graph matrix — Matrix graphs

Description

`graph matrix` draws scatterplot matrices.

Quick start

Scatterplot matrix for variables `v1`, `v2`, `v3`, `v4`, and `v5`

```
graph matrix v1 v2 v3 v4 v5
```

Same as above, but draw only the lower triangle

```
graph matrix v1 v2 v3 v4 v5, half
```

Separate scatterplot matrices for each level of `catvar`

```
graph matrix v1 v2 v3 v4 v5, by(catvar)
```

With hollow circles as markers

```
graph matrix v1 v2 v3 v4 v5, half msymbol(Oh)
```

Same as above, but with periods as markers

```
graph matrix v1 v2 v3 v4 v5, half msymbol(p)
```

Override the default text on the diagonal for `v1` and `v3`

```
graph matrix v1 v2 v3 v4 v5, diagonal("Variable 1" . "Variable 3")
```

Menu

Graphics > Scatterplot matrix

Syntax

```
graph matrix varlist [if] [in] [weight] [, options]
```

options	Description
`half`	draw lower triangle only
marker_options	look of markers
marker_label_options	include labels on markers
`jitter(#)`	perturb location of markers
`jitterseed(#)`	random-number seed for `jitter()`
`diagonal(`*stringlist*`, ...)`	override text on diagonal
`diagopts(`*textbox_options*`)`	rendition of text on diagonal
`scale(#)`	overall size of symbols, labels, etc.
`iscale([*]#)`	size of symbols, labels, within plots
`maxes(`*axis_scale_options* *axis_label_options*`)`	labels, ticks, grids, log scales, etc.
axis_label_options	axis-by-axis control
`by(`*varlist*`, ...)`	repeat for subgroups
std_options	title, aspect ratio, saving to disk

All options allowed by `graph twoway scatter` are also allowed, but they are ignored.

`half`, `diagonal()`, `scale()`, and `iscale()` are *unique*; `jitter()` and `jitterseed()` are *rightmost* and `maxes()` is *merged-implicit*; see [G-4] **Concept: repeated options**.

stringlist`, ...`, the argument allowed by `diagonal()`, is defined

```
[ { . | "string" } ] [ { . | "string" } ... ] [ , textbox_options ]
```

`aweights`, `fweights`, and `pweights` are allowed; see **[U] 11.1.6 weight**. Weights affect the size of the markers. See *Weighted markers* in [G-2] **graph twoway scatter**.

Options

`half` specifies that only the lower triangle of the scatterplot matrix be drawn.

marker_options specify the look of the markers used to designate the location of the points. The important *marker_options* are `msymbol()`, `mcolor()`, and `msize()`.

The default symbol used is `msymbol(O)`—solid circles. You specify `msymbol(Oh)` if you want hollow circles (a recommended alternative). If you have many observations, we recommend specifying `msymbol(p)`; see *Marker symbols and the number of observations* under *Remarks and examples* below. See [G-4] ***symbolstyle*** for a list of marker symbol choices.

The default `mcolor()` is dictated by the scheme; see [G-4] **Schemes intro**. See [G-4] ***colorstyle*** for a list of color choices.

Be careful specifying the `msize()` option. In `graph matrix`, the size of the markers varies with the number of variables specified; see option `iscale()` below. If you specify `msize()`, that will override the automatic scaling.

See [G-3] ***marker_options*** for more information on markers.

marker_label_options allow placing identifying labels on the points. To obtain this, you specify the *marker_label_option* `mlabel(`*varname*`)`; see [G-3] ***marker_label_options***. These options are of little use for scatterplot matrices because they make the graph seem too crowded.

`jitter(`#`)` adds spherical random noise to the data before plotting. # represents the size of the noise as a percentage of the graphical area. This is useful when plotting data which otherwise would result in points plotted on top of each other. See *Jittered markers* in [G-2] **graph twoway scatter** for an explanation of jittering.

`jitterseed(`#`)` specifies the seed for the random noise added by the `jitter()` option. # should be specified as a positive integer. Use this option to reproduce the same plotted points when the `jitter()` option is specified.

`diagonal(`[*stringlist*][`,` *textbox_options*]`)` specifies text and its style to be displayed along the diagonal. This text serves to label the graphs (axes). By default, what appears along the diagonals are the variable labels of the variables of *varlist* or, if a variable has no variable label, its name. Typing

```
. graph matrix mpg weight displ, diag(. "Weight of car")
```

would change the text appearing in the cell corresponding to variable `weight`. We specified period (`.`) to leave the text in the first cell unchanged, and we did not bother to type a third string or a period, so we left the third element unchanged, too.

You may specify *textbox_options* following *stringlist* (which may itself be omitted) and a comma. These options will modify the style in which the text is presented but are of little use here. We recommend that you do not specify `diagonal(,size())` to override the default sizing of the text. By default, the size of text varies with the number of variables specified; see option `iscale()` below. Specifying `diagonal(,size())` will override the automatic size scaling. See [G-3] ***textbox_options*** for more information on textboxes.

`diagopts(`*textbox_options*`)` specify the look of text on the diagonal. This option is a shortcut for `diagonal(,` *textbox_options*`)`.

`scale(`#`)` specifies a multiplier that affects the size of all text and markers in a graph. `scale(1)` is the default, and `scale(1.2)` would make all text and markers 20% larger.
See [G-3] ***scale_option***.

`iscale(`#`)` and `iscale(*`#`)` specify an adjustment (multiplier) to be used to scale the markers, the text appearing along the diagonals, and the labels and ticks appearing on the axes.

By default, `iscale()` gets smaller and smaller the larger n is, the number of variables specified in *varlist*. The default is parameterized as a multiplier $f(n)$—$0 < f(n) < 1$, $f'(n) < 0$—that is used as a multiplier for `msize()`, `diagonal(,size())`, `maxes(labsize())`, and `maxes(tlength())`.

If you specify `iscale(`#`)`, the number you specify is substituted for $f(n)$. We recommend that you specify a number between 0 and 1, but you are free to specify numbers larger than 1.

If you specify `iscale(*`#`)`, the number you specify is multiplied by $f(n)$, and that product is used to scale text. Here you should specify # > 0; # > 1 merely means you want the text to be bigger than `graph matrix` would otherwise choose.

`maxes(`*axis_scale_options axis_label_options*`)` affect the scaling and look of the axes. This is a case where you specify options within options.

Consider the *axis_scale_options* { y | x }scale(log), which produces logarithmic scales. Type maxes(yscale(log) xscale(log)) to draw the scatterplot matrix by using log scales. Remember to specify both xscale(log) and yscale(log), unless you really want just the y axis or just the x axis logged.

Or consider the *axis_label_options* { y | x }label(,grid), which adds grid lines. Specify maxes(ylabel(,grid)) to add grid lines across, maxes(xlabel(,grid)) to add grid lines vertically, and both options to add grid lines in both directions. When using both, you can specify the maxes() option twice—maxes(ylabel(,grid)) maxes(xlabel(,grid))—or once combined—maxes(ylabel(,grid) xlabel(,grid))—it makes no difference because maxes() is *merged-implicit*; see [G-4] **Concept: repeated options**.

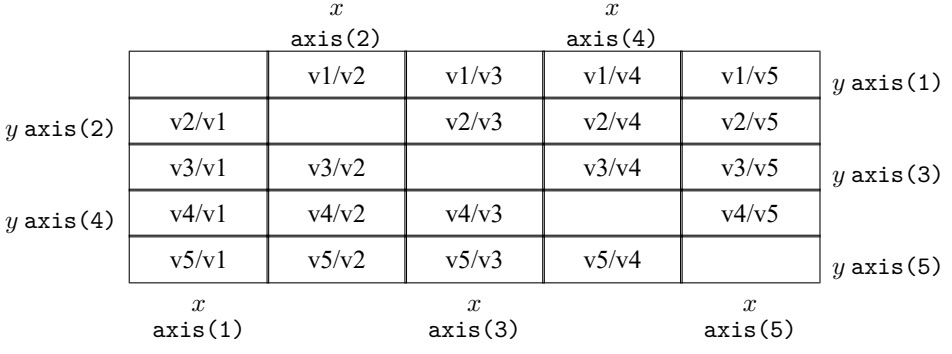
See [G-3] ***axis_scale_options*** and [G-3] ***axis_label_options*** for the suboptions that may appear inside maxes(). In reading those entries, ignore the axis(#) suboption; graph matrix will ignore it if you specify it.

axis_label_options allow you to assert axis-by-axis control over the labeling. Do not confuse this with maxes(*axis_label_options*), which specifies options that affect all the axes. *axis_label_options* specified outside the maxes() option specify options that affect just one of the axes. *axis_label_options* can be repeated for each axis.

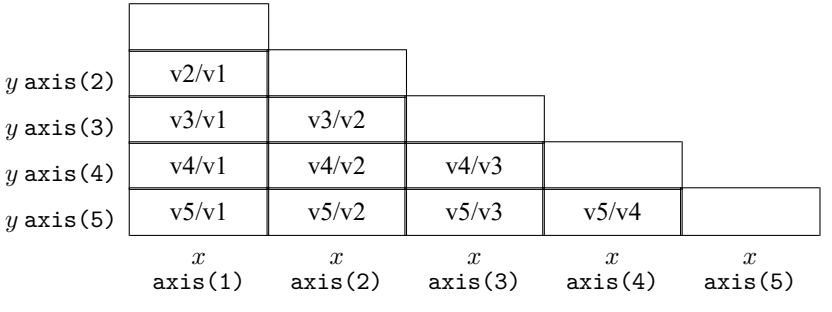
When you specify *axis_label_options* outside maxes(), you must specify the axis-label suboption axis(#). For instance, you might type

```
. graph matrix mpg weight displ, ylabel(0(5)40, axis(1))
```

The effect of that would be to label the specified values on the first y axis (the one appearing on the far right). The axes are numbered as follows:



and if half is specified, the numbering scheme is



See [G-3] ***axis_label_options***; remember to specify the `axis(#)` suboption, and do not specify the `graph matrix` option `maxes()`.

`by(`*varlist*`, ...)` allows drawing multiple graphs for each subgroup of the data. See *Use with by( )* under *Remarks and examples* below, and see [G-3] ***by_option***.

std_options allow you to specify titles (see *Adding titles* under *Remarks and examples* below, and see [G-3] ***title_options***), control the aspect ratio and background shading (see [G-3] ***region_options***), control the overall look of the graph (see [G-3] ***scheme_option***), and save the graph to disk (see [G-3] ***saving_option***).

See [G-3] ***std_options*** for an overview of the standard options.

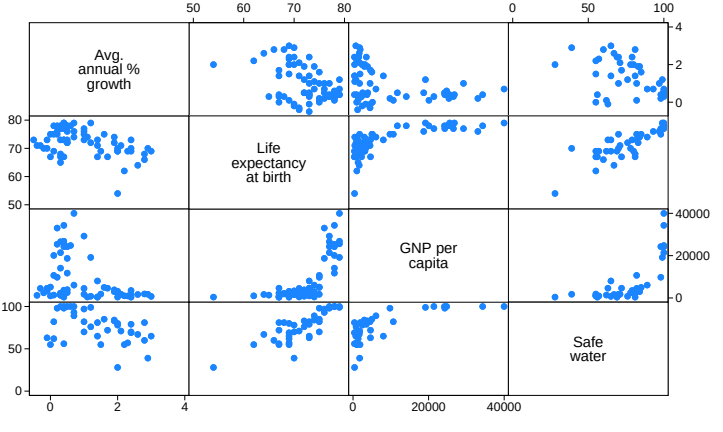
Remarks and examples

Remarks are presented under the following headings:

Typical use

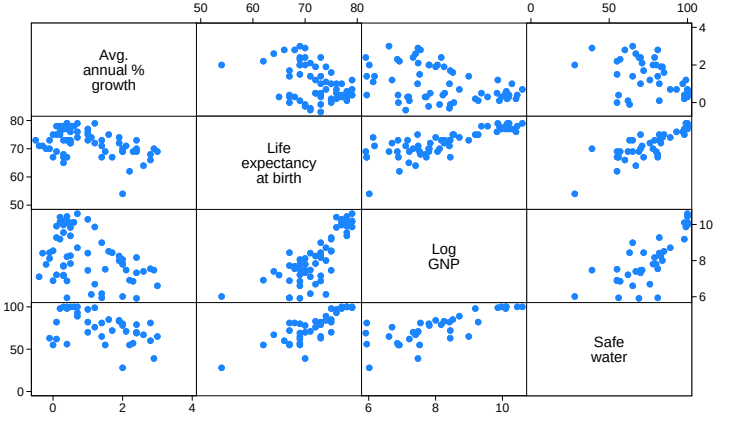
`graph matrix` provides an excellent alternative to correlation matrices (see [R] **correlate**) as a quick way to examine the relationships among variables:

```
. use https://www.stata-press.com/data/r19/lifeexp
(Life expectancy, 1998)
. graph matrix popgrowth-safewater
```



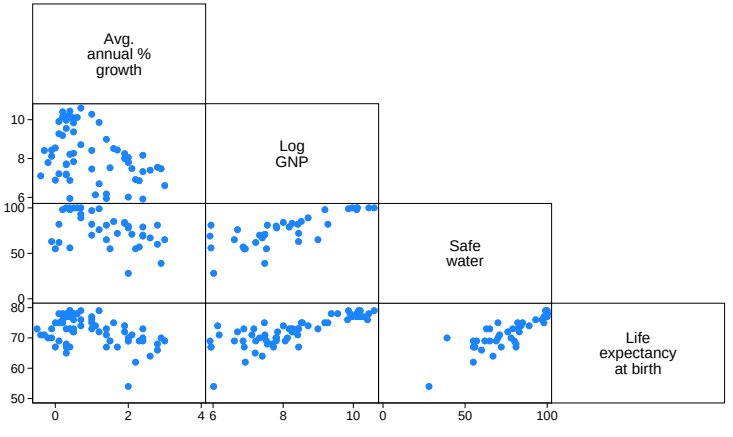
Seeing the above graph, we are tempted to transform gnppc into log units:

```
. generate lgnppc = ln(gnppc)
(5 missing values generated)
. label variable lgnppc "Log GNP"
. graph matrix popgr lexp lgnp safe
```



Some people prefer showing just half the matrix, moving the "dependent" variable to the end of the list:

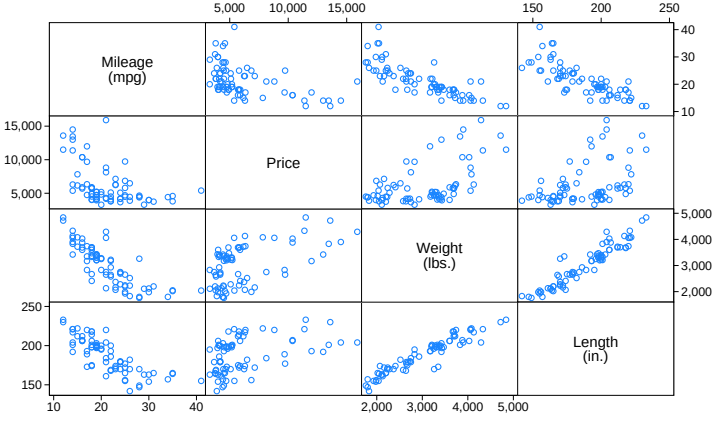
```
. gr matrix popgr lgnp safe lexp, half
```



Marker symbols and the number of observations

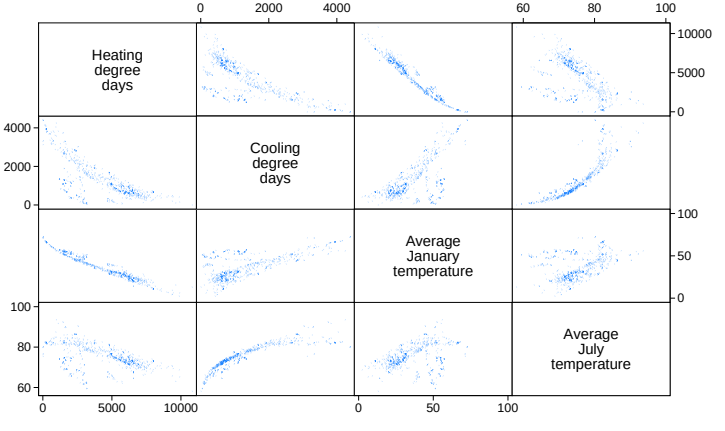
The msymbol() option—abbreviation ms()—allows us to control the marker symbol used; see [G-3] *marker_options*. Hollow symbols sometimes work better as the number of observations increases:

```
. use https://www.stata-press.com/data/r19/auto, clear
(1978 automobile data)
. gr mat mpg price weight length, ms(Oh)
```



Points work best when there are many data:

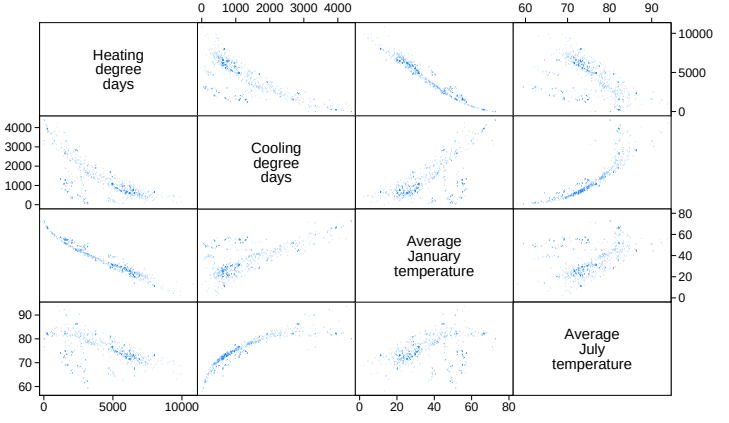
```
. use https://www.stata-press.com/data/r19/citytemp, clear
(City temperature data)
. gr mat heatdd-tempjuly, ms(p)
```



Controlling the axes labeling

By default, approximately three values are labeled and ticked on the y and x axes. When graphing only a few variables, increasing this often works well:

```
. use https://www.stata-press.com/data/r19/citytemp, clear
(City temperature data)
. gr mat heatdd-tempjuly, ms(p) maxes(ylab(#4) xlab(#4))
```

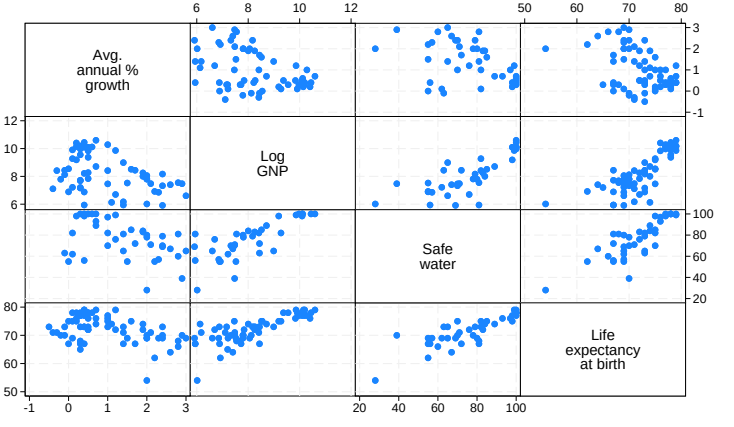


Specifying `#4` does not guarantee four labels; it specifies that approximately four labels be used; see [G-3] ***axis_label_options***. Also see *axis_label_options* under *Options* above for instructions on controlling the axes individually.

Adding grid lines

To add horizontal grid lines, specify `maxes(ylab(,grid))`, and to add vertical grid lines, specify `maxes(xlab(,grid))`. Below we do both and specify that four values be labeled:

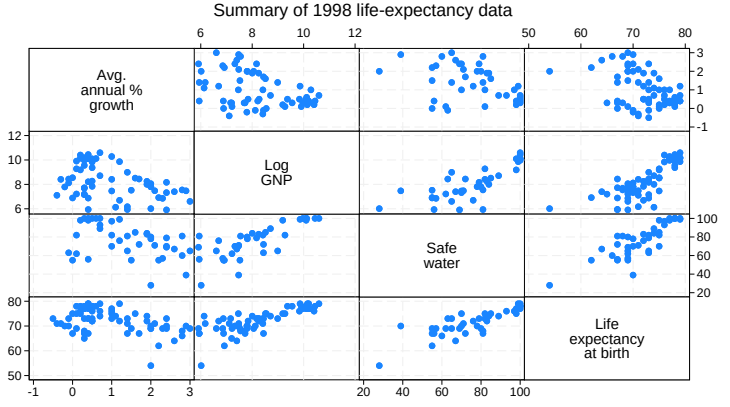
```
. use https://www.stata-press.com/data/r19/lifeexp, clear
(Life expectancy, 1998)
. generate lgnppc = ln(gnppc)
(5 missing values generated)
. label variable lgnppc "Log GNP"
. graph matrix popgr lexp lgnppc safe, maxes(ylab(#4, grid) xlab(#4, grid))
```



Adding titles

The standard title options may be used with graph matrix:

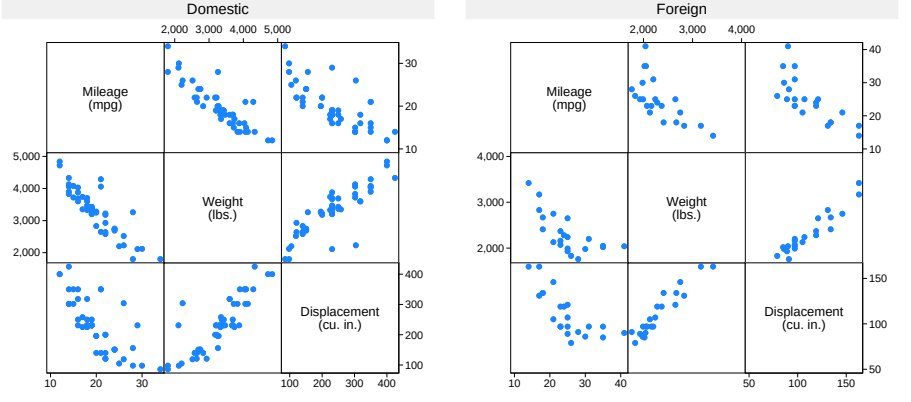
```
. use https://www.stata-press.com/data/r19/lifeexp, clear
(Life expectancy, 1998)
. generate lgnppc = ln(gnppc)
(5 missing values generated)
. label var lgnppc "Log GNP"
. graph matrix popgr lexp lgnp safe, maxes(ylab(#4, grid) xlab(#4, grid))
                      subtitle("Summary of 1998 life-expectancy data")
                      note("Source: The World Bank Group")
```



Use with by()

graph matrix may be used with by():

```
. use https://www.stata-press.com/data/r19/auto, clear
(1978 automobile data)
. gr matrix mpg weight displ, by(foreign) xsize(5)
```



See [G-3] *by_option*.

History

The origin of the scatterplot matrix is unknown, although early written discussions may be found in Hartigan (1975), Tukey and Tukey (1981), and Chambers et al. (1983). The scatterplot matrix has also been called the *draftman's display* and *pairwise scatterplot*. Regardless of the name used, we believe that the first "canned" implementation was by Becker and Chambers in a system called S—see Becker and Chambers (1984)—although S predates 1984. We also believe that Stata provided the second implementation, in 1985.

References

Basford, K. E., and J. W. Tukey. 1998. *Graphical Analysis of Multiresponse Data*. Boca Raton, FL: Chapman and Hall/CRC. https://doi.org/10.1201/9780367811846.

Becker, R. A., and J. M. Chambers. 1984. *S: An Interactive Environment for Data Analysis and Graphics*. Belmont, CA: Wadsworth.

Chambers, J. M., W. S. Cleveland, B. Kleiner, and P. A. Tukey. 1983. *Graphical Methods for Data Analysis*. Belmont, CA: Wadsworth.

Hartigan, J. A. 1975. Printer graphics for clustering. *Journal of Statistical Computation and Simulation* 4: 187–213. https://doi.org/10.1080/00949657508810123.

Tukey, P. A., and J. W. Tukey. 1981. "Preparation; prechosen sequences of views". In *Interpreting Multivariate Data*, edited by V. Barnett, 189–213. Chichester, UK: Wiley.

Also see

[G-2] **graph** — The graph command

[G-2] **graph twoway scatter** — Two-way scatterplots