Data Visualization with Stata 15 Cheat Sheet

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

Basic Plot Syntax:

variables: y first — plot-specific options — – facet – — annotations

graph <plot type> y_1 y_2 ... y_n x **[in]** **[if],** <plot options> **by(var)** **xline(**xint**) yline(**yint**) text(**y x "annotation"**)**

titles — axes

title("title**") subtitle("**subtitle**") xtitle("**x-axis title**") ytitle("**y axis title**") xscale(range(**low high**) log reverse off noline) yscale(**<options>**)**

custom appearance — plot size — save

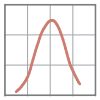
<marker, line, text, axis, legend, background options> **scheme(**s1mono**) play(**customTheme**) xsize(**5**) ysize(**4**) saving("**myPlot.gph**", replace)**

One Variable

sysuse auto, clear

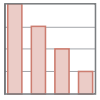
Continuous

histogram mpg, width(5) freq kdensity kdenopts(bwidth(5))
histogram
bin(#) • width(#) • density • fraction • frequency • percent • addlabels addlabopts(<options>) • normal • normopts(<options>) • kdensity kdenopts(<options>)

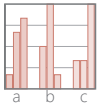


kdensity mpg, bwidth(3)
smoothed histogram
bwidth • kernel(<options> ← main plot-specific options; see help for complete set
normal • normopts(<line options>)

Discrete

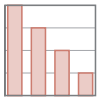


graph bar (count), over(foreign, gap(*0.5)) intensity(*0.5)
bar plot graph hbar draws horizontal bar charts
(asis) • (percent) • (count) • over(<variable>, <options: gap(*#) • relabel • descending • reverse>) • cw •missing • nofill • allcategories • percentages • stack • bargap(#) • intensity(*#) • yalternate • xalternate

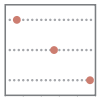


graph bar (percent), over(rep78) over(foreign)
grouped bar plot graph hbar ...
(asis) • (percent) • (count) • over(<variable>, <options: gap(*#) • relabel • descending • reverse>) • cw •missing • nofill • allcategories • percentages • stack • bargap(#) • intensity(*#) • yalternate • xalternate

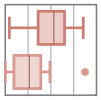
Discrete X, Continuous Y



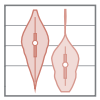
graph bar (median) price, over(foreign) graph hbar ...
bar plot (asis) • (percent) • (count) • (stat: mean median sum min max ...) over(<variable>, <options: gap(*#) • relabel • descending • reverse sort(<variable>)>) • cw • missing • nofill • allcategories • percentages stack • bargap(#) • intensity(*#) • yalternate • xalternate



graph dot (mean) length headroom, over(foreign) m(1, ms(S))
dot plot (asis) • (percent) • (count) • (stat: mean median sum min max ...) over(<variable>, <options: gap(*#) • relabel • descending • reverse sort(<variable>)>) • cw • missing • nofill • allcategories • percentages linegap(#) • marker(#, <options>) • linetype(dot | line | rectangle) dots(<options>) • lines(<options>) • rectangles(<options>) • rwidth



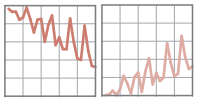
graph hbox mpg, over(rep78, descending) by(foreign) missing
box plot graph box draws vertical boxplots
over(<variable>, <options: total • gap(*#) • relabel • descending • reverse sort(<variable>)>) • missing • allcategories • intensity(*#) • boxgap(#) medtype(line | line | marker) • medline(<options>) • medmarker(<options>)



violplot price, over(foreign) ssc install violplot
violin plot over(<variable>, <options: total • missing>) • nofill • vertical • horizontal • obs • kernel(<options>) • bwidth(#) • barwidth(#) • dscale(#) • ygap(#) • ogap(#) • density(<options>) bar(<options>) • median(<options>) • obsopts(<options>)

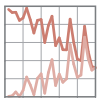
Plot Placement

Juxtapose (Facet)



twoway scatter mpg price, **by(foreign, norescale)**
total • missing • colfirst • rows(#) • cols(#) • holes(<numlist>) compact • [no]edgelabel • [no]rescale • [no]yrescal • [no]xrescale [no]iyaxes • [no]ixaxes • [no]iytick • [no]ixtick • [no]iylabel [no]ixlabel • [no]iytitle • [no]ixtitle • imargin(<options>)

Superimpose



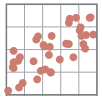
graph combine plot1.gph plot2.gph...
combine 2+ saved graphs into a single plot

scatter y3 y2 y1 x, msymbol(i o i) mlabel(var3 var2 var1)
plot several y values for a single x value

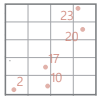
graph twoway scatter mpg price in 27/74 || **scatter** mpg price /* */ if mpg < 15 & price > 12000 in 27/74, mlabel(make) m(i)
combine twoway plots using ||

Two+ Continuous Variables

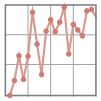
graph matrix mpg price weight, half
scatter plot of each combination of variables
half • jitter(#) • jitterseed(#) diagonal • [aweights(<variable>)]



twoway scatter mpg weight, jitter(7)
scatter plot
jitter(#) • jitterseed(#) • sort • cmissing(yes | no) connect(<options>) • [aweight(<variable>)]

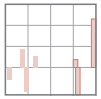


twoway scatter mpg weight, **mlabel(**mpg**)**
scatter plot with labelled values
jitter(#) • jitterseed(#) • sort • cmissing(yes | no) connect(<options>) • [aweight(<variable>)]

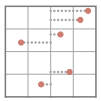


twoway connected mpg price, sort(price)
scatter plot with connected lines and symbols
jitter(#) • jitterseed(#) • sort see also line
connect(<options>) • cmissing(yes | no)

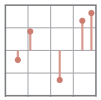
twoway area mpg price, sort(price)
line plot with area shading
sort • cmissing(yes | no) • vertical, • horizontal base(#)



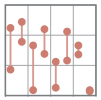
twoway bar price rep78
bar plot
vertical, • horizontal • base(#) • barwidth(#)



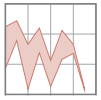
twoway dot mpg rep78
dot plot vertical, • horizontal • base(#) • ndots(#) dcolor(<color>) • dfcolor(<color>) • dlcolor(<color>) dsize(<markersize>) • dsymbol(<marker type>) dlwidth(<strokesize>) • dotextend(yes | no)



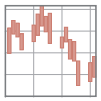
twoway dropline mpg price **in** 1/5
dropped line plot
vertical, • horizontal • base(#)



twoway rcapsym length headroom price
range plot (y_1 ÷ y_2) with capped lines
vertical • horizontal see also rcap



twoway rarea length headroom price, sort
range plot (y_1 ÷ y_2) with area shading
vertical • horizontal • sort
cmissing(yes | no)



twoway rbar length headroom price
range plot (y_1 ÷ y_2) with bars
vertical • horizontal • barwidth(#) • mwidth msize(<marker size>)

twoway pcspike wage68 ttl_exp68 wage88 ttl_exp88
Parallel coordinates plot (sysuse nlswide1)
vertical, • horizontal

twoway pccapsym wage68 ttl_exp68 wage88 ttl_exp88
Slope/bump plot (sysuse nlswide1)
vertical • horizontal • headlabel

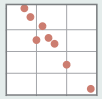
Three Variables

twoway contour mpg price weight, level(20) crule(intensity)
3D contour plot
ccuts(#s) • levels(#) • minmax • crule(hue | chue | intensity | linear) • scolor(<color>) • ecolor (<color>) • ccolors(<colorlist>) • heatmap interp(thinplatespline | shepard | none)

regress price mpg trunk weight length turn, **nocons**
matrix regmat = e(V) ssc install plotmatrix
plotmatrix, mat(regmat**) color(**green**)**
heatmap mat(<variable) • split(<options>) • color(<color>) • freq

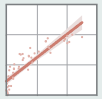
Summary Plots

twoway mband mpg weight || **scatter** mpg weight
plot median of the y values
bands(#)

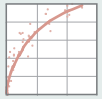


binscatter weight mpg, line(none) ssc install binscatter
plot a single value (mean or median) for each x value
medians • nquantiles(#) • discrete • controls(<variables>) • linetype(lfit | qfit | connect | none) • aweight[<variable>]

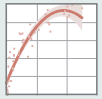
Fitting Results



twoway lfitci mpg weight || **scatter** mpg weight
calculate and plot linear fit to data with confidence intervals
level(#) • stdp • stdf • nofit • fitplot(<plottype>) • ciplot(<plottype>) • range(# #) • n(#) • atobs • estopts(<options>) • predopts(<options>)

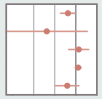


twoway lowess mpg weight || **scatter** mpg weight
calculate and plot lowess smoothing
bwidth(#) • mean • noweight • logit • adjust

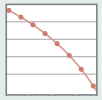


twoway qfitci mpg weight, alwidth(none) || **scatter** mpg weight
calculate and plot quadriatic fit to data with confidence intervals
level(#) • stdp • stdf • nofit • fitplot(<plottype>) • ciplot(<plottype>) • range(# #) • n(#) • atobs • estopts(<options>) • predopts(<options>)

Regression Results



regress price mpg headroom trunk length turn
coefplot, drop(_cons**) xline(**0**)** ssc install coefplot
Plot regression coefficients
baselevels • b(<options>) • at(<options>) • noci • levels(#) keep(<variables>) • drop(<variables>) • rename(<list>) horizontal • vertical • generate(<variable>)



regress mpg weight length turn
margins, eyex(weight**) at(**weight = (1800(200)4800))
marginsplot, noci
Plot marginal effects of regression
horizontal • noci

Laura Hughes (lhughes@usaid.gov) • Tim Essam (tessam@usaid.gov) inspired by RStudio's awesome Cheat Sheets (rstudio.com/resources/cheatsheets)
follow us @flaneuseks and @StataRGIS

geocenter.github.io/StataTraining updated February 2016