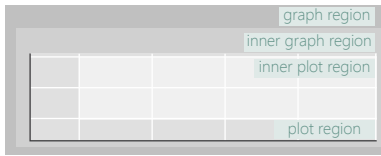


Plotting in Stata 14.1

Customizing Appearance

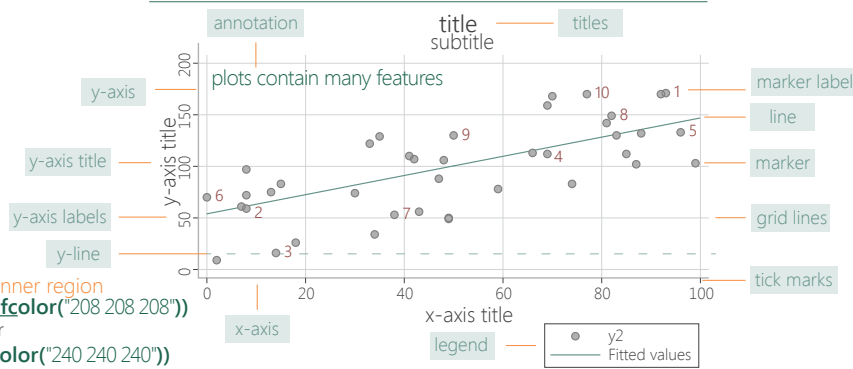
For more info see Stata's reference manual (stata.com)



`scatter price mpg, graphregion(fcolor("192 192 192") ifcolor("208 208 208"))`
 specify the fill of the background in RGB or with a Stata color

`scatter price mpg, plotregion(fcolor("224 224 224") ifcolor("240 240 240"))`
 specify the fill of the plot background in RGB or with a Stata color

ANATOMY OF A PLOT



SYMBOLS

SYNTAX

`marker` arguments for the plot objects (in green) go in the options portion of these commands (in orange)
 for example:
`scatter price mpg, xline(20, lwidth(vthick))`

COLOR

`mcolor("145 168 208")` specify the fill and stroke of the marker in RGB or with a Stata color

`mlcolor("145 168 208")` specify the fill of the marker

SIZE / THICKNESS

`msize(medium)` specify the marker size:

	ehuge		medlarge
	vhuge		medium
	huge		medsmall
	vlarge		small
	large		vsmall
			tiny
			vtiny

APPEARANCE

`msymbol(Dh)` specify the marker symbol:

	O		D		T		S
	o		d		t		s
	Oh		Dh		Th		Sh
	oh		dh		th		sh
	+		X		.		none
			X		.		none
			X		.		none

POSITION

`jitter(#)` randomly displace the markers

`jitterseed(#)` set seed

LINES / BORDERS

`line` arguments for the plot lines (in green) go in the options portion of these commands (in orange)

`xline(...)` `ylines(...)`

`width(medthick)` specify the thickness (stroke) of a line:

	vwthick		medthick
	vthick		thin
	vwthick		vthin
	thick		vvthin
	medthick		none
	medium		

`lcolor("145 168 208")` specify the stroke color of the line or border

`mlcolor("145 168 208")`

`tlcolor("145 168 208")`

`glcolor("145 168 208")`

`width(medthick)` specify the thickness (stroke) of a line:

	vwthick		medthick
	vthick		thin
	vwthick		vthin
	thick		vvthin
	medthick		none
	medium		

`line` `axes` `lpattern(dash)` specify the line pattern

`grid lines` `glpattern(dash)`

	solid		longdash		longdash_dot
	dash		shortdash		shortdash_dot
	dot		dash_dot		blank

`axes` `noline` `axes` `off` no axis/labels

`tick marks` `noticks` `tick marks` `length(2)`

`grid lines` `nogrid` `nogmin` `nogmax`

`tick marks` `xlabel(#10, tposition(crossing))` number of tick marks, position (outside | crossing | inside)

TEXT

`marker label` `titles` `axis labels`

`<marker options>` `title(...)` `xlabel(...)`

`annotation` `text(...)` `ytile(...)` `legend`

`color("145 168 208")` specify the color of the text

`mlabcolor("145 168 208")`

`labcolor("145 168 208")`

`size(medsmall)` specify the size of the text:

	vhuge		medsmall
	huge		small
	vhuge		vsmall
	large		tiny
	medlarge		half_tiny
	medium		third_tiny
			quarter_tiny
			minuscule

`marker label` `mlabel(foreign)` label the points with the values of the foreign variable

`axis labels` `nolabels` no axis labels

`axis labels` `format(%12.2f)` change the format of the axis labels

`legend` `off` turn off legend

`legend` `label("# label")` change legend label text

`marker label` `mlabposition(5)` label location relative to marker (clock position: 0 – 12)

Apply Themes

Schemes are sets of graphical parameters, so you don't have to specify the look of the graphs every time.

USING A SAVED THEME

`twoway scatter mpg price, scheme(customTheme)`

help scheme entries Create custom themes by saving options in a .scheme file
 see all options for setting scheme properties

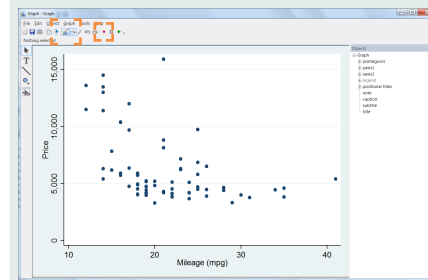
`adopath ++ "~/<location>/StataThemes"`
 set path of the folder (StataThemes) where custom .scheme files are saved

`set scheme customTheme, permanently`
 change the theme

`net inst brewscheme, from("https://wbuchanan.github.io/brewscheme/")` replace
 install William Buchanan's package to generate custom schemes and color palettes (including ColorBrewer)

USING THE GRAPH EDITOR

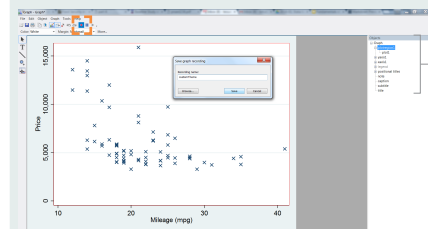
`twoway scatter mpg price, play(graphEditorTheme)`



Select the Graph Editor



Click Record



Double click on symbols and areas on plot, or regions on sidebar to customize

Unclick Record



Save theme as a .grec file

Save Plots

`graph twoway scatter y x, saving("myPlot.gph")` replace
 save the graph when drawing

`graph save "myPlot.gph", replace`
 save current graph to disk

`graph combine plot1.gph plot2.gph...`
 combine 2+ saved graphs into a single plot

`graph export "myPlot.pdf", as(.pdf)` see options to set size and resolution
 export the current graph as an image file