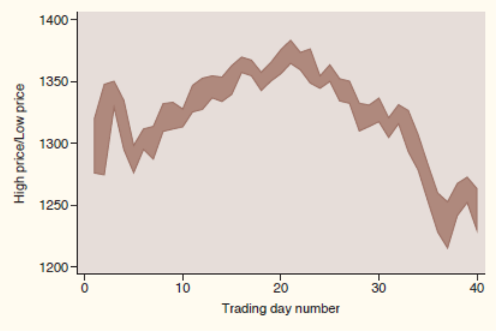
```
twoway rscatter high low tradeday
```

The `rscatter` graph is similar to the `rconnected` graph, except that lines connecting the symbols are not plotted. *Uses spjanfeb2001.dta & scheme vg_rose*



Range Plots

```
twoway rarea high low tradeday, sort
```

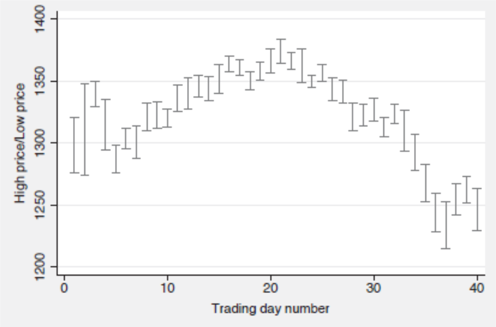


The rarea graph is similar to the rline graph, except that you can control the fill color of the area betwe the high and low values.

Uses spjanfeb2001.dta & scheme vg_rose

Range Plots

```
twoway rcap high low tradeday
```



The rcap graph shows a spike rangin from the low to high values and puts cap at the top and bottom of each spike.

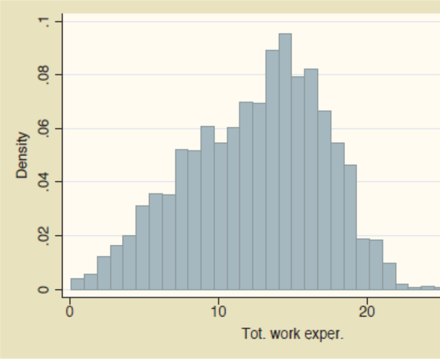
Uses spjanfeb2001.dta & scheme vg_s2m

Distribution Plots

```
twoway histogram ttl_exp
```

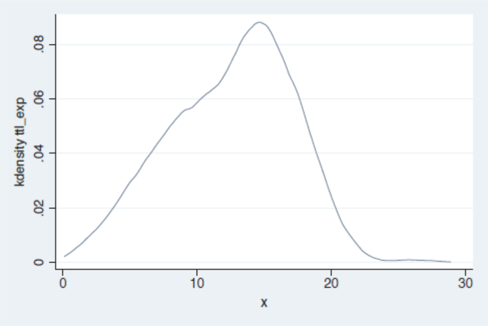
We begin by showing a histogram of the variable total work experience. Note that, unlike many other twoway plots, this command takes only one variable that is graphed on the x-axis. The y-axis represents the density, such that the sum of the areas of the bars equals 1. If you are not going to combine this graph with other twoway graphs, the **histogram** command may be preferable to **twoway histogram**.

Uses nlsw.dta & scheme vg_past



Distribution Plots

```
twoway kdensity ttl_exp
```

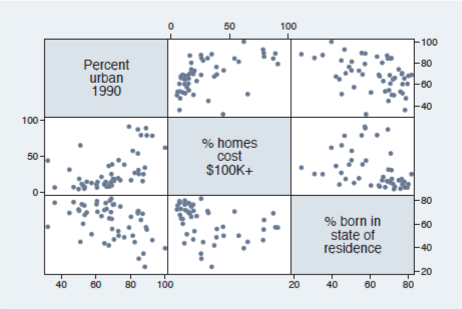


Here is a kernel-density plot of total work experience. We could have added the **horizontal** option to display the graph as a horizontal plot, but this option is not shown.

Uses nlsw.dta & scheme vg_s2c

Graph - Matrix

```
graph matrix urban propval100 borninstate
```



Let's look at a scatterplot matrix of three variables: **urban**, **propval100**, and **borninstate**.

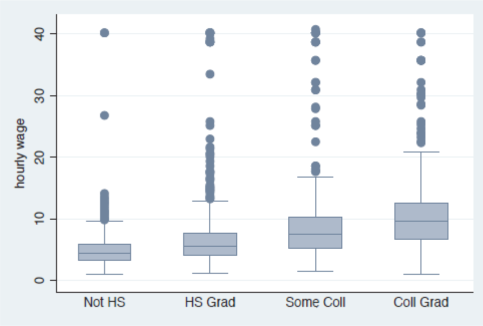
Uses allstates.dta & scheme vg_s2c

Box plot

```
graph box wage, over(grade4)
```

This is a box plot of wages broken down by education. The `over(grade4)` option breaks down wages by education level (in four categories). By default, the separate levels of **grade4** are graphed using the same color, and the levels are labeled on the x-axis. The graph shows a large number of outside values that are displayed as markers beyond the whiskers. The following example shows how we can suppress the display of the outside values.

Uses nlsw.dta & scheme vg_s2c

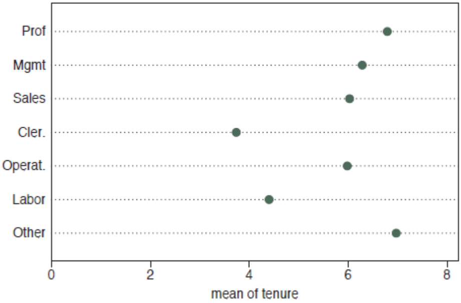


Dot Graph

```
graph dot tenure, over(occ7)
```

Here, we use the over() option to show the average current work experience broken down by occupation. By default, the y-variable (tenure) is placed on the bottom axis and is considered to be the y-axis. Likewise, the levels of occ7 are placed on the left axis and are considered to form the x-axis, or categorical axis.

Uses nlsw.dta & scheme vg_s1c

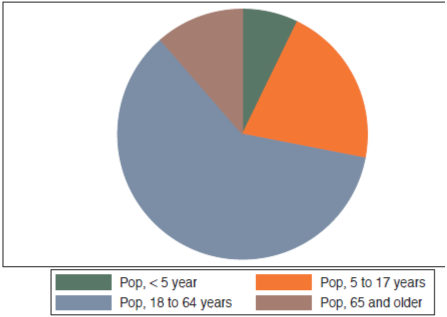


Pie Graph

```
graph pie poplt5 pop5_17 pop18_64 pop65p
```

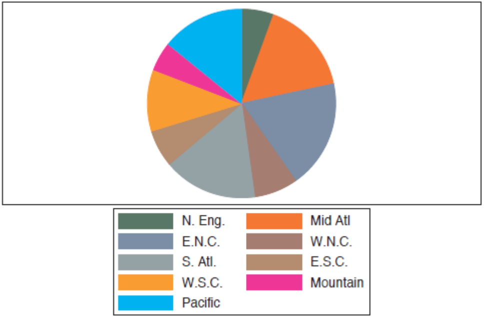
In this syntax, you supply multiple *y*-variables, and each *y*-variable corresponds to a slice in the pie. The first *y*-variable is the population in the state that is younger than 5 years old, the next the population 5 to 17 years old, the next 18 to 64 years old, and the last 65 years and older. The entire pie would correspond to the sum of all of these variables across all states. The first slice then corresponds to the percentage of the total population that is younger than 5 years old.

Uses allstates.dta & scheme vg_s1c



Pie Graph

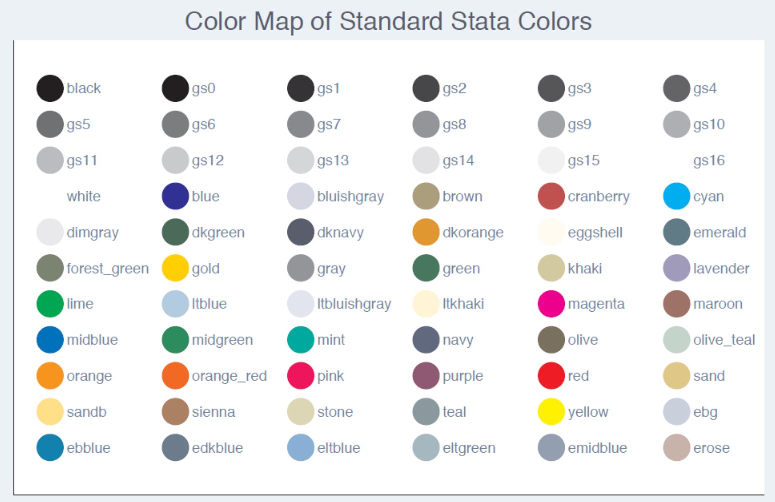
```
graph pie pop, over(division)
```



In this syntax, you supply a single *y*-variable and an `over()` option. In this case, the *y*-variable corresponds to the population of the state, the entire pie corresponds to the entire population, and each slice corresponds to the percentage of the population for each level of `division`.

Uses allstates.dta & scheme vg_s1c

Color Map


Color Map of Standard Stata Colors
black
gs0
gs1
gs2
gs3
gs4
gs5
gs6
gs7
gs8
gs9
gs10
gs11
gs12
gs13
gs14
gs15
gs16
white
blue
bluishgray
brown
cranberry
cyan
dimgray
dkgreen
dknavy
dkorange
eggshell
emerald
forest_green
gold
gray
green
khaki
lavender
lime
ltblue
ltbluishgray
ltkhaki
magenta
maroon
midblue
midgreen
mint
navy
olive
olive_teal
orange
orange_red
pink
purple
red
sand
sandb
sienna
stone
teal
yellow
ebg
ebblue
edkblue
eltblue
eltgreen
emidblue
erose

Options

There are many options that can be edited. Most common ones are:

- Title, legends and captions.
- Axis scales, lables and titles.
- Line styles and width.

More advanced options are:

- Label points or in-graph text.
- Combine graphs.

Options

Title, subtitles

- title(”title here”)
- subtitle(”subtitle here”)

Axis title

- xtitle(”title here”)
- ytitle(”title here”)

Axis scale and label

- xscale(range for axis here).
- xlabel: control the placement and the look of ticks and labels on an axis.

Options

Line style

- lpattern(linepatternstyle) whether line solid, dashed, etc.
- lwidth(linewidthstyle) thickness of line
- lcolor(colorstyle) color and opacity of line
- lalign(linealignmentstyle) line alignment (inside, outside, center)

Marker style

- msymbol(symbolstyle) (choice of symbol)
- mcolor(colorstyle) (choice of color and opacity)
- msize(markersizestyle) (choice of size)

Options

Other options

- yline(): Creates an horizontal line.
- xline(): Creates a vertical line.
- text(): Allows to introduce some text to a graph.
- plotregion(): Controls the style of the plot region.
- graphregion(): Controls the stye of the graph region.

Saving and Exporting

In the same way as with datasets, there are two possibilities:

- Save the graph in the Stata default ".gph". In that case, we will use graph save.
- Export the graph to a different file format. In that case, we will use graph export.

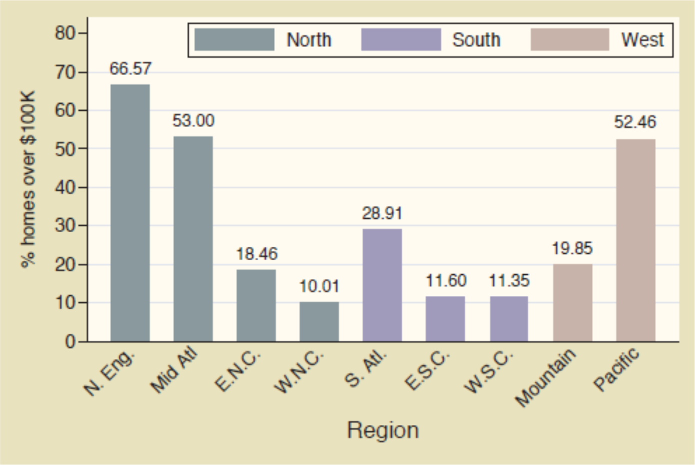
Finally, we can combine different figures using the combine command.

Exercices

1. Use the Stata web dataset womenwage: webuse womenwage.
2. Make a scatter plot between wage (wage) and schooling (school).
3. Plot the fitted line in the same graph.
4. Add title, subtitles,...
5. Make two scatter plots with their respective fitted lines, one for the subsample of women in rural areas and the other for the subsample of women in urban areas.
6. Combine the previous graphs.
7. Export the graph as a png file.

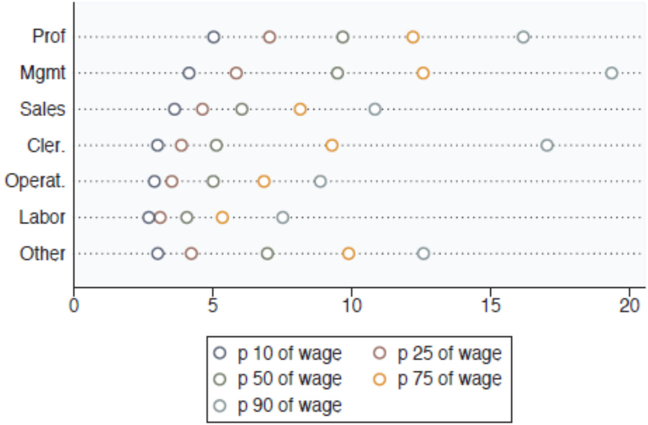
Exercices

Replicate using allstates data.



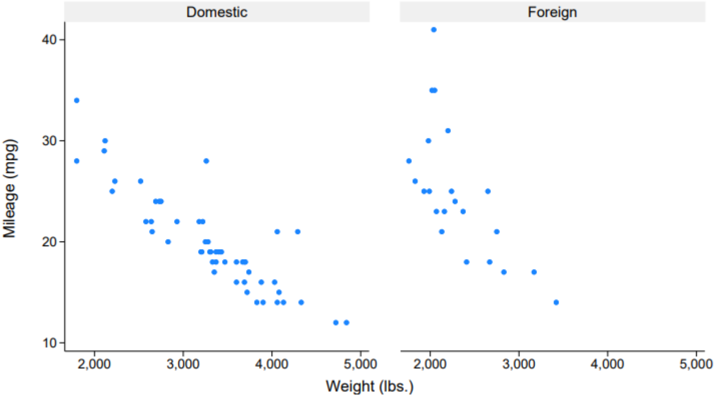
Exercices

Replicate using nlsw data.



Exercices

Replicate using auto data.



Final Exercice

Using the NSLY data set called "data1":

- Try to find evidence in favor of earning differences across race and gender.
- Use the variable poor to analyze if growing up poor had any effect. You could create pie chart of the fraction of poor individuals from each race. Also you could relate race, poor, and earnings in one graph.
- Repeat one of the previous plots but only for females.
- Are male respondents more likely to work full-time (i.e. 40 hours or more)?
- Explore the relationship between change in weight (from 1985 to 2002) and wages earned in 2002. Is there any difference by gender?