

Description

Some commands that draw graphs (but do not start with the word `graph`) are documented in the other reference manuals. Many of those commands allow the `addplot()` option. This option allows them to overlay their results on top of `graph twoway` plots; see [G-2] **graph twoway**.

Syntax

```
command ... [ , ... addplot(plot ... [ || plot ... [...] ] [ , below] ) ... ]
```

where *plot* may be any subcommand of `graph twoway` (see [G-2] **graph twoway**), such as `scatter`, `line`, or `histogram`.

Option

`addplot(plots [, below])` specifies the rest of the `graph twoway` subcommands to be added to the `graph twoway` command issued by *command*.

below is a suboption of the `addplot()` option and specifies that the added plots be drawn before the plots drawn by the command. Thus the added plots will appear below the plots drawn by *command*. The default is to draw the added plots after the command's plots so that they appear above the command's plots. *below* affects only the added plots that are drawn on the same *x* and *y* axes as the command's plots.

Remarks and examples

Remarks are presented under the following headings:

Commands that allow the `addplot()` option

Advantage of `graph twoway` commands

Advantages of graphic commands implemented outside `graph twoway`

Use of the `addplot()` option

Commands that allow the `addplot()` option

`graph` commands never allow the `addplot()` option. The `addplot()` option is allowed by commands outside `graph` that are implemented in terms of `graph twoway`.

For instance, the `histogram` command—see [R] **histogram**—allows `addplot()`. `graph twoway histogram`—see [G-2] **graph twoway histogram**—does not.

Advantage of graph twoway commands

The advantage of graph twoway commands is that they can be overlaid, one on top of the other. For instance, you can type

```
. graph twoway scatter yvar xvar || lfit yvar xvar
```

and the separate graphs produced, scatter and lfit, are combined. The variables to which each refers need not even be the same:

```
. graph twoway scatter yvar xvar || lfit y2var x2var
```

Advantages of graphic commands implemented outside graph twoway

Graphic commands implemented outside graph twoway can have simpler syntax. For instance, the histogram command has an option, normal, that will overlay a normal curve on top of the histogram:

```
. histogram myvar, normal
```

That is easier than typing

```
. summarize myvar
. graph twoway histogram myvar ||
    function normalden(x,'r(mean)','r(sd)'), range(myvar)
```

which is the graph twoway way of producing the same thing.

Thus the tradeoff between graph and nograph commands is one of greater flexibility versus easier use.

Use of the addplot() option

The addplot() option attempts to give back flexibility to nograph graphic commands. Such commands are, in fact, implemented in terms of graph twoway. For instance, when you type

```
. histogram ...
```

or you type

```
. sts graph ...
```

the result is that those commands construct a complicated graph twoway command

```
→ graph twoway something_complicated
```

and then run that for you. When you specify the addplot() option, such as in

```
. histogram ..., addplot(your_contribution)
```

or

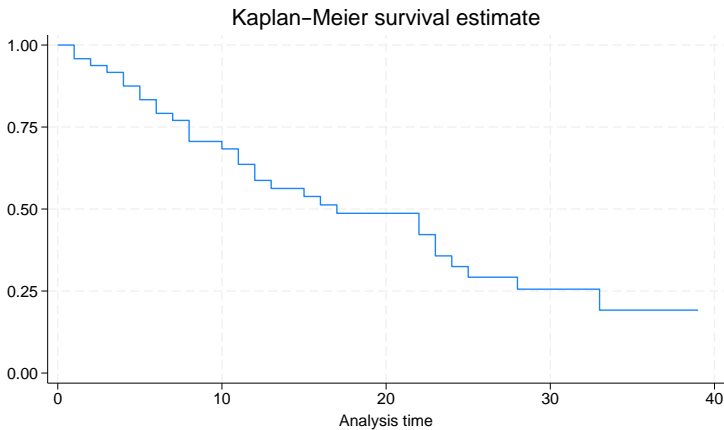
```
. sts graph, addplot(your_contribution)
```

the result is that the commands construct

```
→ graph twoway something_complicated || your_contribution
```

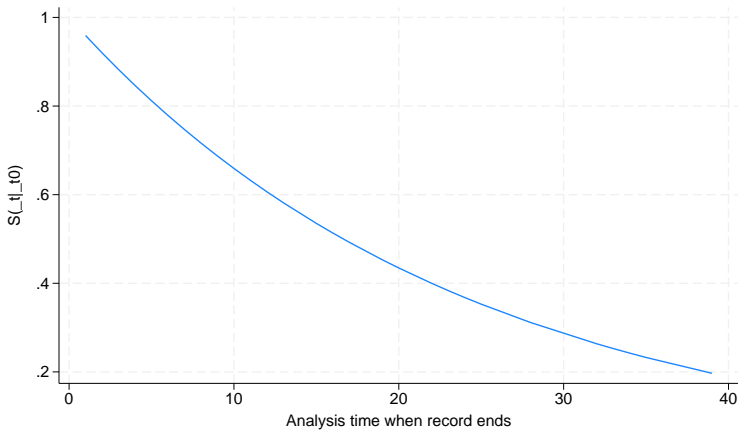
Let us assume that you have survival data and wish to visually compare the Kaplan–Meier (that is, the empirical survivor function) with the function that would be predicted if the survival times were assumed to be exponentially distributed. Simply typing

```
. use https://www.stata-press.com/data/r19/cancer, clear
(Patient survival in drug trial)
. quietly stset studytime, fail(died)
. sts graph
```



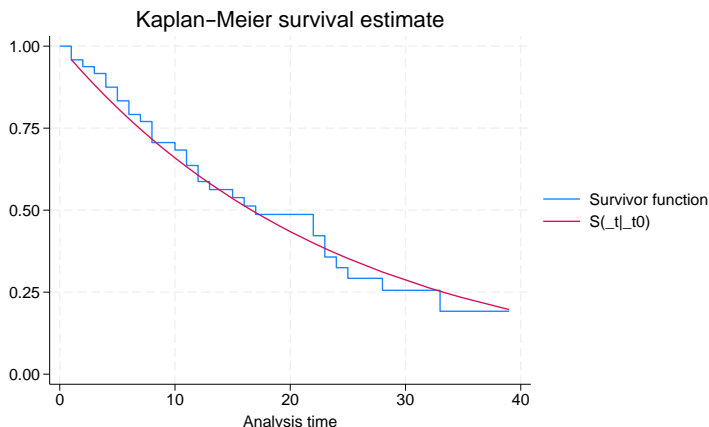
will obtain a graph of the empirical estimate. To obtain the exponential estimate, you might type

```
. quietly streg, distribution(exponential)
. predict S, surv
. graph twoway line S _t, sort
```



To put these two graphs together, you can type

```
. sts graph, addplot(line S_t, sort)
```



The result is just as if you typed

```
. sts graph || line S_t, sort
```

if only that were allowed.

Reference

Jann, B. 2015. [A note on adding objects to an existing twoway graph](#). *Stata Journal* 15: 751–755.

Also see

[G-2] [graph twoway](#) — Two-way graphs

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).