

Title

sts graph — Graph the survivor and cumulative hazard functions

Syntax

`sts graph [if] [in] [, options]`

<i>options</i>	description
Main	
<u>survival</u>	graph Kaplan–Meier survivor function; the default
<u>failure</u>	graph Kaplan–Meier failure function
<u>cumhaz</u>	graph Nelson–Aalen cumulative hazard function
<u>hazard</u>	graph smoothed hazard estimate
by(<i>varlist</i>)	calculate separately on different groups of <i>varlist</i>
<u>adjustfor</u> (<i>varlist</i>)	adjust the estimates to zero values of <i>varlist</i>
<u>strata</u> (<i>varlist</i>)	stratify on different groups of <i>varlist</i>
<u>separate</u>	show curves on separate graphs; default is to show curves one on top of another
ci	show pointwise confidence bands
At-risk table	
<u>risktable</u>	show table of number at risk beneath graph
<u>risktable</u> (<i>risk_spec</i>)	show customized table of number at risk beneath graph
Options	
<u>level</u> (#)	set confidence level; default is <code>level(95)</code>
<u>per</u> (#)	units to be used in reported rates
<u>noshow</u>	do not show st setting information
<u>tmax</u> (#)	show graph for $t \leq \#$
<u>tmin</u> (#)	show graph for $t \geq \#$
<u>noorigin</u>	begin survival (failure) curve at first exit time; default is to begin at $t = 0$
<u>width</u> (# [<i>#...</i>])	override default bandwidth(s)
<u>kernel</u> (<i>kernel</i>)	kernel function; use with <code>hazard</code>
<u>noboundary</u>	no boundary correction; use with <code>hazard</code>
<u>lost</u>	show number lost
<u>enter</u>	show number entered and number lost
<u>atrisk</u>	show numbers at risk at beginning of each interval
<u>censored</u> (<u>single</u>)	show one hash mark at each censoring time, no matter what number is censored
<u>censored</u> (<u>number</u>)	show one hash mark at each censoring time and number censored above hash mark
<u>censored</u> (<u>multiple</u>)	show multiple hash marks for multiple censoring at the same time
<u>censopts</u> (<i>hash_options</i>)	affect rendition of hash marks
<u>lostopts</u> (<i>marker_label_options</i>)	affect rendition of numbers lost
<u>atriskopts</u> (<i>marker_label_options</i>)	affect rendition of numbers at risk

Plot	
<u>plotopts</u> (<i>cline_options</i>)	affect rendition of the plotted lines
<u>plot#opts</u> (<i>cline_options</i>)	affect rendition of the #th plotted line; may not be combined with <code>separate</code>
CI plot	
<u>ciopts</u> (<i>area_options</i>)	affect rendition of the confidence bands
<u>ci#opts</u> (<i>area_options</i>)	affect rendition of the #th confidence band; may not be combined with <code>separate</code>
Add plots	
<code>addplot</code> (<i>plot</i>)	add other plots to the generated graph
Y axis, X axis, Titles, Legend, Overall	
<i>twoway_options</i>	any options documented in [G] <i>twoway_options</i>
<u>byopts</u> (<i>byopts</i>)	how subgraphs are combined, labeled, etc.
<hr/>	
See [R] <code>kdensity</code> for information on <i>kernel</i> , and see [G] <i>marker_label_options</i> , [G] <i>cline_options</i> , and [G] <i>area_options</i> .	

where *risk_spec* is

`[numlist] [, table_options group(group)]`

numlist specifies the points at which the number at risk is to be evaluated, *table_options* customizes the table of number at risk, and `group(group)` specifies a specific group/row for *table_options* to be applied.

<i>table_options</i>	description
<hr/>	
Main	
<i>axis_label_options</i>	control table by using axis labeling options; seldom used
<code>order</code> (<i>order_spec</i>)	select which rows appear and their order
<u>righttitles</u>	place titles on right side of the table
<u>failevents</u>	show number failed in the at-risk table
<i>text_options</i>	affect rendition of table elements and titles
Row titles	
<u>rowtitle</u> (<code>[text] [, rtext_options]</code>)	change title for a row
Title	
<code>title</code> (<code>[text] [, ttext_options]</code>)	change overall table title
<hr/>	
See [G] <i>axis_label_options</i> .	

where *order_spec* is

`# ["text" ["text" ...]] [...]`

<i>text_options</i>	description
<code>size(textsizestyle)</code>	size of text
<code>color(colorstyle)</code>	color of text
<code>justification(justificationstyle)</code>	text left-justified, centered, or right-justified
<code>format(%fmt)</code>	format values per <code>%fmt</code>
<code>topgap(relativesize)</code>	margin above rows
<code>bottomgap(relativesize)</code>	margin beneath rows
<code>† style(textstyle)</code>	overall style of text

`† style()` does not appear in the dialog box.

See [G] `textsizestyle`, [G] `colorstyle`, [G] `justificationstyle`, [G] `relativesize`, and [G] `textstyle`.

<i>rtext_options</i>	description
<code>size(textsizestyle)</code>	size of text
<code>color(colorstyle)</code>	color of text
<code>justification(justificationstyle)</code>	text left-justified, centered, or right-justified
<code>at(#)</code>	override <i>x</i> position of titles
<code>topgap(relativesize)</code>	margin above rows
<code>† style(textstyle)</code>	overall style of text

`† style()` does not appear in the dialog box.

See [G] `textsizestyle`, [G] `colorstyle`, [G] `justificationstyle`, [G] `relativesize`, and [G] `textstyle`.

<i>ttext_options</i>	description
<code>size(textsizestyle)</code>	size of text
<code>color(colorstyle)</code>	color of text
<code>justification(justificationstyle)</code>	text left-justified, centered, or right-justified
<code>at(#)</code>	override <i>x</i> position of titles
<code>topgap(relativesize)</code>	margin above rows
<code>bottomgap(relativesize)</code>	margin beneath rows
<code>† style(textstyle)</code>	overall style of text

`† style()` does not appear in the dialog box.

See [G] `textsizestyle`, [G] `colorstyle`, [G] `justificationstyle`, [G] `relativesize`, and [G] `textstyle`.

<i>group</i>	description
<code>#rownum</code>	specify group by row number in table
<code>value</code>	specify group by value of group
<code>label</code>	specify group by text of value label associated with group

<i>hash_options</i>	description
<i>line_options</i>	change look of dropped lines
<i>marker_label_options</i>	add marker labels

See [G] `line_options` and [G] `marker_label_options`, except for `mlabel()`.

`risktable()` may be repeated and is *merged-explicit*; see [G] **concept: repeated options**.
You must `stset` your data before using `sts graph`; see [ST] `stset`.
`fweights`, `iweights`, and `pweights` may be specified using `stset`; see [ST] `stset`.

Menu

Statistics > Survival analysis > Graphs > Survivor and cumulative hazard functions

Description

`sts graph` graphs the estimated survivor (failure) function, the Nelson–Aalen estimated cumulative (integrated) hazard function, or the estimated hazard function. See [ST] `sts` for an introduction to this command.

`sts graph` can be used with single- or multiple-record or single- or multiple-failure `st` data.

Options

Main

`survival`, `failure`, `cumhaz`, and `hazard` specify the function to graph.

`survival` specifies that the Kaplan–Meier survivor function be plotted. This option is the default if a function is not specified.

`failure` specifies that the Kaplan–Meier failure function, $1 - S(t + 0)$, be plotted.

`cumhaz` specifies that the Nelson–Aalen estimate of the cumulative hazard function be plotted.

`hazard` specifies that an estimate of the hazard function be plotted. This estimate is calculated as a weighted kernel-density estimate using the estimated hazard contributions, $\Delta \hat{H}(t_j) = \hat{H}(t_j) - \hat{H}(t_{j-1})$. These hazard contributions are the same as those obtained by `sts generate newvar = h`.

`by(varlist)` produces separate survivor, cumulative hazard, or hazard functions by making separate calculations for each group identified by equal values of the variables in `varlist` and plots all the groups on one graph. `by()` may not be combined with `strata()`.

If you have more than one `by()` variable but need only one, use `egen` to create it; see [D] `egen`.

`adjustfor(varlist)` adjusts the estimate of the survivor or hazard functions to that for 0 values of `varlist`. If you want to adjust the function to values different from 0, you need to center the variables around those values before issuing the command. Say that you want to plot the survivor function adjusted to age of patients and the ages in your sample are 40–60 years. Then

```
. sts graph, adjustfor(age)
```

will graph the survivor function adjusted to age 0. If you want to adjust the function to age 40, type

```
. gen age40 = age - 40
```

```
. sts graph, adjustfor(age40)
```

`adjustfor()` is not available with `cumhaz` or `ci`.

If you specify `adjustfor()` with `by()`, `sts` fits separate Cox regression models for each group, using the `adjustfor()` variables as covariates. The separately calculated baseline survivor functions are then retrieved.

If you specify `adjustfor()` with `strata()`, `sts` fits a stratified-on-group Cox regression model using the `adjustfor()` variables as covariates. The stratified, baseline survivor function is then retrieved.

`strata(varlist)` produces estimates of the survivor (failure) or hazard functions stratified on variables in `varlist` and plots all the groups on one graph. It requires specifying `adjustfor()` and may not be combined with `by()`.

If you have more than one `strata()` variable but need only one, use `egen` to create it; see [D] **egen**.

`separate` is meaningful only with `by()` or `strata()`; it requests that each group be placed on its own graph rather than one on top of the other. Sometimes curves have to be placed on separate graphs—such as when you specify `ci`—because otherwise it would be too confusing.

`ci` includes pointwise confidence bands. The default is not to produce these bands. `ci` is not allowed with `adjustfor()` or `pweights`.

At-risk table

`risktable`(`([numlist][, table_options])`) displays a table showing the number at risk beneath the plot. `risktable` may not be used with `separate` or `adjustfor()`.

`risktable` displays the table in the default format with number at risk shown for each time reported on the *x* axis.

`risktable`(`[numlist][, table_options]`) specifies that the number at risk be evaluated at the points specified in `numlist` or that the rendition of the table be changed by `table_options`.

There are two ways to change the points at which the numbers at risk are evaluated.

1. The *x* axis of the graph may be altered. For example:

```
. sts graph, xlabel(0(5)40) risktable
```

2. A `numlist` can be specified directly in the `risktable()` option, which affects only the at-risk table. For example:

```
. sts graph, risktable(0(5)40)
```

The two examples produce the same at-risk table, but the first also changes the time labels on the graph's *x* axis.

`table_options` affect the rendition of the at-risk table and may be any of the following:

`group(#rownum | value | label)` specifies that all the suboptions specified in the `risktable()` apply only to the specified group. Because the `risktable()` option may be repeated, this option allows different rows of the at-risk table to be displayed with different colors, font sizes, etc.

When both a value and a value label are matched, the value label takes precedence.

`risktable()` may be specified with or without the `group()` suboption. When specified without `group()`, each suboption is applied to all available groups or rows. `risktable()` specified without `group()` is considered to be global and is itself merged-explicit. See [G] **concept: repeated options** for more information on how repeated options are merged.

Consider the following example:

```
. sts graph, by(drug) risktable(, color(red) size(small))
> risktable(, color(navy))
```

The example above would produce a table where all rows are colored navy with small text.

Combining global `risktable()` options with group-specific `risktable()` options can be useful. When global options are combined with group-specific options, group-specific options always take precedence.

Consider the following example:

```
. sts graph, by(drug) risktable(, color(navy))
> risktable(, color(red) group(#1))
```

The example above would produce a table with the first row colored red and all remaining rows colored navy.

Main

`axis_label_options` control the table by using axis labeling options. These options are seldom used. See [G] [axis_label_options](#).

`order()` specifies which and in what order rows are to appear in the at-risk table. Optionally, `order()` can be used to override the default text.

`order(## # ...)` is the syntax used for identifying which rows to display and their order. `order(1 2 3)` would specify that row 1 is to appear first in the table, followed by row 2, followed by row 3. `order(1 2 3)` is the default if there are three groups. If there were four groups, `order(1 2 3 4)` would be the default, and so on. If there were four groups and you specified `order(1 2 3)`, the fourth row would not appear in the at-risk table. If you specified `order(2 1 3)`, row 2 would appear first, followed by row 1, followed by row 3.

`order(# "text" # "text" ...)` is the syntax used for specifying the row order and alternate row titles.

Consider the following at-risk table:

drug = 1	20	8	2	
drug = 2	14	10	4	1
drug = 3	14	13	10	5

Specifying `order(1 "Placebo" 3 2)` would produce

Placebo	20	8	2	
drug = 3	14	13	10	5
drug = 2	14	10	4	1

and specifying `order(1 "Placebo" 3 "Drug 2" 2 "Drug 1")` would produce

Placebo	20	8	2	
Drug 2	14	13	10	5
Drug 1	14	10	4	1

`righttitles` specifies that row titles be placed to the right of the at-risk values. The default is to place row titles to the left of the at-risk values.

`failevents` specifies that the number of failure events be shown in parentheses, after the time in which the risk values were calculated.

`text_options` affect the rendition of both row titles and number at risk and may be any of the following:

`size(textsizestyle)` specifies the size of text.

`color(colorstyle)` specifies the color of text.

`justification(justificationstyle)` specifies how text elements are to be justified.

`format(%fmt)` specifies how numeric values are to be formatted.

`topgap(relativesize)` specifies how much space is to be placed above each row.

`bottomgap(relativesize)` specifies how much space is to be placed beneath each row.

`style(textstyle)` specifies the style of text. This option does not appear in the dialog box.

Row titles

`rowtitle([text][, rtext_options])` changes the default text or rendition of row titles. Specifying `rowtitle(, color(navy))` would change the color of all row titles to navy.

`rowtitle()` is often combined with `group()` to change the text or rendition of a title. Specifying `rowtitle(Placebo) group(#2)` would change the title of the second row to Placebo. Specifying `rowtitle(, color(red)) group(#3)` would change the color of the row title for the third row to red.

Row titles may include more than one line. Lines are specified one after the other, each enclosed in double quotes. Specifying `rowtitle("Experimental drug") group(#1)` would produce a one-line row title, and specifying `rowtitle("Experimental" "Drug") group(#1)` would produce a multiple-line row title.

`rtext_options` affect the rendition of both row titles and number at risk and may be any of the following:

`size(textsizestyle)` specifies the size of text.

`color(colorstyle)` specifies the color of text.

`justification(justificationstyle)` specifies how text elements are to be justified.

`at(#)` allows you to reposition row titles or the overall table title to align with a specific location on the x axis.

`topgap(relativesize)` specifies how much space is to be placed above each row.

`style(textstyle)` specifies the style of text. This option does not appear on the dialog box.

Title

`title([title][, ttext_options])` may be used to override the default title for the at-risk table and affect the rendition of its text.

Titles may include one line of text or multiple lines. `title("At-risk table")` will produce a one-line title, and `title("At-risk" "table")` will produce a multiple-line title.

`ttext_options` affect the rendition of both row titles and number at risk and may be any of the following:

`size(textsizestyle)` specifies the size of text.

`color(colorstyle)` specifies the color of text.

`justification(justificationstyle)` specifies how text elements are to be justified.

`at(#)` allows you to reposition row titles or the overall table title to align with a specific location on the x axis.

`at(rowtitles)` places the overall table title at the default position calculated for the row titles. This option is sometimes useful for alignment when the default justification has not been used.

`topgap(relativesize)` specifies how much space is to be placed above each row.

`bottomgap(relativesize)` specifies how much space is to be placed beneath each row.

`style(textstyle)` specifies the style of text. This option does not appear on the dialog box.

Options

`level(#)` specifies the confidence level, as a percentage, for the pointwise confidence interval around the survivor, failure, or cumulative hazard function; see [U] **20.7 Specifying the width of confidence intervals**.

`per(#)` specifies the units used to report the survival or failure rates. For example, if the analysis time is in years, specifying `per(100)` results in rates per 100 person-years.

`noshow` prevents `sts graph` from showing the key `st` variables. This option is seldom used because most people type `stset`, `show` or `stset`, `noshow` to set whether they want to see these variables mentioned at the top of the output of every `st` command; see [ST] **stset**.

`tmax(#)` specifies that the plotted curve be graphed only for $t \leq \#$. This option does not affect the calculation of the function, rather the portion that is displayed.

`tmin(#)` specifies that the plotted curve be graphed only for $t \geq \#$. This option does not affect the calculation of the function, rather the portion that is displayed.

`noorigin` requests that the plot of the survival (failure) curve begin at the first exit time instead of beginning at $t = 0$ (the default). This option is ignored when `cumhaz` or `hazard` is specified.

`width(# [#...])` is for use with `hazard` and specifies the bandwidth to be used in the kernel smooth used to plot the estimated hazard function. If `width()` is not specified, a default bandwidth is used as described in [R] **kdensity**. If it is used with `by()`, multiple bandwidths may be specified, one for each group. If there are more groups than the k bandwidths specified, the default bandwidth is used for the $k + 1, \dots$ remaining groups. If any bandwidth is specified as `.` (dot), the default bandwidth is used for that group.

`kernel(kernel)` is for use with `hazard` and specifies the kernel function to be used in calculating the weighted kernel-density estimate required to produce a smoothed hazard-function estimator. The default kernel is Epanechnikov, yet *kernel* may be any of the kernels supported by `kdensity`; see [R] **kdensity**.

`noboundary` is for use with `hazard`. It specifies that no boundary-bias adjustments are to be made when calculating the smoothed hazard-function estimator. By default, the smoothed hazards are adjusted near the boundaries. If the `epan2`, `biweight`, or `rectangular` kernel is used, the bias correction near the boundary is performed using boundary kernels. For other kernels, the plotted range of the smoothed hazard function is restricted to be within one bandwidth of each endpoint. For these other kernels, specifying `noboundary` merely removes this range restriction.

`lost` specifies that the numbers lost be shown on the plot. These numbers are shown as small numbers over the flat parts of the function.

If `enter` is not specified, the numbers displayed are the number censored minus the number who enter. If you do specify `enter`, the numbers displayed are the pure number censored. The underlying logic is described in [ST] `sts`.

`lost` may not be used with `hazard`.

`enter` specifies that the number who enter be shown on the graph, as well as the number lost. The number who enter are shown as small numbers beneath the flat parts of the plotted function.

`enter` may not be used with `hazard`.

`atrisk` specifies that the numbers at risk at the beginning of each interval be shown on the plot. The numbers at risk are shown as small numbers beneath the flat parts of the plotted function.

`atrisk` may not be used with `hazard`.

`censored(single | number | multiple)` specifies that hash marks be placed on the graph to indicate censored observations.

`censored(single)` places one hash mark at each censoring time, regardless of the number of censorings at that time.

`censored(number)` places one hash mark at each censoring time and displays the number of censorings about the hash mark.

`censored(multiple)` places multiple hash marks for multiple censorings at the same time. For instance, if 3 observations are censored at time 5, three hash marks are placed at time 5. `censored(multiple)` is intended for use when there are few censored observations; if there are too many censored observations, the graph can look bad. In such cases, we recommend that `censored(number)` be used.

`censored()` may not be used with `hazard`.

`censopts(hash_options)` specifies options that affect how the hash marks for censored observations are rendered; see [G] `line_options`. When combined with `censored(number)`, `censopts()` also specifies how the count of censoring is rendered; see [G] `marker_label_options`, except `mlabel()` is not allowed.

`lostopts(marker_label_options)` specifies options that affect how the numbers lost are rendered; see [G] `marker_label_options`. This option implies the `lost` option.

`atriskopts(marker_label_options)` specifies options that affect how the numbers at risk are rendered; see [G] `marker_label_options`. This option implies the `atrisk` option.

Plot

`plotopts(cline_options)` affects the rendition of the plotted lines; see [G] `cline_options`. This option may not be combined with `by(varlist)` or `strata(varlist)`, unless `separate` is also specified.

`plot#opts(cline_options)` affects the rendition of the #th plotted line; see [G] `cline_options`. This option may not be combined with `separate`.

CI plot

`ciopts(area_options)` affects the rendition of the confidence bands; see [G] `area_options`. This option may not be combined with `by(varlist)` or `strata(varlist)`, unless `separate` is also specified.

`ci#opts(area_options)` affects the rendition of the #th confidence band; see [G] `area_options`. This option may not be combined with `separate`.

Add plots

`addplot(plot)` provides a way to add other plots to the generated graph; see [G] *addplot_option*.

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G] *twoway_options*. These include options for titling the graph (see [G] *title_options*) and for saving the graph to disk (see [G] *saving_option*).

`byopts(byopts)` affects the appearance of the combined graph when `by()` or `adjustfor()` is specified, including the overall graph title and the organization of subgraphs. `byopts()` may not be specified with `separate`. See [G] *by_option*.

Remarks

Remarks are presented under the following headings:

- Including the number lost on the graph*
- Graphing the Nelson–Aalen cumulative hazard function*
- Graphing the hazard function*
- Adding an at-risk table*
- On boundary bias for smoothed hazards*

If you have not read [ST] **sts**, please do so.

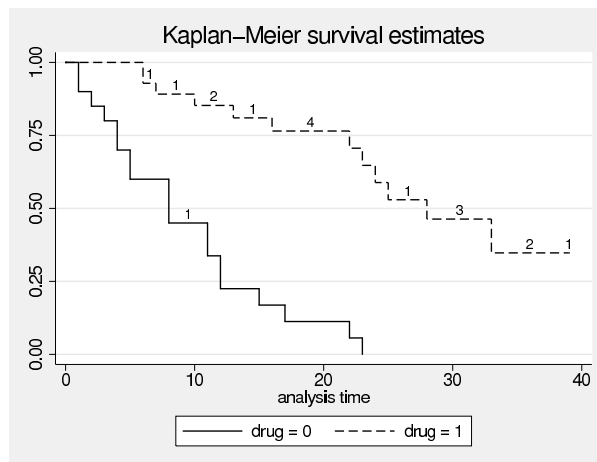
By default, **sts graph** displays the Kaplan–Meier product-limit estimate of the survivor (failure) function. Only one of **sts graph**'s options, `adjustfor()`, modifies the calculation. All the other options merely determine how the results of the calculation are graphed.

We demonstrate many of **sts graph**'s features in [ST] **sts**. This discussion picks up where that entry leaves off.

Including the number lost on the graph

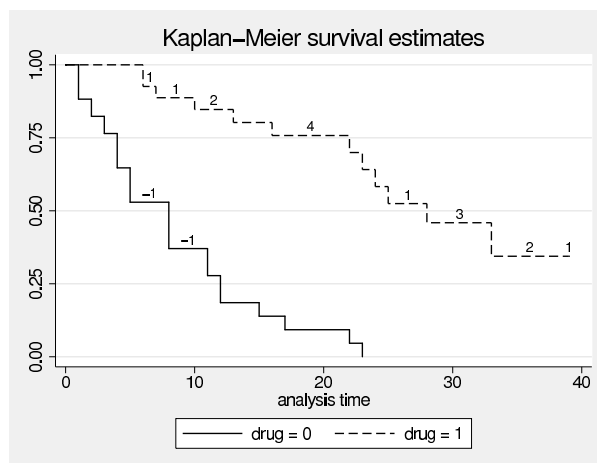
In *Adjusted estimates* in [ST] **sts**, we introduced a simple drug-trial dataset with 1 observation per subject. Here is a graph of the survivor functions, by drug, including the number lost because of censoring:

```
. use http://www.stata-press.com/data/r11/drug2
(Patient Survival in Drug Trial)
. sts graph, by(drug) lost
      failure _d: died
      analysis time _t: studytime
```



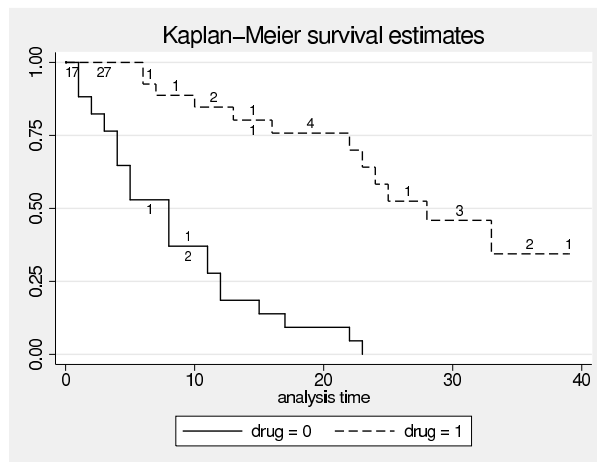
There is no late entry in these data, so we modify the data so that a few subjects entered late. Here is the same graph on the modified data:

```
. use http://www.stata-press.com/data/r11/drug2b
(Patient Survival in Drug Trial)
. sts graph, by(drug) lost
      failure _d: died
      analysis time _t: studytime
```



Note the negative numbers. These occur because, by default, `lost` means censored minus entered. Here `-1` means that 1 entered, or 2 entered and 1 was lost, etc. If we specify the `enter` option, we will see the censored and entered separately:

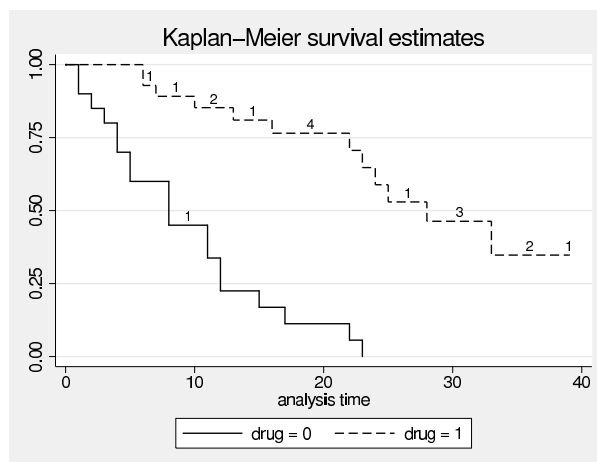
```
. sts graph, by(drug) lost enter
      failure _d: died
      analysis time _t: studytime
```



Although it might appear that specifying `enter` with `lost` is a good idea, that is not always true.

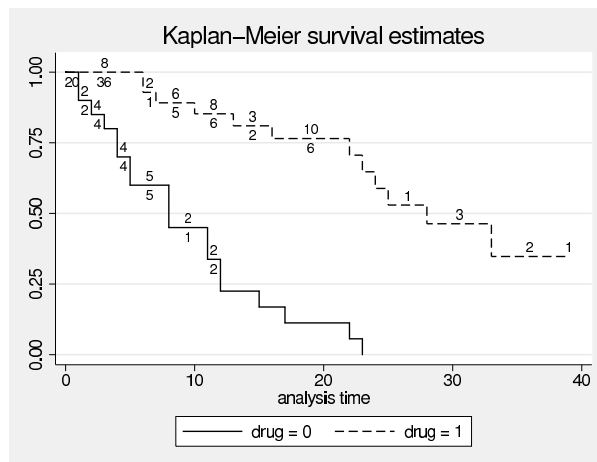
We have yet another version of the data—the correct data not adjusted to have late entry—but in this version we have multiple records per subject. The data are the same, but where there was one record in the first dataset, sometimes there are now two because we have a covariate that is changing over time. From this dataset, here is the graph with the number lost shown:

```
. use http://www.stata-press.com/data/r11/drug2c
(Patient Survival in Drug Trial)
. sts graph, by(drug) lost
      failure _d: died
      analysis time _t: studytime
      id: id
```



This looks just like the first graph we presented, as indeed it should. Again we emphasize that the data are logically, if not physically, equivalent. If, however, we graph the number lost and entered, we get a graph showing a lot of activity:

```
. sts graph, by(drug) lost enter
      failure _d: died
      analysis time _t: studytime
      id: id
```



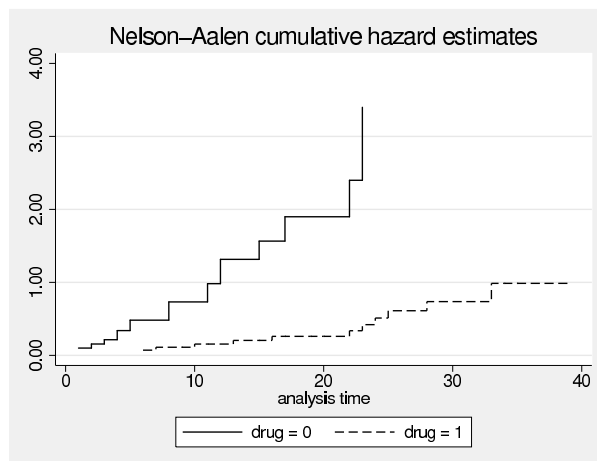
All that activity goes by the name *thrashing*—subjects are being censored to enter the data again, but with different covariates. This graph was better when we did not specify `enter` because the censored-minus-entered calculation smoothed out the thrashing.

Graphing the Nelson–Aalen cumulative hazard function

We can plot the Nelson–Aalen estimate of the cumulative (integrated) hazard function by specifying the `cumbaz` option. For example, from the 1-observation-per-subject drug-trial dataset, here is a graph of the cumulative hazard functions by drug:

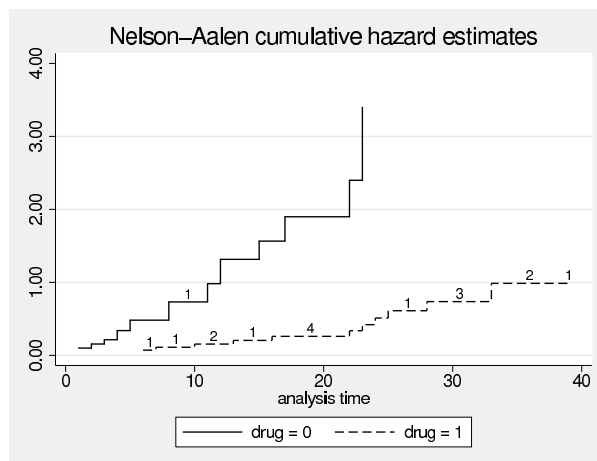
(Continued on next page)

```
. use http://www.stata-press.com/data/r11/drug2
(Patient Survival in Drug Trial)
. stset, noshow
. sts graph, cumhaz by(drug)
```



And here is a plot including the number lost because of censoring:

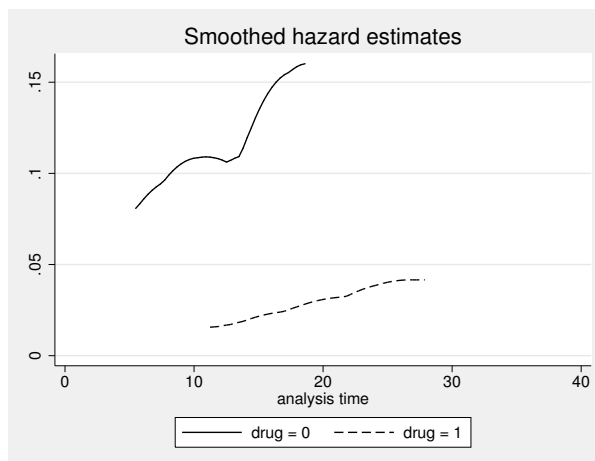
```
. sts graph, cumhaz by(drug) lost
```



Graphing the hazard function

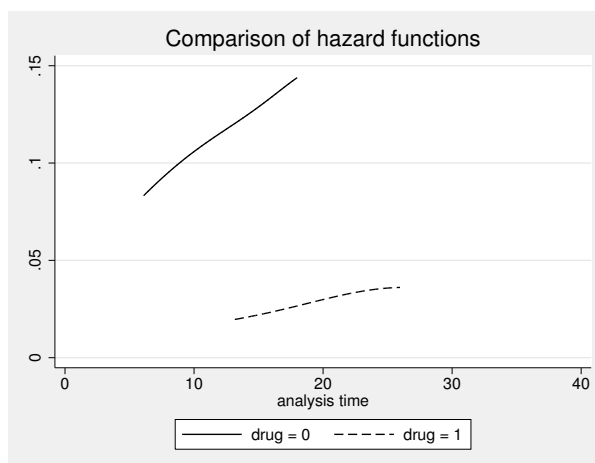
`sts graph` may also be used to plot an estimate of the hazard function. This graph is based on a weighted kernel smooth of the estimated hazard contributions, $\Delta \hat{H}(t_j) = \hat{H}(t_j) - \hat{H}(t_{j-1})$, obtained by `sts generate newvar = h`. There are thus issues associated with selecting a kernel function and a bandwidth, although `sts graph` will use defaults if we do not want to worry about this.

```
. sts graph, hazard by(drug)
```



We can also adjust and customize the kernel smooth.

```
. sts graph, hazard by(drug) kernel(gauss) width(5 7)
> title(Comparison of hazard functions)
```

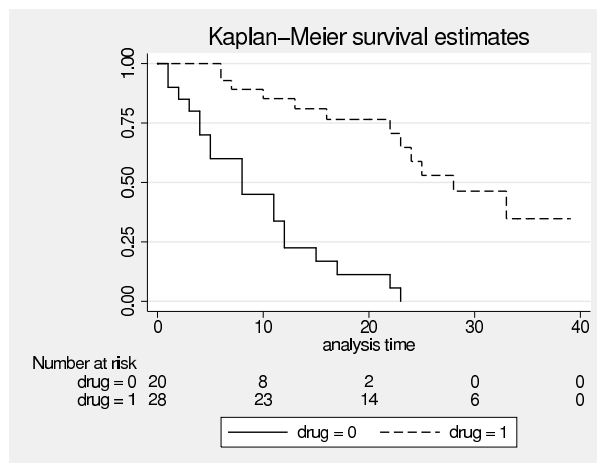


Adding an at-risk table

A table showing the number at risk may be added beneath a survivor, failure, or Nelson–Aalen cumulative hazard plot.

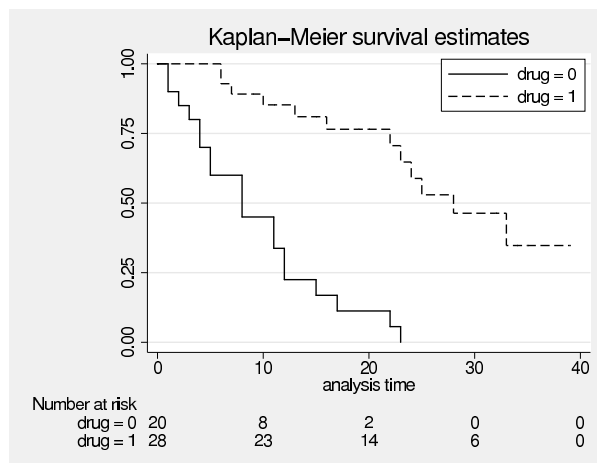
(Continued on next page)

```
. sts graph, by(drug) risktable
```



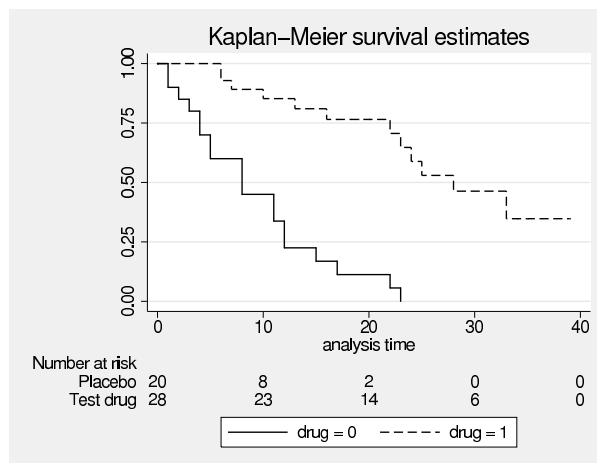
By default, both the legend and the at-risk table share space at the bottom of the graph. Placing the legend in an empty area inside the plot may often be desirable.

```
. sts graph, by(drug) risktable legend(ring(0) position(2) rows(2))
```



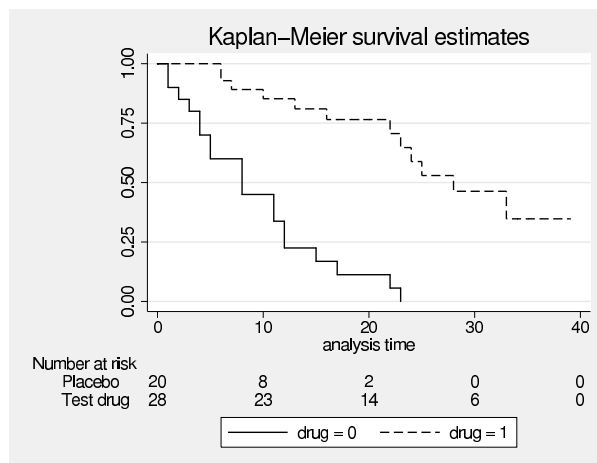
By default, row titles are placed on the left of the at-risk table and are right-justified. We can illustrate this by changing the text of the row titles to have an unequal length.

```
. sts graph, by(drug) risktable(, order(1 "Placebo" 2 "Test drug"))
```



If desired, the text of row titles can be left-justified.

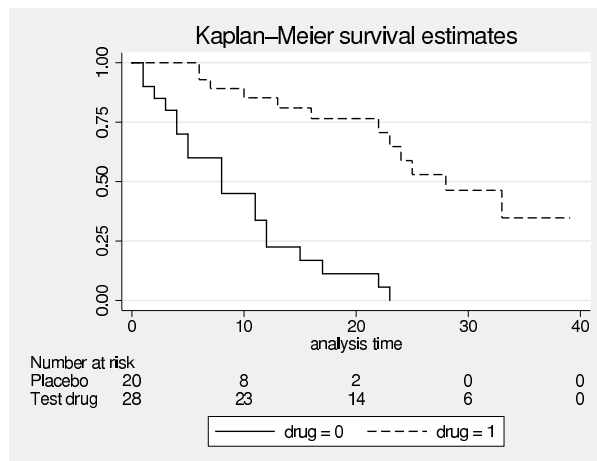
```
. sts graph, by(drug) risktable(, order(1 "Placebo" 2 "Test drug")
> rowtitle(, justification(left)))
```



(Continued on next page)

In addition to left justification, the table title can be aligned with the row titles.

```
. sts graph, by(drug) risktable(, order(1 "Placebo" 2 "Test drug"))
> rowtitle(, justification(left)) title(, at(rowtitle))
```



On boundary bias for smoothed hazards

`sts graph` uses the usual smoothing kernel technique to estimate the hazard function. Kernel estimators commonly encounter bias when estimating near the boundaries of the data range, and therefore estimates of the hazard function in the boundary regions are generally less reliable. To alleviate this problem, estimates that use the `epan2`, `biweight`, and `rectangular` kernels are adjusted at the boundaries with what are known as *boundary kernels* (e.g., Müller and Wang [1994]; Hess, Serachitopol, and Brown [1999]). For estimates using other kernels, no boundary adjustment is made. Instead, the default graphing range is constrained to be the range $[L + b, R - b]$, where L and R are the respective minimum and maximum analysis times at which failure occurred and b is the bandwidth.

Methods and formulas

See [ST] `sts`.

The estimated hazard is calculated as a kernel smooth of the estimated hazard contributions, $\Delta\hat{H}(t_j) = \hat{H}(t_j) - \hat{H}(t_{j-1})$, using

$$\hat{h}(t) = b^{-1} \sum_{j=1}^D K_t \left(\frac{t - t_j}{b} \right) \Delta\hat{H}(t_j)$$

where $K_t(\cdot)$ is the kernel (Müller and Wang 1994) function, b is the bandwidth, and the summation is over the D times at which failure occurs (Klein and Moeschberger 2003, 167). If `adjustfor()` is specified, the $\Delta\hat{H}(t_j)$ are instead obtained from `stcox` as the estimated baseline contributions from a Cox model; see [ST] `stcox` for details on how the $\Delta\hat{H}(t_j)$ are calculated in this case.

Pointwise confidence bands for smoothed hazard functions are calculated using the method based on a log transformation,

$$\widehat{h}(t) \exp \left[\pm \frac{Z_{1-\alpha/2} \sigma \{ \widehat{h}(t) \}}{\widehat{h}(t)} \right]$$

See Klein and Moeschberger (2003, 168) for details.

References

- Hess, K. R., D. M. Serachitopol, and B. W. Brown. 1999. Hazard function estimators: A simulation study. *Statistics in Medicine* 18: 3075–3088.
- Klein, J. P., and M. L. Moeschberger. 2003. *Survival Analysis: Techniques for Censored and Truncated Data*. 2nd ed. New York: Springer.
- Müller, H.-G., and J.-L. Wang. 1994. Hazard rate estimation under random censoring with varying kernels and bandwidths. *Biometrics* 50: 61–76.

Also see [ST] **sts** for more references.

Also see

- [ST] **sts** — Generate, graph, list, and test the survivor and cumulative hazard functions
- [ST] **sts generate** — Create variables containing survivor and related functions
- [ST] **sts list** — List the survivor or cumulative hazard function
- [ST] **sts test** — Test equality of survivor functions
- [ST] **stset** — Declare data to be survival-time data
- [R] **kdensity** — Univariate kernel density estimation