

Glossary

added line. Added lines are lines added by *added_line_options* that extend across the plot region and perhaps across the plot region's margins. For instance, `yline()` adds horizontal lines at the specified values, whereas `xline()` adds vertical lines at the specified values. See [G-3] [added_line_options](#).

addedlinestyle. *addedlinestyle* specifies the overall look of added lines. The look of the added line is defined by the *linestyle* and whether or not the line extends through the margins of the plot region. See [G-4] [addedlinestyle](#) and [G-4] [linestyle](#).

alignmentstyle. *alignmentstyle* specifies how text is vertically aligned in a textbox, that is, whether text should appear on the baseline of a textbox, near the bottom, in the middle, or at the top. See [G-4] [alignmentstyle](#).

anglestyle. *anglestyle* specifies the angle at which text is to be displayed. The text could be displayed horizontally, vertically, at a 45-degree angle, or at any angle that you desire. See [G-4] [anglestyle](#).

areastyle. *areastyle* determines whether an area is to be outlined and filled, and, if so, in what color. The area might be an entire region, bars, an area under a curve, or other enclosed areas such as boxes in box plots. See [G-4] [areastyle](#).

aspect ratio. The aspect ratio of an object is the ratio of its width to its height. The aspect ratio of an overall Stata graph, also known as the [available area](#), is controlled by specifying options `ysize()` or `xsize()`; see [G-3] [region_options](#). `graph describe` will show you the `ysize()` and `xsize()` for a graph. The aspect ratio of the [plot region](#) is controlled by the `aspect(#)` option; see [G-3] [aspect_option](#).

available area. The available area is defined by the height and width of the overall graph area, as determined by options `ysize(#)` and `xsize(#)`. See [G-3] [region_options](#). Also see [graph region](#) and [plot region](#).

axisstyle. *axisstyle* is a composite style that holds and sets all attributes of an axis, including whether there are ticks, gridlines, and axis lines. Axis styles are used only in scheme files. You would rarely want to change axis styles. See [G-4] [axisstyle](#).

baseline of text. The baseline of text is the imaginary line on which text rests. Letters with descenders, such as p and g, will have the descenders fall beneath the baseline. See [G-4] [alignmentstyle](#).

bitmap image format. See [image format](#), [raster image format](#), and [vector image format](#).

by-graph. A by-graph is one graph that contains an array of separate graphs, each of the same type and each reflecting a different subset of the data. See [G-4] [bystyle](#) and [G-3] [by_option](#).

bystyle. *bystyle* specifies the overall look for by-graphs, including whether individual graphs have their own axes and labels or whether they are shared; whether the axis scales are different for each graph; and how close the graphs are to each other. See [G-4] [bystyle](#).

clockposstyle. *clockposstyle*, an integer from 0 to 12, specifies a location around a central point. *clockposstyle* of 0 is always allowed and refers to the center. See [compassdirstyle](#) for another method for specifying directions. See [G-4] [clockposstyle](#).

CMYK values. See [cyan](#), [magenta](#), [yellow](#), and [blue key or black \(CMYK\) values](#).

color saturation. See [intensity](#).

colorstyle. *colorstyle* sets the color and opacity of graph components such as lines, backgrounds, and bars. See [G-4] [colorstyle](#).

- compassdirstyle.** *compassdirstyle* specifies a direction such as north, neast, or east. See *clockposstyle* for another method for specifying directions. See [G-4] *compassdirstyle*.
- composite style.** See *style, composite style, and detail style*.
- connectstyle.** *connectstyle* specifies if and how points in a scatterplot are to be connected. The most common choice is `connect(1)`, which means to connect with straight lines. See [G-4] *connectstyle*.
- cyan, magenta, yellow, and key or black (CMYK) values.** Cyan, magenta, yellow, and key or black (CMYK) values are used to specify a color. For example, values 0, 0, 255, 0 specify yellow. Also see *red, green, and blue (RGB) values* and *hue, saturation, and value (HSV) values*.
- detail style.** See *style, composite style, and detail style*.
- EMF.** See *Enhanced Metafile*.
- Encapsulated PostScript (EPS).** Encapsulated PostScript (EPS) is a vector image format that is based on PostScript with certain restrictions, created by Adobe. The restrictions allow an EPS file to be included within other PostScript documents. A common use of EPS files is for them to contain a graph that can then be inserted into another document such as a report. An EPS file may contain both graphics and text. See [G-3] *eps_options*. Also see *image format, raster image format, and vector image format*.
- Enhanced Metafile (EMF).** Enhanced Metafile (EMF) was initially released in 1993 by Microsoft Corporation. EMF is primarily a vector format, although it can contain raster formats. EMF is also used as a graphics language for printer drivers and as a clipboard image format. Also see *image format, raster image format, and vector image format*.
- EPS.** See *Encapsulated PostScript (EPS)*.
- family and plottype.** A family describes a group of related graphs. Stata graphs can be categorized into six different families: bar, box, dot, matrix, pie, and twoway. Each family has different plottypes. The twoway family, for instance, has 42 different plottypes, including scatterplots, lineplots, dropline plots, and range plots.
- GIF.** See *Graphics Interchange Format (GIF)*.
- .gph file.** Stata stores graphs in `.gph` files, along with the original data from which the graph was drawn. These `.gph` files are binary files written in a machine-and-operating-system independent format, which means that `.gph` files may be read on a Mac, Windows, or Unix computer, regardless of the type of computer on which it was originally created. See [G-4] **Concept: gph files**.
- graph region.** The graph region is typically larger than the plot region but smaller than the available area. The graph region includes the plot region and the area where the titles and the axis labels occur. See the *image of regions* in [G-3] *region_options*. Also see *available area* and *plot region*.
- Graphics Interchange Format (GIF).** Graphics Interchange Format (GIF) is a type of raster image format, developed by CompuServe in 1987. See [G-3] *gif_options*. Also see *image format, raster image format, and vector image format*.
- grids.** Grids are lines that extend from an axis across the *plot region*.
- gridstyle.** *gridstyle* specifies the overall look of grids, such as whether the grid lines extend into the plot region's margin. See [G-4] *gridstyle*.
- HSV values.** See *hue, saturation, and value (HSV) values*.
- hue, saturation, and value (HSV) values.** Hue, saturation, and value (HSV) values are used to specify a color. For example, values 0, 0, 1 specify white. Also see *cyan, magenta, yellow, and key or black (CMYK) values* and *red, green, and blue (RGB) values*.

image format, raster image format, and vector image format. An image format is a way of storing a digital image. A raster image format, also known as a bitmap image format, describes the image in terms of pixels. A vector image format describes the image in terms of its individual components such as lines, points, and text. Examples of raster image formats are [GIF](#), [JPEG](#), [PNG](#), and [TIFF](#). Examples of vector image formats are [EPS](#), [PDF](#), [PS](#), and [SVG](#).

intensity. Color intensity is the brightness of a color. As intensity decreases, the color is paler. Another term for intensity of color is color saturation. See [\[G-4\] *colorstyle*](#) for how to adjust intensity.

intensitystyle. *intensitystyle* specifies the intensity of colors as a percentage from 0 to 100. For example, 0% means no color at all; 100% means full color. *intensitystyle* is used primarily in scheme files. See [\[G-4\] *intensitystyle*](#).

Joint Photographic Experts Group (JPEG). Joint Photographic Experts Group (JPEG) is a type of raster image format, created in 1992. See [\[G-3\] *jpg_options*](#). Also see [image format](#), [raster image format](#), and [vector image format](#).

JPEG. See [Joint Photographic Experts Group \(JPEG\)](#).

justificationstyle. *justificationstyle* specifies how text is horizontally aligned in a textbox. Choices include left-justified, centered, or right-justified. See [\[G-4\] *justificationstyle*](#).

legend and legend key. A legend is a table that shows the symbols used in a graph along with the text describing their meaning. A legend key is a symbol and the descriptive text describing the symbol. Stata has a standard legend, a `contourline` plot legend, and a `contour` plot legend, controlled by `legend()`, `plegend()`, and `clegend()`, respectively. See [\[G-3\] *legend_options*](#) and [\[G-3\] *legend_option*](#).

legendstyle. *legendstyle* defines the look of a legend, which is defined by 14 attributes, including the number of columns or rows of the table, the gaps between lines and columns, and the margin around the legend. See [\[G-4\] *legendstyle*](#).

linealignmentstyle. *linealignmentstyle* specifies the alignment of a border or outline for markers, fill areas, bars, and boxes. The line can be placed inside the outline, outside the outline, or centered on the outline. See [\[G-4\] *linealignmentstyle*](#).

linepatternstyle. *linepatternstyle* specifies the pattern of a line, such as solid line, dashed line, or dotted line. See [\[G-4\] *linepatternstyle*](#).

linestyle. sets the overall pattern, thickness, color and opacity, and alignment of a line. See [\[G-4\] *linestyle*](#).

linewidthstyle. *linewidthstyle* specifies the thickness of lines, from so thin as to be invisible to very thick. See [\[G-4\] *linewidthstyle*](#).

marginstyle. *marginstyle* specifies where to include margins (bottom, top, left, right) as well as their size (from no margin to very large). See [\[G-4\] *marginstyle*](#).

marker. A marker is the ink used to mark where points are on a plot. The overall look of a marker is determined by the *markerstyle*, which consists of the symbol used (*symbolstyle*), the size of the marker (*markersizestyle*), and the color of the marker (*colorstyle*). See [\[G-4\] *markerstyle*](#), [\[G-4\] *symbolstyle*](#), [\[G-4\] *markersizestyle*](#), and [\[G-4\] *colorstyle*](#). Also see [markerstyle](#), [symbolstyle](#), [markersizestyle](#), and [colorstyle](#).

markerlabelstyle. *markerlabelstyle* specifies the overall look of marker labels by defining the position, gap, angle, size, and color of the marker label. See [\[G-4\] *markerlabelstyle*](#).

markersizestyle. *markersizestyle* determines the size of markers, which ranges from tiny to huge. You can make the marker any size that you want; see [\[G-4\] *size*](#). See [\[G-4\] *markersizestyle*](#). Also see [marker](#) and [symbolstyle](#).

markerstyle. *markerstyle* defines the five attributes of a marker: the shape of the marker; the size of the marker; the color and opacity of the marker; the interior color and opacity of the marker; and the overall style, thickness, color and opacity, and alignment of the marker's outline. See [G-4] *markerstyle*.

merged-explicit option. An option allowed with a graph command is called merged-explicit if it is treated as rightmost unless a suboption is specified. The option `title()` is a merged-explicit option. If you do not specify any suboptions within option `title()`, then the rightmost rule applies. For example, *graph_command ... , title(hello) title(goodbye)* will result in the single title of "goodbye" on the graph. If, however, you were to specify *graph_command ... , title(hello) title(goodbye, suffix)*, then the title of "hello goodbye" would appear on the graph. Each merged-explicit option will have documentation on what the merge options are and how they work. See [G-4] **Concept: repeated options**.

merged-implicit option. An option allowed with a graph command is called merged-implicit if repeated instances of the same option within a graph command are merged. Examples of merged-implicit options are `yline()` and `xline()` for adding horizontal and vertical lines, respectively, to a graph. See [G-4] **Concept: repeated options**.

object. Graph objects are the building blocks of Stata graphs: plot regions, axes, legends, notes, captions, titles, subtitles, and positional titles. Objects may themselves contain other objects. A plot region may contain several different plots, each of which are objects. A legend contains a key region, which is itself an object.

opacity and transparency. Opacity is the percentage of a color that covers the background color. Opacity varies between 0% (color has no coverage) to 100% (color fully hides the background). The inverse of opacity is transparency, so 0% opacity means completely transparent and 100% opacity means not transparent. See [G-4] *colorstyle* for information on how to adjust opacity.

orientationstyle. *orientationstyle* specifies whether textboxes are horizontal or vertical. See [G-4] *orientationstyle*.

PDF. See *Portable Document Format (PDF)*.

plot region. The plot region is the area enclosed by the axes. It is where data are plotted. This is not to be confused with the *graph region*, which encompasses the plot region and the area with the titles and axes labels. See the *image of regions* in [G-3] *region_options*. Also see *available area* and *graph region*.

plotregionstyle. *plotregionstyle* controls the overall look of a plot region and is defined by four sets of attributes: *marginstyle*, *overall areastyle*, *internal areastyle*, and horizontal and vertical positioning of the plot region. See [G-4] *plotregionstyle*.

plotype. See *family and plotype*.

PNG. See *Portable Network Graphics (PNG)*.

Portable Document Format (PDF). Portable Document Format (PDF) is a vector file format that can be used to represent graphics and documents independent of hardware, software, or operating system. It was developed by Adobe and initially released in 1993. PDF was standardized (ISO) as an open format in 2008. Also see *image format*, *raster image format*, and *vector image format*.

Portable Network Graphics (PNG). Portable Network Graphics (PNG) is a raster image format that is nonpatented and was originally released in 1996. See [G-3] *png_options*. Also see *image format*, *raster image format*, and *vector image format*.

PostScript (PS). PostScript (PS) is a vector file format that is actually its own language. It was originally created by Adobe Systems in 1984 as a page description language with hopes that it could be used by any printer. Although not all printers support PostScript, it has become a widely

used format supported by many printers, screen rendering programs, and graphic design programs. See [G-3] *ps_options*. Also see *Encapsulated PostScript (EPS)* and *image format, raster image format, and vector image format*.

PS. See *PostScript (PS)*.

pstyle. *pstyle* specifies the overall look of a plot, which is defined by the look of markers, marker labels, and lines (markerstyle, markerlabelstyle, linestyle); how points are connected by lines (connectstyle); whether missing values cause lines to be broken when connected; the way areas are filled, colored, shaded, and outlined (areastyle); the look of dots in dot plots; and the look of arrowheads. See [G-4] *pstyle*.

raster image format. See *image format, raster image format, and vector image format*.

red, green, and blue (RGB) values. Red, green, and blue (RGB) values are a triplet of numbers, each of which specifies, on a scale of 0–255, the amount of red, green, and blue to be mixed. For example, to obtain the following colors, you would use these values:

red	=	255	0	0
green	=	0	255	0
blue	=	0	0	255
white	=	255	255	255
black	=	0	0	0
gray	=	128	128	128
navy	=	26	71	111

The overall scale of the triplet affects intensity, so changing 255 to 128 would keep the color the same but make it dimmer. Also see *cyan, magenta, yellow, and key or black (CMYK) values* and *hue, saturation, and value (HSV) values*.

repeated option. A repeated option is an option to a graph command that can be specified more than once. It is unusual for Stata to allow a repeated option for a command, but graph commands are an exception because so many other commands are implemented in terms of graph commands. Repeated options are either *rightmost* or *merged*. *rightmost* instructs Stata to take the rightmost occurrence; *merged* specifies to merge repeated instances together. See [G-4] **Concept: repeated options**.

RGB values. See *red, green, and blue (RGB) values*.

rightmost option. An option allowed with a graph command is called *rightmost* if an option is repeated within the same graph command, and all but the rightmost occurrence of the option are ignored. `msymbol()` is an example of a rightmost option. See [G-4] **Concept: repeated options**. Also see *unique option, merged-implicit option, and merged-explicit option*.

ringposstyle. *ringposstyle*, a number from 0 to 100, specifies the distance from the plot region for titles, subtitles, etc. Positioning of titles is controlled by options `position(clockposstyle)` and `ring(ringposstyle)`. The option `position()` specifies a direction according to the hours of a clock, whereas the option `ring()` controls the distance of the title from the plot region. If two titles were specified at the same `position()`, the title with the larger `ring()` would be the furthest from the plot region. See [G-4] *ringposstyle*.

saturation. See *intensity*.

Scalable Vector Graphics (SVG). Scalable Vector Graphics (SVG) is a vector image format based on XML. SVG is an open standard developed by the World Wide Web Consortium and initially released in 2001. See [G-3] *svg_options*. Also see *image format, raster image format, and vector image format*.

scheme. A scheme specifies the overall look of a graph, which includes everything from foreground color and background color to whether the y axis appears on the left or the right. A scheme is a set of defaults. Stata's default setting is scheme `s2color`. Type `graph query, schemes` to see the list of schemes installed on your computer. See *Examples of schemes* in [G-4] [Schemes intro](#).

shadestyle. `shadestyle` sets the color ([G-4] [colorstyle](#)) and intensity ([G-4] [intensitystyle](#)) of the color for a filled area. See [G-4] [shadestyle](#).

style, composite style, and detail style. `style` specifies a composite of related option settings that together determine the overall look of the graph. Styles come in two forms: detail styles and composite styles. Detail styles specify precisely how an attribute of something looks, and composite styles specify an overall look in terms of detail-style values.

Examples of composite style options are `mstyle(symbolstyle)`, `mlabstyle(markerlabelstyle)`, and `lstyle(linestyle)`, which specify the overall look of symbols, marker labels, and lines, respectively. Note that composite style options all end in "style": `mstyle()`, `mlabstyle()`, and `lstyle()`.

Examples of detail styles are `mcolor(colorstyle)`, `mlwidth(linewidthstyle)`, and `mlabsize(textsizestyle)`, which specify the color of an object, the outline thickness, and the size of the label, respectively. Note that the option names do not end in the word `style`: `mcolor()`, `mlwidth()`, and `mlabsize()`.

SVG. See *Scalable Vector Graphics (SVG)*.

symbolstyle. The `symbolstyle` determines the shape of markers. Choices include circles, diamonds, triangles, squares (these four can be different sizes and hollow or not), plus signs, X's (these two can be only different sizes), dots and even an invisible symbol. See [G-4] [symbolstyle](#). Also see [marker](#) and [markersizestyle](#).

Tagged Image File Format (TIFF). Tagged Image File Format (TIFF) is a raster graphics image format created by Aldus Corporation in 1986 and subsequently updated by Adobe Systems after Adobe acquired Aldus. See [G-3] [tif_options](#). Also see [image format](#), [raster image format](#), and [vector image format](#).

textbox. A textbox is one or more lines of text with an optional border around it. Also see [textboxstyle](#).

textboxstyle. Textboxes are defined by 11 attributes such as whether the textbox is vertical or horizontal ([orientationstyle](#)); the size of the text ([textsizestyle](#)), the color of the text ([colorstyle](#)), and whether the text is left-justified, centered, or right-justified ([justificationstyle](#)). See [G-4] [textboxstyle](#). Also see [orientationstyle](#), [colorstyle](#), and [justificationstyle](#).

textsizestyle. `textsizestyle` specifies the size of text. See [G-4] [textsizestyle](#) for choices for the size of text.

textstyle. `textstyle` specifies the overall look of text, which is defined by five attributes: whether the text is vertical or horizontal ([orientationstyle](#)); the size of the text ([textsizestyle](#)); the color of the text ([colorstyle](#)); whether the text is left-justified, centered, or right-justified ([justificationstyle](#)); and how the text aligns with the baseline ([alignmentstyle](#)). See [G-4] [textstyle](#). Also see [orientationstyle](#), [textsizestyle](#), [colorstyle](#), [symbolstyle](#), [justificationstyle](#), and [alignmentstyle](#).

ticks. Ticks are the marks that appear on axes. See [G-4] [tickstyle](#).

ticksetstyle. `ticksetstyle` specifies the overall look of axis ticks, which is determined by the `tickstyle`, the `gridstyle` (if a grid is associated with the tickset), and other tick details. See [G-4] [ticksetstyle](#). Also see [tickstyle](#) and [gridstyle](#).

tickstyle. `tickstyle` specifies the overall look of axis ticks and their labels. Ticks are defined by three attributes: the length of the tick; the line style of the tick; and whether the tick extends out, extends

in, or crosses the axis. Tick labels are defined by two attributes: their size and color. *tickstyle* also determines the gap between the tick and the tick label. See [G-4] *tickstyle*.

TIFF. See *Tagged Image File Format (TIFF)*.

transparency. See *opacity and transparency*.

twoway graph. Twoway graphs are graphs with two dimensions; they display the relationship between the variables in the numeric y and x axes. See [G-2] **graph twoway**.

unique option. An option allowed with a graph command is called unique if the option may be specified only once. `saving()` is an example of a unique option. Also see *rightmost option*, *merged-implicit option*, and *merged-explicit option*.

vector image format. See *image format, raster image format, and vector image format*.