Title

stata.com

Graph intro — Introduction to graphics

Remarks and examples References Also see

Remarks and examples

stata.com

Remarks are presented under the following headings:

Suggested reading order

We recommend that you read the entries in this manual in the following order:

Read *A quick tour* below, then read *Quick start* in [G-1] **Graph Editor**, and then . . .

Entry	Description
[G-2] **graph**	Overview of the `graph` command
[G-2] **graph twoway**	Overview of the `graph twoway` command
[G-2] **graph twoway scatter**	Overview of the `graph twoway scatter` command

When reading those sections, follow references to other entries that interest you. They will take you to such useful topics as

Entry	Description
[G-3] ***marker_label_options***	Options for specifying marker labels
[G-3] ***by_option***	Option for repeating graph command
[G-3] ***title_options***	Options for specifying titles
[G-3] ***legend_options***	Option for specifying legend

We could list many, many more, but you will find them on your own. Follow the references that interest you, and ignore the rest. Afterward, you will have a working knowledge of twoway graphs. Now glance at each of

Entry	Description
[G-2] **graph twoway line**	Overview of the `graph twoway line` command
[G-2] **graph twoway connected**	Overview of the `graph twoway connected` command
etc.	

Turn to [G-2] **graph twoway**, which lists all the different `graph twoway` plottypes, and browse the manual entry for each.

Now is the time to understand schemes, which have a great effect on how graphs look. You may want to specify a different scheme before printing your graphs.

Entry	Description
[G-4] **Schemes intro**	Schemes and what they do
[G-2] **set printcolor**	Set how colors are treated when graphs are printed
[G-2] **graph print**	Printing graphs the easy way
[G-2] **graph export**	Exporting graphs to other file formats

Now you are an expert on the `graph twoway` command, and you can even print the graphs it produces.

To learn about the other types of graphs, see

Entry	Description
[G-2] **graph matrix**	Scatterplot matrices
[G-2] **graph bar**	Bar and dot charts
[G-2] **graph box**	Box plots
[G-2] **graph dot**	Dot charts (summary statistics)
[G-2] **graph pie**	Pie charts

To learn tricks of the trade, see

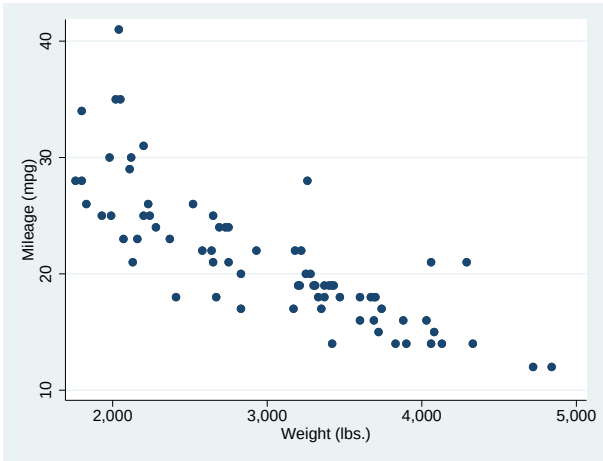
Entry	Description
[G-2] **graph save**	Saving graphs to disk
[G-2] **graph use**	Redisplaying graphs from disk
[G-2] **graph describe**	Finding out what is in a .gph file
[G-3] ***name_option***	How to name a graph in memory
[G-2] **graph display**	Display graph stored in memory
[G-2] **graph dir**	Obtaining directory of named graphs
[G-2] **graph rename**	Renaming a named graph
[G-2] **graph copy**	Copying a named graph
[G-2] **graph drop**	Eliminating graphs in memory
[G-2] **graph close**	Closing Graph windows
[G-2] **graph replay**	Replaying multiple graphs
[P] **discard**	Clearing memory

For a completely different and highly visual approach to learning Stata graphics, see Mitchell (2022). For a mix of scholarly review and tutorial exposition, see Cox (2014). Hamilton (2013) offers a concise 40-page overview within the larger context of statistical analysis with Stata. Excellent suggestions for presenting information clearly in graphs can be found in Cleveland (1993 and 1994), in Wallgren et al. (1996), and even in chapters of books treating larger subjects, such as Good and Hardin (2012).

A quick tour

`graph` is easy to use:

```
. use https://www.stata-press.com/data/r17/auto
(1978 automobile data)
. graph twoway scatter mpg weight
```



All the commands documented in this manual begin with the word `graph`, but often the `graph` is optional. You could get the same graph by typing

```
. twoway scatter mpg weight
```

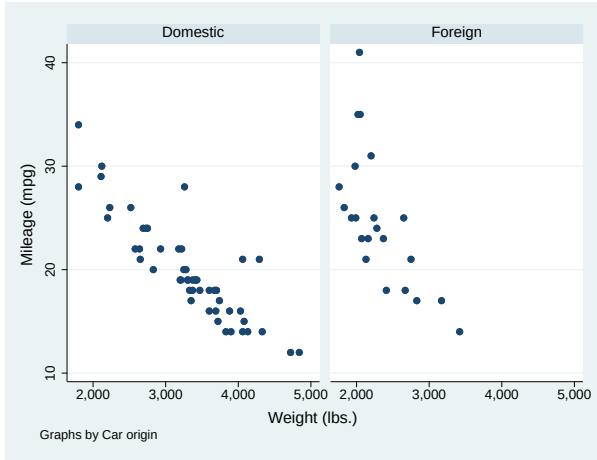
and, for `scatter`, you could omit the `twoway`, too:

```
. scatter mpg weight
```

We, however, will continue to type `twoway` to emphasize when the graphs we are demonstrating are in the twoway family.

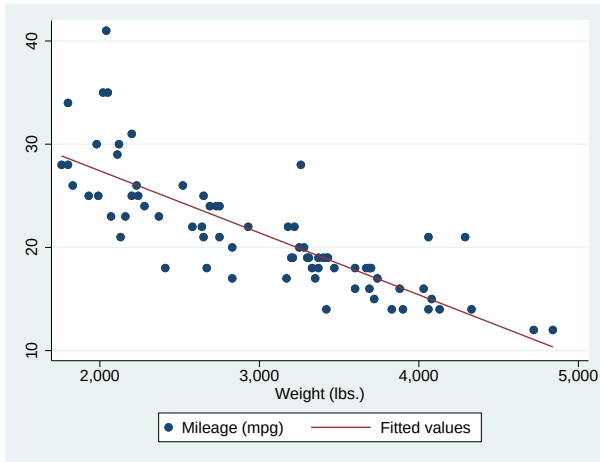
Twoway graphs can be combined with `by()`:

```
. twoway scatter mpg weight, by(foreign)
```



Graphs in the `twoway` family can also be overlaid. The members of the `twoway` family are called *plottypes*; `scatter` is a plottype, and another plottype is `lfit`, which calculates the linear prediction and plots it as a line chart. When we want one plottype overlaid on another, we combine the commands, putting `||` in between:

```
. twoway scatter mpg weight || lfit mpg weight
```



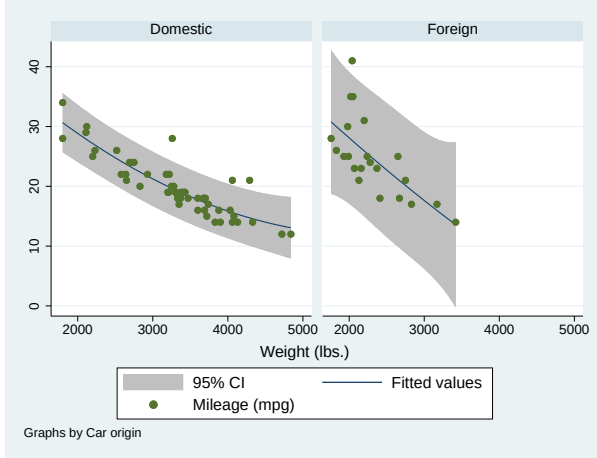
Another notation for this is called the `()`-binding notation:

```
. twoway (scatter mpg weight) (lfit mpg weight)
```

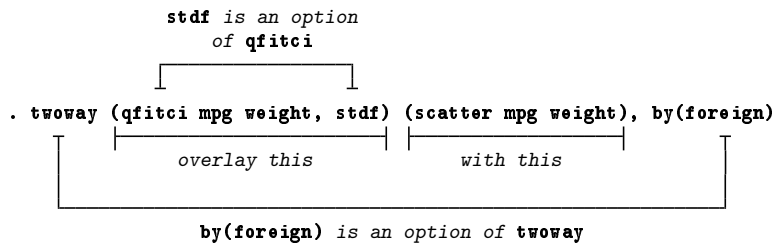
It does not matter which notation you use.

Overlaying can be combined with by(). This time, substitute qfitci for lfit. qfitci plots the prediction from a quadratic regression, and it adds a confidence interval. Then add the confidence interval on the basis of the standard error of the forecast:

```
. twoway (qfitci mpg weight, stdf) (scatter mpg weight), by(foreign)
```



We used the ()-binding notation just because it makes it easier to see what modifies what:



We could just as well have typed this command with the ||-separator notation,

```
. twoway qfitci mpg weight, stdf || scatter mpg weight ||, by(foreign)
```

and, as a matter of fact, we do not have to separate the twoway option by(foreign) (or any other twoway option) from the qfitci and scatter options, so we can type

```
. twoway qfitci mpg weight, stdf || scatter mpg weight, by(foreign)
```

or even

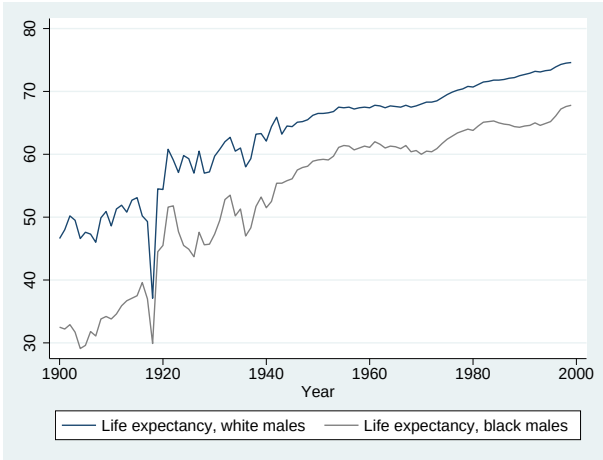
```
. twoway qfitci mpg weight, stdf by(foreign) || scatter mpg weight
```

All of these syntax issues are discussed in [G-2] **graph twoway**. In our opinion, the ()-binding notation is easier to read, but the ||-separator notation is easier to type. You will see us using both.

It was not an accident that we put qfitci first and scatter second. qfitci shades an area, and had we done it the other way around, that shading would have been put right on top of our scattered points and erased (or at least hidden) them.

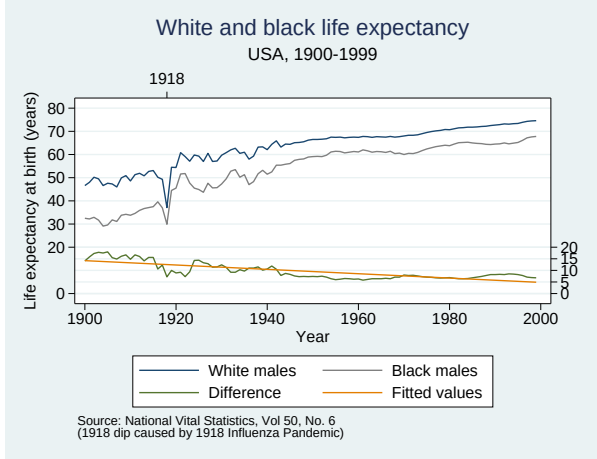
Plots of different types or the same type may be overlaid:

```
. use https://www.stata-press.com/data/r17/uslifeexp
(U.S. life expectancy, 1900-1999)
. twoway line le_wm year || line le_bm year
```



Here is a rather fancy version of the same graph:

```
. generate diff = le_wm - le_bm
. label var diff "Difference"
. twoway line le_wm year, yaxis(1 2) xaxis(1 2)
      || line le_bm year
      || line diff  year
      || lfit diff  year
      ||,
         ytitle( "",          axis(2) )
         xtitle( "",          axis(2) )
         xlabel( 1918,        axis(2) )
         ylabel( 0(5)20,      axis(2) grid gmin angle(horizontal) )
         ylabel( 0 20(10)80,              gmax angle(horizontal) )
         ytitle( "Life expectancy at birth (years)" )
         ylabel(, axis(2) grid)
         title( "White and black life expectancy" )
         subtitle( "USA, 1900-1999" )
         note( "Source: National Vital Statistics, Vol 50, No. 6"
               "(1918 dip caused by 1918 Influenza Pandemic)" )
         legend( label(1 "White males") label(2 "Black males") )
```



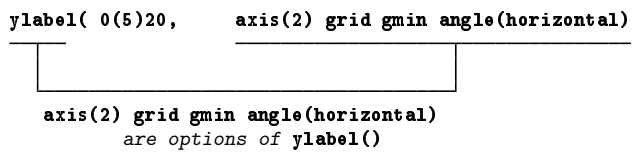
There are many options on this command. (All except the first two options could have been accomplished in the Graph Editor; see [G-1] **Graph Editor** for an overview of the Editor.) Strip away the obvious options, such as `title()`, `subtitle()`, and `note()`, and you are left with

```
. twoway line le_wm year, yaxis(1 2) xaxis(1 2)
      || line le_bm year
      || line diff  year
      || lfit diff  year
      ||,
         ytitle( "",          axis(2) )
         xtitle( "",          axis(2) )
         xlabel( 1918,        axis(2) )
         ylabel( 0(5)20,      axis(2) grid gmin angle(horizontal) )
         ylabel( 0 20(10)80,              gmax angle(horizontal) )
         legend( label(1 "White males") label(2 "Black males") )
```

Let's take the longest option first:

```
ylabel( 0(5)20,      axis(2) grid gmin angle(horizontal) )
```

The first thing to note is that options have options:



Now look back at our graph. It has two y axes, one on the right and a second on the left. Typing

```
ylabel( 0(5)20,      axis(2) grid gmin angle(horizontal) )
```

caused the right axis—`axis(2)`—to have labels at 0, 5, 10, 15, and 20—`0(5)20`. `grid` requested grid lines for each labeled tick on this right axis, and `gmin` forced the grid line at 0 because, by default, `graph` does not like to draw grid lines too close to the axis. `angle(horizontal)` made the 0, 5, 10, 15, and 20 horizontal rather than, as usual, vertical.

You can now guess what

```
ylabel( 0 20(10)80,          gmax angle(horizontal) )
```

did. It labeled the left y axis—`axis(1)` in the jargon—but we did not have to specify an `axis(1)` suboption because that is what `ylabel()` assumes. The purpose of

```
xlabel( 1918,       axis(2) )
```

is now obvious, too. That labeled a value on the second x axis.

So now we are left with

```
. twoway line le_wm year, yaxis(1 2) xaxis(1 2)
      || line le_bm year
      || line diff  year
      || lfit diff  year
      ||,
         ytitle( "",         axis(2) )
         xtitle( "",         axis(2) )
         legend( label(1 "White males") label(2 "Black males") )
```

Options `ytitle()` and `xtitle()` specify the axis titles. We did not want titles on the second axes, so we got rid of them. The `legend()` option,

```
legend( label(1 "White males") label(2 "Black males") )
```

merely respecified the text to be used for the first two keys. By default, `legend()` uses the variable label, which in this case would be the labels of variables `le_wm` and `le_bm`. In our dataset, those labels are "Life expectancy, white males" and "Life expectancy, black males". It was not necessary—and undesirable—to repeat "Life expectancy", so we specified an option to change the label. It was either that or change the variable label.

So now we are left with

```
. twoway line le_wm year, yaxis(1 2) xaxis(1 2)
      || line le_bm year
      || line diff  year
      || lfit diff  year
```

and that is almost perfectly understandable. The `yaxis()` and `xaxis()` options caused the creation of two y and two x axes rather than, as usual, one.

Understand how we arrived at

```
. twoway line le_wm year, yaxis(1 2) xaxis(1 2)
      || line le_bm year
      || line diff  year
      || lfit diff  year
      ||,
         ytitle( "",         axis(2) )
         xtitle( "",         axis(2) )
         xlabel( 1918,       axis(2) )
         ylabel( 0(5)20,     axis(2) grid gmin angle(horizontal) )
         ylabel( 0 20(10)80,             gmax angle(horizontal) )
         ytitle( "Life expectancy at birth (years)" )
         title( "White and black life expectancy" )
         subtitle( "USA, 1900-1999" )
         note( "Source: National Vital Statistics, Vol 50, No. 6"
               "(1918 dip caused by 1918 Influenza Pandemic)" )
         legend( label(1 "White males") label(2 "Black males") )
```

We started with the first graph we showed you,

```
. twoway line le_wm year || line le_bm year
```

and then, to emphasize the comparison of life expectancy for whites and blacks, we added the difference,

```
. twoway line le_wm year,
      || line le_bm year
      || line diff  year
```

and then, to emphasize the linear trend in the difference, we added "`lfit diff year`",

```
. twoway line le_wm year,
      || line le_bm year
      || line diff  year,
      || lfit diff  year
```

and then we added options to make the graph look more like what we wanted. We introduced the options one at a time. It was rather fun, really. As our command grew, we switched to using the Do-file Editor, where we could add an option and hit the **Do** button to see where we were. Because the command was so long, when we opened the Do-file Editor, we typed on the first line

```
#delimit ;
```

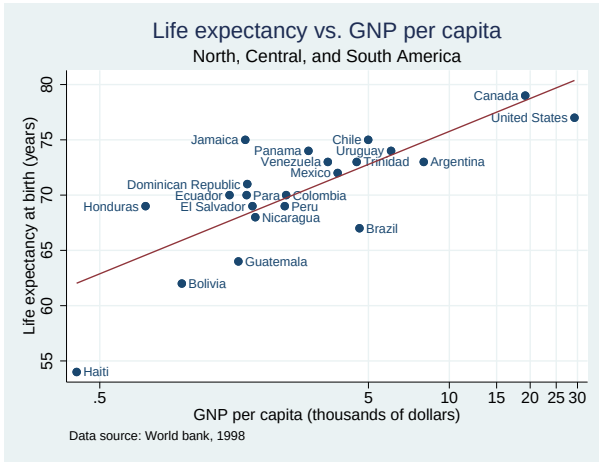
and we typed on the last line

```
;
```

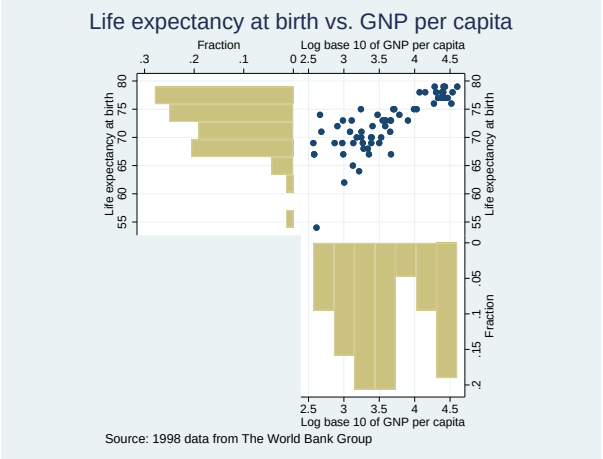
and then we typed our ever-growing command between.

Many of the options we used above are common to most of the graph families, including `twoway`, `bar`, `box`, `dot`, and `pie`. If you understand how the `title()` or `legend()` option is used with one family, you can apply that knowledge to all graphs, because these options work the same across families.

While we are on the subject of life expectancy, using another dataset, we drew



See [G-3] ***marker_label_options*** for an explanation of how we did this. Staying with life expectancy, we produced



which we drew by separately drawing three rather easy graphs

```
. twoway scatter lexp loggnp,
        yscale(alt) xscale(alt)
        xlabel(, grid gmax)                saving(yx)
. twoway histogram lexp, fraction
        xscale(alt reverse) horiz          saving(hy)
. twoway histogram loggnp, fraction
        yscale(alt reverse)
        ylabel(,nogrid)
        xlabel(,grid gmax)                 saving(hx)
```

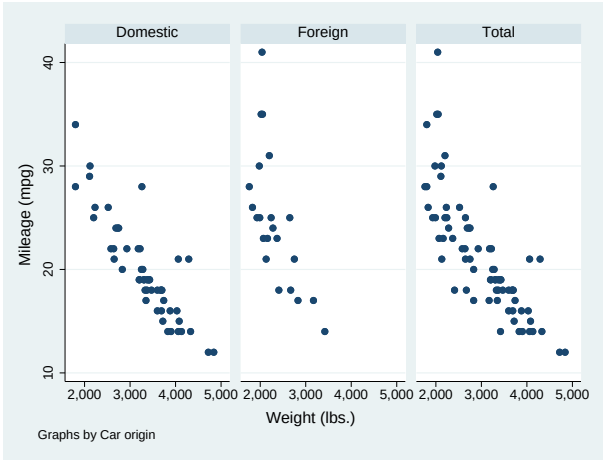
and then combining them:

```
. graph combine hy.gph yx.gph hx.gph,
        hole(3)
        imargin(0 0 0 0) grapharea(margin(l 22 r 22))
        title("Life expectancy at birth vs. GNP per capita")
        note("Source:  1998 data from The World Bank Group")
```

See [G-2] **graph combine** for more information.

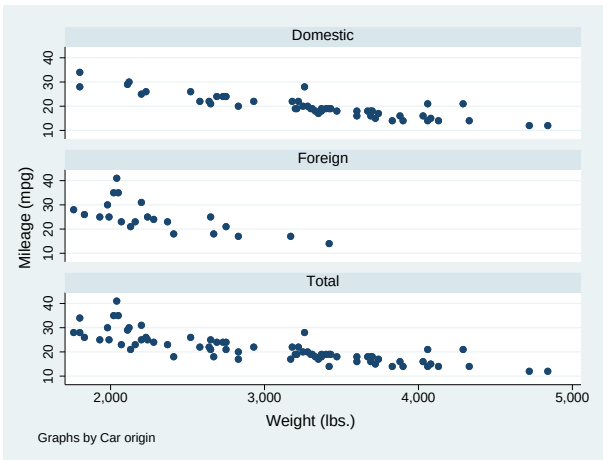
Back to our tour, `twoway, by()` can produce graphs that look like this

```
. use https://www.stata-press.com/data/r17/auto, clear
(1978 automobile data)
. scatter mpg weight, by(foreign, total row(1))
```



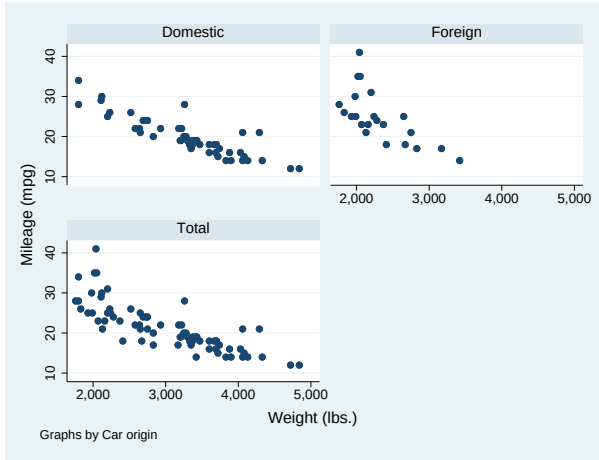
or this

```
. scatter mpg weight, by(foreign, total col(1))
```



or this

```
. scatter mpg weight, by(foreign, total)
```

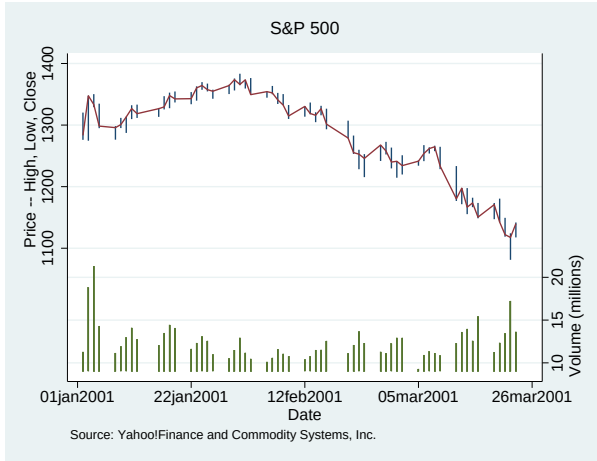


See [G-3] *by_option*.

`by()` is another of those options that is common across all graph families. If you know how to use it on one type of graph, then you know how to use it on any type of graph.

There are many plottypes within the `twoway` family, including areas, bars, spikes, dropped lines, and dots. Just to illustrate a few:

```
. use https://www.stata-press.com/data/r17/sp500
(S&P 500)
. replace volume = volume/1000
(248 real changes made)
. twoway
        rspike hi low date ||
        line   close  date ||
        bar    volume date, barw(.25) yaxis(2) ||
  in 1/57
  , yscale(axis(1) r(900 1400))
    yscale(axis(2) r(  9   45))
    ytitle("                          Price -- High, Low, Close")
    ytitle(" Volume (millions)", axis(2) bexpand just(left))
    legend(off)
    subtitle("S&P 500", margin(b+2.5))
    note("Source:  Yahoo!Finance and Commodity Systems, Inc.")
```

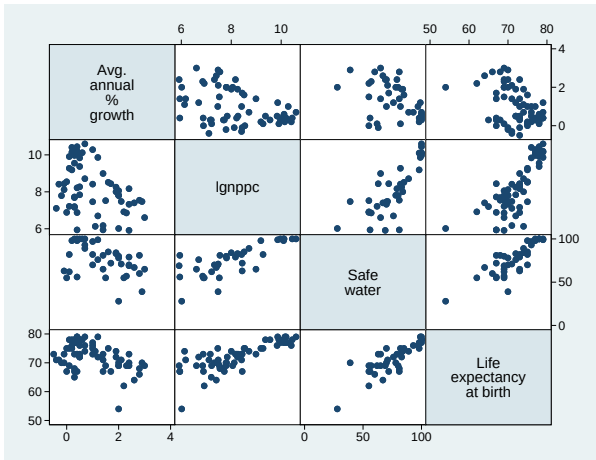


The above graph is explained in [G-2] **graph twoway rspike**. See [G-2] **graph twoway** for a listing of all available `twoway` plottypes.

Moving outside the `twoway` family, `graph` can draw scatterplot matrices, box plots, pie charts, and bar and dot plots. Here are examples of each.

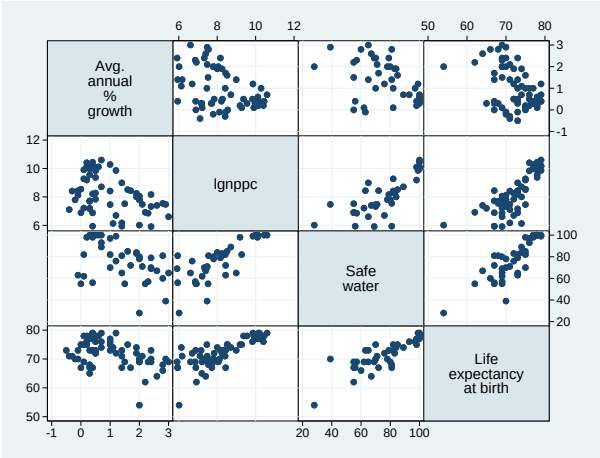
A scatterplot matrix of the variables `popgr`, `lexp`, `lgnppc`, and `safe`:

```
. use https://www.stata-press.com/data/r17/lifeexp, clear
(Life expectancy, 1998)
. generate lgnppc = ln(gnppc)
(5 missing values generated)
. graph matrix popgr lgnppc safe lexp
```



Or, with grid lines and more axis labels:

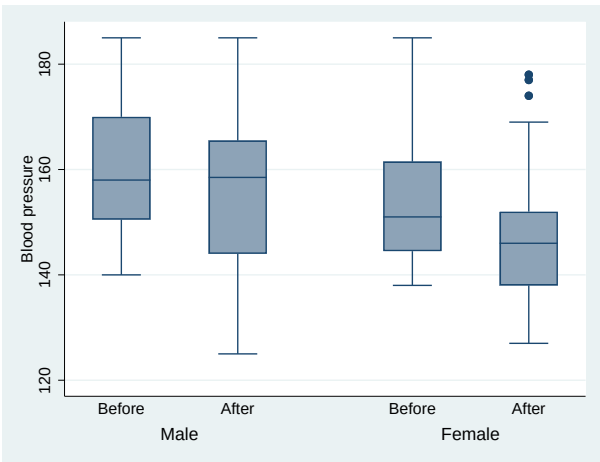
```
. graph matrix popgr lgnppc safe lexp, maxes(ylab(#4, grid) xlab(#4, grid))
```



See [G-2] **graph matrix**.

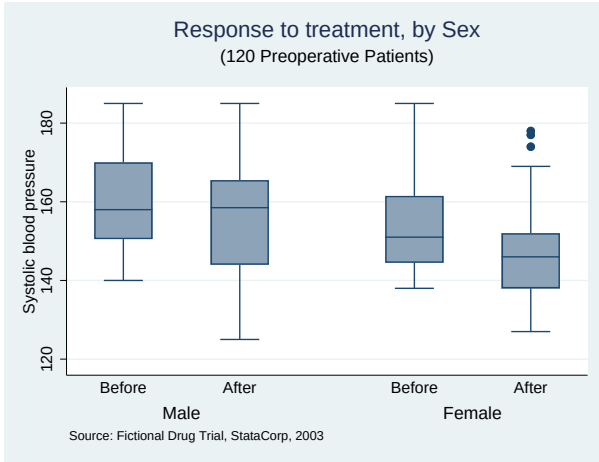
A box plot of blood pressure, variable `bp`, over each group in the variable `when` and each group in the variable `sex`:

```
. use https://www.stata-press.com/data/r17/bplong, clear
(Fictional blood-pressure data)
. graph box bp, over(when) over(sex)
```



Or, for a graph with complete titles:

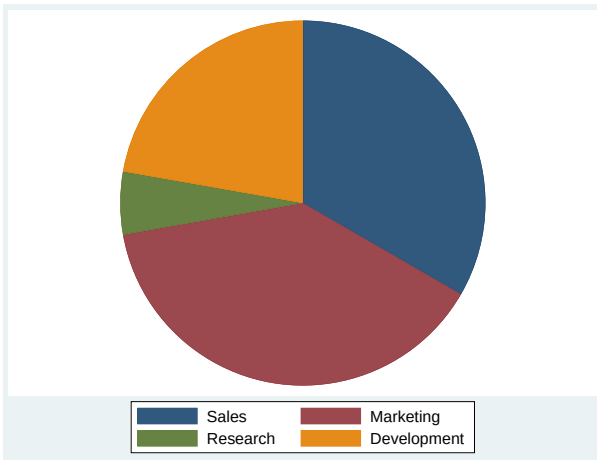
```
. graph box bp, over(when) over(sex)
        ytitle("Systolic blood pressure")
        title("Response to Treatment, by Sex")
        subtitle("(120 Preoperative Patients)" " ")
        note("Source:  Fictional Drug Trial, StataCorp, 2003")
```



See [G-2] **graph box**.

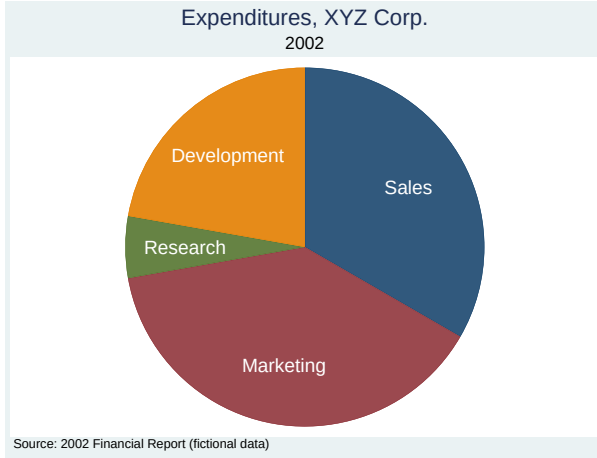
A pie chart showing the proportions of the variables `sales`, `marketing`, `research`, and `development`:

```
. graph pie sales marketing research development
```



Or, for a graph with nice titles and better labeling of the pie slices:

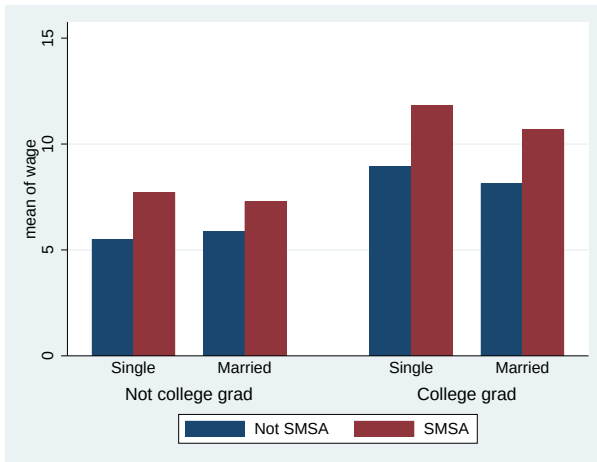
```
. graph pie sales marketing research development,
        plabel(_all name, size(*1.5) color(white))
        legend(off)
        plotregion(lstyle(none))
        title("Expenditures, XYZ Corp.")
        subtitle("2002")
        note("Source:  2002 Financial Report (fictional data)")
```



See [G-2] **graph pie**.

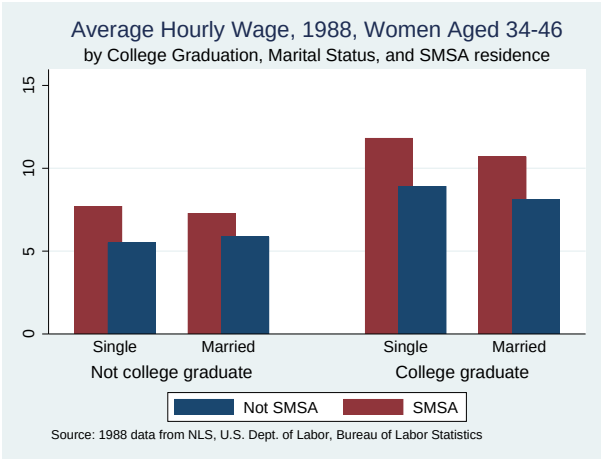
A vertical bar chart of average wages over each group in the variables `smsa`, `married`, and `collgrad`:

```
. use https://www.stata-press.com/data/r17/nlsw88
(NLSW, 1988 extract)
. graph bar wage, over(smsa) over(married) over(collgrad)
```



Or, for a prettier graph with overlapping bars, titles, and better labels:

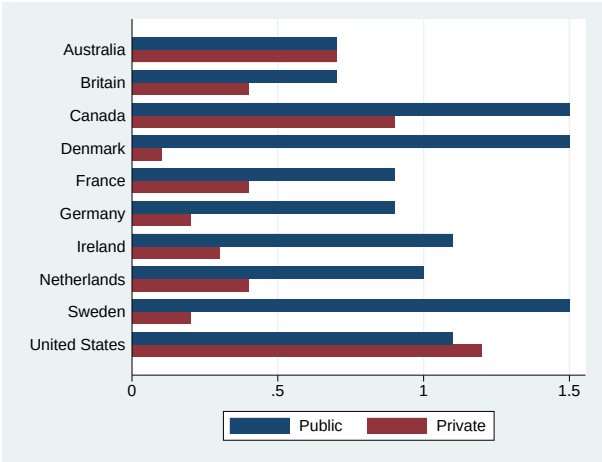
```
. graph bar wage,
                over( smsa, descend gap(-30) )
                over( married )
                over( collgrad, relabel(1 "Not college graduate"
                                        2 "College graduate"    ) )
                ytitle("")
                title("Average Hourly Wage, 1988, Women Aged 34-46")
                subtitle("by College Graduation, Marital Status,
                          and SMSA residence")
                note("Source:  1988 data from NLS, U.S. Dept of Labor,
                      Bureau of Labor Statistics")
```



See [G-2] **graph bar**.

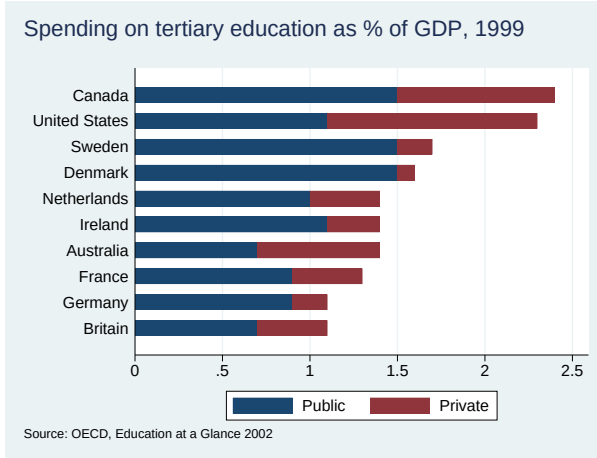
A horizontal bar chart of private versus public spending over countries:

```
. use https://www.stata-press.com/data/r17/educ99gdp
(Education and GDP)
. generate total = private + public
. graph hbar (asis) public private, over(country)
```



Or, the same information with stacked bars, an informative sorting of total spending, and nice titles:

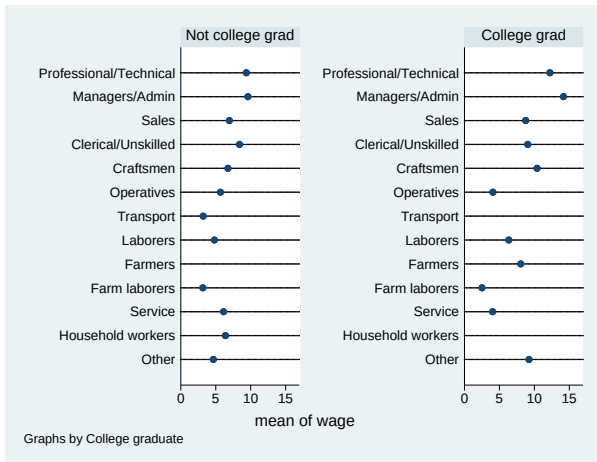
```
. graph hbar (asis) public private,
                over(country, sort(total) descending)
                stack
                title("Spending on tertiary education as % of GDP,
                       1999", span position(11) )
                subtitle(" ")
                note("Source:  OECD, Education at a Glance 2002", span)
```



See [G-2] **graph bar**.

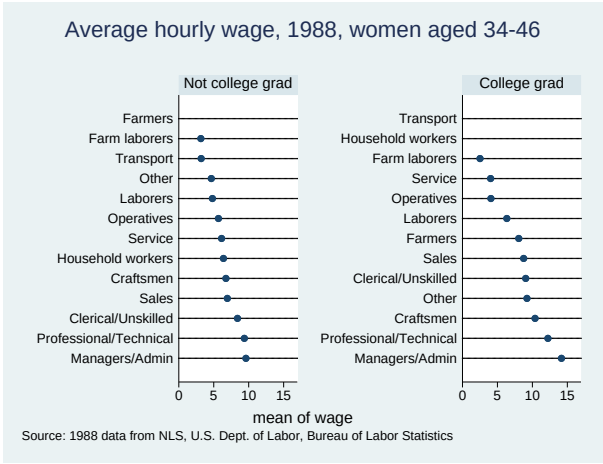
A dot chart of average hourly wage over occupation, variable `occ`, with separate subgraphs for college graduates and not college graduates, variable `collgrad`:

```
. use https://www.stata-press.com/data/r17/nlsw88, clear
(NLSW, 1988 extract)
. graph dot wage, over(occ) by(collgrad)
```



Or, for a plot that orders the occupations by wage and has nice titles:

```
. graph dot wage,
        over(occ, sort(1))
        by(collgrad,
             title("Average hourly wage, 1988, women aged 34-46", span)
             subtitle(" ")
             note("Source:  1988 data from NLS, U.S. Dept. of Labor,
                   Bureau of Labor Statistics", span)
        )
```



See [G-2] **graph dot**.

Have fun. Follow our advice in the *Suggested reading order* above: turn to [G-2] **graph**, [G-2] **graph twoway**, and [G-2] **graph twoway scatter**.

Using the menus

In addition to using the command-line interface, you can access most of `graph`'s features by Stata's pulldown menus. To start, load a dataset, select **Graphics**, and select what interests you.

When you have finished filling in the dialog box (do not forget to click on the tabs—lots of useful features are hidden there), rather than click on **OK**, click on **Submit**. This way, once the graph appears, you can easily modify it and click on **Submit** again.

Feel free to experiment. Clicking on **Submit** (or **OK**) never hurts; if you have left a required field blank, you will be told. The dialog boxes make it easy to spot what you can change.

References

Cleveland, W. S. 1993. *Visualizing Data*. Summit, NJ: Hobart.

——. 1994. *The Elements of Graphing Data*. Rev. ed. Summit, NJ: Hobart.

Cox, N. J. 2004a. Speaking Stata: Graphing distributions. *Stata Journal* 4: 66–88.

——. 2004b. Speaking Stata: Graphing categorical and compositional data. *Stata Journal* 4: 190–215.

——. 2004c. Speaking Stata: Graphing agreement and disagreement. *Stata Journal* 4: 329–349.

——. 2004d. Speaking Stata: Graphing model diagnostics. *Stata Journal* 4: 449–475.

——. 2014. *Speaking Stata Graphics*. College Station, TX: Stata Press.

——. 2019. Speaking Stata: Some simple devices to ease the spaghetti problem. *Stata Journal* 19: 989–1008.

——. 2021. Speaking Stata: Front-and-back plots to ease spaghetti and paella problems. *Stata Journal* 21: 539–554.

Good, P. I., and J. W. Hardin. 2012. *Common Errors in Statistics (and How to Avoid Them)*. 4th ed. Hoboken, NJ: Wiley.

Hamilton, L. C. 2013. *Statistics with Stata: Updated for Version 12*. 8th ed. Boston: Brooks/Cole.

Hoffmann, J. P. 2017. *Principles of Data Management and Presentation*. Oakland, CA: University of California Press.

Mitchell, M. N. 2022. *A Visual Guide to Stata Graphics*. 4th ed. College Station, TX: Stata Press.

Wallgren, A., B. Wallgren, R. Persson, U. Jorner, and J.-A. Haaland. 1996. *Graphing Statistics and Data: Creating Better Charts*. Newbury Park, CA: SAGE.

Also see

[G-2] **graph** — The graph command

[G-2] **graph other** — Other graphics commands

[G-1] **Graph Editor** — Graph Editor