

graph twoway dot — Twoway dot plots

[Syntax](#) [Menu](#) [Description](#) [Options](#)
 Remarks and examples Reference Also see

Syntax

```
twoway dot yvar xvar [if] [in] [, options]
```

<i>options</i>	Description
<code>vertical</code>	vertical bar plot; the default
<code>horizontal</code>	horizontal bar plot
<code>dotextend(yes no)</code>	dots extend beyond point
<code>base(#)</code>	value to drop to if <code>dotextend(no)</code>
<code>ndots(#)</code>	# of dots in full span of <i>y</i> or <i>x</i>
<code>dstyle(markerstyle)</code>	overall marker style of dots
<code>dsymbol(symbolstyle)</code>	marker symbol for dots
<code>dcolor(colorstyle)</code>	fill and outline color for dots
<code>dfcolor(colorstyle)</code>	fill color for dots
<code>dsize(markersizestyle)</code>	size of dots
<code>dlstyle(linestyle)</code>	overall outline style of dots
<code>dlcolor(colorstyle)</code>	outline color for dots
<code>dlwidth(linewidthstyle)</code>	thickness of outline for dots
<i>scatter_options</i>	any options other than <i>connect_options</i> documented in [G-2] graph twoway scatter

See [\[G-4\] markerstyle](#), [\[G-4\] symbolstyle](#), [\[G-4\] colorstyle](#), [\[G-4\] markersizestyle](#), [\[G-4\] linestyle](#), and [\[G-4\] linewidthstyle](#).

All options are *rightmost*, except `vertical` and `horizontal`, which are *unique*; see [\[G-4\] concept: repeated options](#).

Menu

Graphics > Twoway graph (scatter, line, etc.)

Description

`twoway dot` displays numeric (*y,x*) data as dot plots. Also see [\[G-2\] graph dot](#) to create dot plots of categorical variables. `twoway dot` is useful in programming contexts.

Options

`vertical` and `horizontal` specify either a vertical or a horizontal dot plot. `vertical` is the default.

If `horizontal` is specified, the values recorded in `yvar` are treated as x values, and the values recorded in `xvar` are treated as y values. That is, to make horizontal plots, do not switch the order of the two variables specified.

In the `vertical` case, dots are drawn at the specified `xvar` values and extend up and down.

In the `horizontal` case, lines are drawn at the specified `xvar` values and extend left and right.

`dotextend(yes|no)` determines whether the dots extend beyond the y value (or x value if `horizontal` is specified). `dotextend(yes)` is the default.

`base(#)` is relevant only if `dotextend(no)` is also specified. `base()` specifies the value from which the dots are to extend. The default is `base(0)`.

`ndots(#)` specifies the number of dots across a line; `ndots(75)` is the default. Depending on printer/screen resolution, using fewer or more dots can make the graph look better.

`dstyle(markerstyle)` specifies the overall look of the markers used to create the dots, including their shape and color. The other options listed below allow you to change their attributes, but `dstyle()` provides the starting point.

You need not specify `dstyle()` just because there is something you want to change. You specify `dstyle()` when another style exists that is exactly what you desire or when another style would allow you to specify fewer changes to obtain what you want.

See [G-4] [markerstyle](#) for a list of available marker styles.

`dsymbol(symbolstyle)` specifies the shape of the marker used for the dot. See [G-4] [symbolstyle](#) for a list of symbol choices, although it really makes little sense to change the shape of dots; else why would it be called a dot plot?

`dcolor(colorstyle)` specifies the color of the symbol used for the dot. See [G-4] [colorstyle](#) for a list of color choices.

`dfcolor(colorstyle)`, `dsize(markersizestyle)`, `dlstyle(linestyle)`, `dlcolor(colorstyle)`, and `dlinewidth(linewidthstyle)` are rarely (never) specified options. They control, respectively, the fill color, size, outline style, outline color, outline width, and, if you are really using dots, dots are affected by none of these things. For these options to be useful, you must also specify `dsymbol()`; as we said earlier, why then would it be called a dot plot? In any case, see [G-4] [colorstyle](#), [G-4] [markersizestyle](#), [G-4] [linestyle](#), and [G-4] [linewidthstyle](#) for a list of choices.

`scatter_options` refer to any of the options allowed by `scatter`, and most especially the `marker_options`, which control how the marker (not the dot) appears. `connect_options`, even if specified, are ignored. See [G-2] [graph twoway scatter](#).

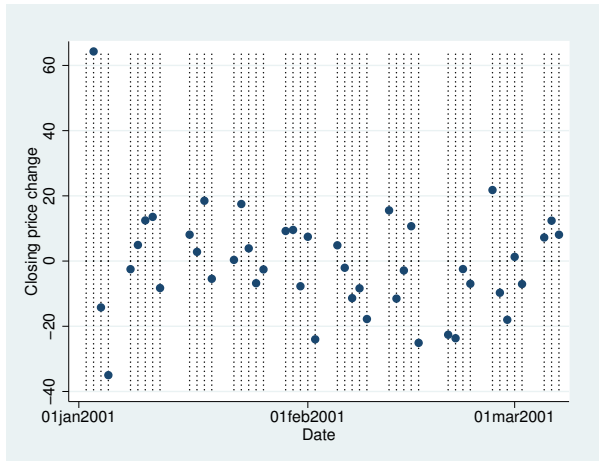
Remarks and examples

[stata.com](http://www.stata.com)

`twoway dot` is of little, if any use. We cannot think of a use for it, but perhaps someday, somewhere, someone will. We have nothing against the dot plot used with categorical data—see [G-2] [graph dot](#) for a useful command—but using the dot plot in a twoway context would be bizarre. It is nonetheless included for logical completeness.

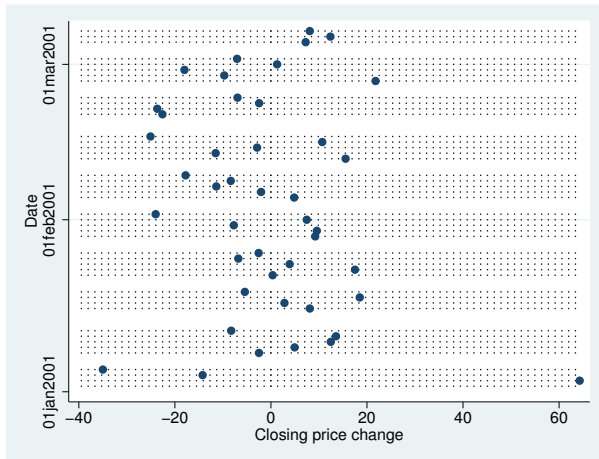
In [G-2] [graph twoway bar](#), we graphed the change in the value for the S&P 500. Here are a few of that data graphed as a dot plot:

```
. use http://www.stata-press.com/data/r13/sp500
(S&P 500)
. twoway dot change date in 1/45
```



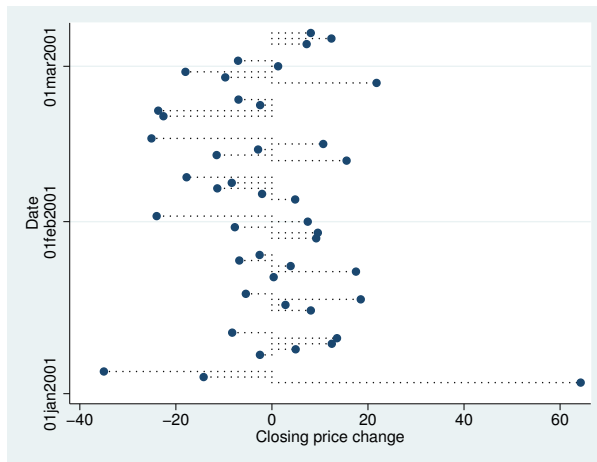
Dot plots are usually presented horizontally,

```
. twoway dot change date in 1/45, horizontal
```



and below we specify the `dotextend(n)` option to prevent the dots from extending across the range of x :

```
. twoway dot change date in 1/45, horizontal dotext(n)
```



Reference

Cox, N. J. 2008. [Speaking Stata: Between tables and graphs](#). *Stata Journal* 8: 269–289.

Also see

[G-2] [graph twoway scatter](#) — Twoway scatterplots

[G-2] [graph dot](#) — Dot charts (summary statistics)