

## graph twoway function — Twoway line plot of function

[Description](#)  
[Options](#)

[Quick start](#)  
[Remarks and examples](#)

[Menu](#)  
[Reference](#)

[Syntax](#)  
[Also see](#)

## Description

twoway function plots  $y = f(x)$ , where  $f(x)$  is some function of  $x$ . That is, you type

```
. twoway function y=sqrt(x)
```

It makes no difference whether  $y$  and  $x$  are variables in your data.

## Quick start

Graph the function  $\ln\{x/(1-x)\}$  from 0 to 1

```
twoway function y = ln(x/(1-x))
```

Same as above

```
twoway function y = logit(x)
```

Graph the function  $y = x^2$  from  $-1$  to  $1$

```
twoway function y = x^2, range(-1 1)
```

As above, but as a horizontal graph

```
twoway function y = x^2, range(-1 1) horizontal
```

Probability density function of Student's  $t$  distribution with 4 degrees of freedom

```
twoway function y = tden(4,x), range(-4 4)
```

As above, but add the normal probability density function

```
twoway function y = tden(4,x), range(-4 4) || ///
function y = normalden(x), range(-4 4)
```

Add a legend

```
twoway function y = tden(4,x), range(-4 4) || ///
function y = normalden(x), range(-4 4) ///
legend(label(1 "t density with 4 df") ///
label(2 "Normal density"))
```

Normal probability density function with lines from the curve to 0 at  $-1.96$  and  $1.96$

```
twoway function y = normalden(x), range(-4 4) dropline(-1.96 1.96)
```

## Menu

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

## Syntax

```
twoway function [[y] =] f(x) [if] [in] [, options]
```

<i>options</i>	Description
<code>range(# #)</code>	plot over $x = \#$ to $\#$
<code>range(varname)</code>	plot over $x = \text{min}$ to $\text{max}$ of <i>varname</i>
<code>n(#)</code>	evaluate at $\#$ points; default is 300
<code>droplines(numlist)</code>	draw lines to axis at specified $x$ values
<code>base(#)</code>	base value for <code>dropline()</code> ; default is 0
<code>horizontal</code>	draw plot horizontally
<code>yvarformat(%fmt)</code>	display format for $y$
<code>xvarformat(%fmt)</code>	display format for $x$
<i>cline_options</i>	change look of plotted line
<i>axis_choice_options</i>	associate plot with alternative axis
<i>twoway_options</i>	titles, legends, axes, added lines and text, by, regions, name, aspect ratio, etc.

All explicit options are *rightmost*, except `horizontal`, which is *unique*; see [G-4] **concept: repeated options**.

if *exp* and *in* *range* play no role unless option `range(varname)` is specified.

In the above syntax diagram,  $f(x)$  stands for an *expression* in terms of  $x$ .

## Options

`range(# #)` and `range(varname)` specify the range of values for  $x$ . In the first syntax, `range()` is a pair of numbers identifying the minimum and maximum. In the second syntax, `range()` is a variable name, and the range used will be obtained from the minimum and maximum values of the variable. If `range()` is not specified, `range(0 1)` is assumed.

`n(#)` specifies the number of points at which  $f(x)$  is to be evaluated. The default is `n(300)`.

`droplines(numlist)` adds dropped lines from the function down to, or up to, the axis (or  $y = \text{base}()$  if `base()` is specified) at each  $x$  value specified in *numlist*.

`base(#)` specifies the base for the `droplines()`. The default is `base(0)`. This option does not affect the range of the axes, so you may also want to specify the *axis\_scale\_option* `yscale(range(#))` as well; see [G-3] *axis\_scale\_options*.

`horizontal` specifies that the roles of  $y$  and  $x$  be interchanged and that the graph be plotted horizontally rather than vertically (that the plotted function be reflected along the identity line).

`yvarformat(%fmt)` and `xvarformat(%fmt)` specify the display format to be used for  $y$  and  $x$ . These formats are used when labeling the axes; see [G-3] *axis\_label\_options*.

*cline\_options* specify how the function line is rendered; see [G-3] *cline\_options*.

*axis\_choice\_options* associate the plot with a particular  $y$  or  $x$  axis on the graph; see [G-3] *axis\_choice\_options*.

*twoway\_options* are a set of common options supported by all *twoway* graphs. These options allow you to title graphs, name graphs, control axes and legends, add lines and text, set aspect ratios, create graphs over by() groups, and change some advanced settings. See [G-3] *twoway\_options*.

## Remarks and examples

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

Remarks are presented under the following headings:

*Typical use*

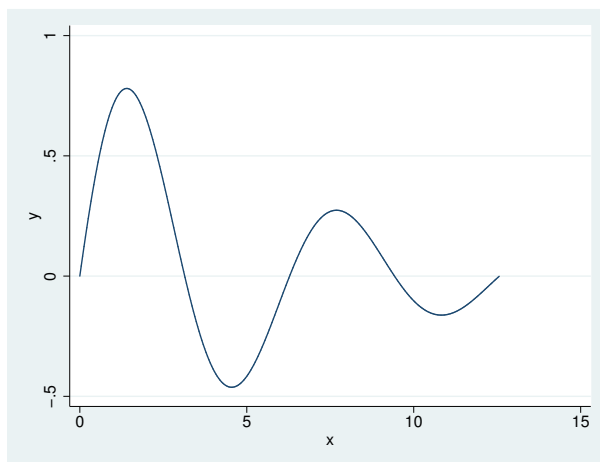
*Advanced use 1*

*Advanced use 2*

### Typical use

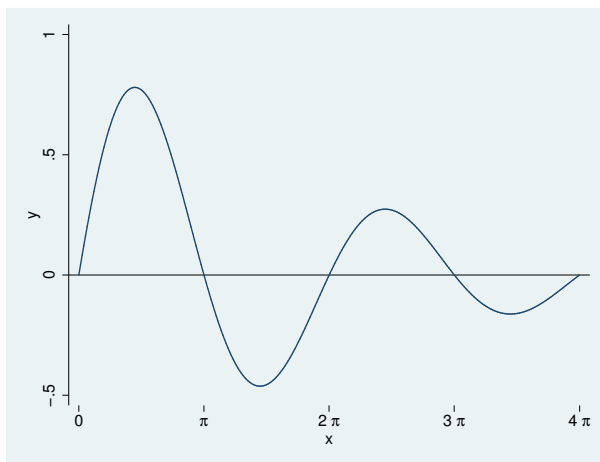
You wish to plot the function  $y = \exp(-x/6)\sin(x)$  over the range 0 to  $4\pi$ :

```
. twoway function y=exp(-x/6)*sin(x), range(0 12.57)
```



A better rendition of the graph above is

```
. twoway function y=exp(-x/6)*sin(x), range(0 12.57)
  yline(0, lstyle(foreground))
  xlabel(0 3.14 "{&pi}" 6.28 "2{&pi}" 9.42 "3{&pi}" 12.57 "4{&pi}")
  plotregion(style(none))
  xsca(noline)
```



`yline(0, lstyle(foreground))` added a line at  $y = 0$ ; `lstyle(foreground)` gave the line the same style as used for the axes. See [G-3] [added\\_line\\_options](#).

`xlabel(0 3.14 "{&pi}" 6.28 "2{&pi}" 9.42 "3{&pi}" 12.57 "4{&pi}")` labeled the  $x$  axis with the numeric values given; see [G-3] [axis\\_label\\_options](#).

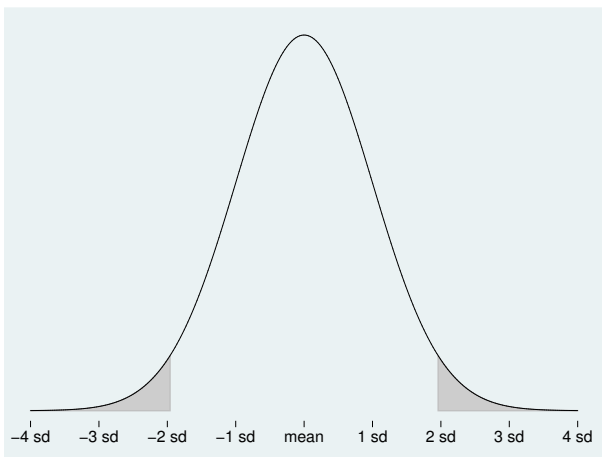
`plotregion(style(none))` suppressed the border around the plot region; see [G-3] [region\\_options](#).

`xsca(noline)` suppressed the drawing of the  $x$ -axis line; see [G-3] [axis\\_scale\\_options](#).

## Advanced use 1

The following graph appears in many introductory textbooks:

```
. twoway
  function y=normalden(x), range(-4 -1.96) color(gs12) recast(area)
  || function y=normalden(x), range(1.96 4) color(gs12) recast(area)
  || function y=normalden(x), range(-4 4) lstyle(foreground)
  ||,
  plotregion(style(none))
  ysca(off) xsca(noline)
  legend(off)
  xlabel(-4 "-4 sd" -3 "-3 sd" -2 "-2 sd" -1 "-1 sd" 0 "mean"
        1 "1 sd" 2 "2 sd" 3 "3 sd" 4 "4 sd"
        , grid gmin gmax)
  xtitle("")
```

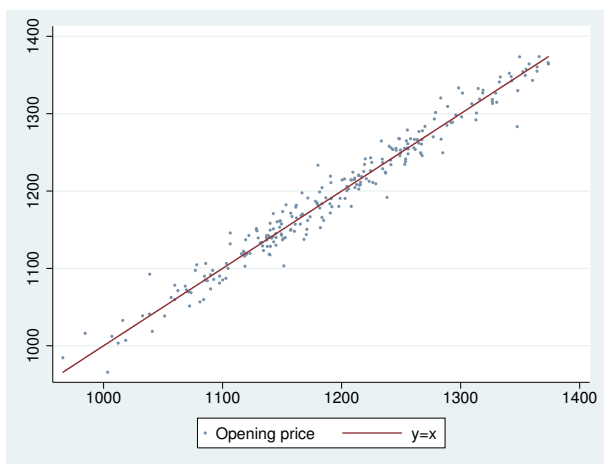


We drew the graph in three parts: the shaded area on the left, the shaded area on the right, and then the overall function. To obtain the shaded areas, we used the *advanced\_option* `recast(area)` so that, rather than the function being plotted by `graph twoway line`, it was plotted by `graph twoway area`; see [G-3] *advanced\_options* and [G-2] *graph twoway area*. Concerning the overall function, we drew it last so that its darker foreground-colored line would not get covered up by the shaded areas.

## Advanced use 2

`function` plots may be overlaid with other `twoway` plots. For instance, `function` is one way to add  $y = x$  lines to a plot:

```
. use http://www.stata-press.com/data/r14/sp500, clear
(S&P 500)
. scatter open close, msize(*.25) mcolor(*.6) ||
  function y=x, range(close) yvarlab("y=x") clwidth(*1.5)
```



In the above, we specified the *advanced\_option* `yvarlab("y=x")` so that the variable label of  $y$  would be treated as “ $y=x$ ” in the construction of the legend; see [G-3] *advanced\_options*. We specified

`msize(*.25)` to make the marker symbols smaller, and we specified `mcolor(*.6)` to make them dimmer; see [G-4] *relativesize* and [G-4] *colorstyle*.

## Reference

Cox, N. J. 2004. *Stata tip 15: Function graphs on the fly*. *Stata Journal* 4: 488–489.

## Also see

[G-2] `graph twoway line` — Twoway line plots