



The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.

From the help desk: Kaplan–Meier plots with **stsatrisk**

Jean Marie Linhart
Jeffrey S. Pitblado
James Hassell

StataCorp

Abstract. `stsatrisk` is a wrapper for `sts graph` that adds a table to a survival plot with at-risk information, making it easy to create graphs that follow the list of recommendations given by Pocock et al. (2002) for Kaplan–Meier plots. We use `stsatrisk` to create plots in the desired format with the desired information.

Keywords: st0058, `stsatrisk`, Kaplan–Meier, survival plots

1 Introduction

Pocock, Clayton, and Altman (2002) make the following recommendations for survival plots:

1. Survival plots are best presented going upwards, to maximize detail without needing a break in the scale.
2. Plots should only be extended through the period of follow-up achieved by a reasonable proportion of participants.
3. The extent of follow-up should be explained—e.g., by listing at regular intervals under the time axis the number still at risk in each treatment group.
4. Plots should include some measure of statistical uncertainty; otherwise, any visual signs of treatment differences might look more convincing than they really are. Either standard errors or confidence intervals should be displayed at regular time points, or an overall estimate of treatment difference (e.g., relative risk) with its 95% CI should be given.
5. Authors and readers should be cautious in interpreting the shape of survival plots. The lack of follow-up and poor estimation to the right-hand end, the lack of any prespecified hypothesis, and the lack of statistical power to explore subtleties of treatment difference other than the overall comparison should be recognized.

With the exception of item 3, most of these suggestions are straightforward to implement with options to `sts graph` or the graphics system. Consequently, we are primarily concerned with item 3, which calls for a table combined with a graph to give the at-risk

information associated with the ticks on the time axis. A combined graph and table is not part of the current Stata 8 graphics system; however, use of the `addtext()` option allows an ad hoc table to be created. `stsatrisk` is a wrapper for `sts graph` that creates such a table automatically for Kaplan–Meier plots. It works well under a wide variety of circumstances, but the program is not infinitely flexible and can only be used with a limited number of `by()` groups and tick marks.

2 Description of `stsatrisk`

`stsatrisk` is a wrapper to `sts graph` that adds notation to the Kaplan–Meier graph with the number at risk. Only the survivor and failure functions can be graphed. By default, `stsatrisk` will calculate 5 good values for the major ticks on the time axis and label the at-risk information at these points.

This command is limited in its facilities. It does not work with every `sts graph` option and does not necessarily produce pretty graphs with every possible graph option or scheme. You can have up to 6 `by()` or `strata()` groups but no more. Even with 6 or fewer groups, you can still run into a “too many options” error message—see the notes in section 8 of this article or the help file for more information on this error.

3 Syntax

```
stsatrisk [if exp] [in range] [, catrisk(numlist [, textbox_options])
      nolabel showevents tablegend clabel(labels) tablabel(string) llength(#)
      lspace(#) vspace(#) sts_graph_options cline_options twoway_options]
```

4 Options

`catrisk(numlist [, textbox_options])` customizes the time values at which the at-risk information is to be noted. This list will also provide the (labeled) major ticks on the time axis.

textbox_options affect how the added text for the at-risk information is displayed. They are described in [G] *textbox_options*.

`nolabel` suppresses the use of value labels of the `by()` or `strata()` variable to label the at-risk table and instead labels by its values. `nolabel` may only be specified with `by()` or `strata()`.

`showevents` shows the number failed for the period after the time point when the at-risk information was calculated in parentheses after the at-risk information.

`tablegend` includes a table summarizing the at-risk and event data with the legend.

`clabel(labels)` provides custom labels for the at-risk table rows. These labels will also be used to label the legend if one is generated.

`tablabel(string)` provides a customized label or title for the at risk table.

`llength(#)` specifies the maximum length of labels used in the at-risk table and legend if value labels are used. If custom labels are specified with the `clabel` option, the length is unrestricted. If there is one string `by()` or `strata()` variable, its values will also be used without truncation. The default value is 16.

`lspace(#)` allows the user to increase or decrease the horizontal (label) space for the at-risk table labels. The default value is one, and `lspace()` multiplies the horizontal space parameters.

`vspace(#)` allows the user to increase or decrease the vertical space for the at-risk table. The default value is one, and `vspace()` multiplies the vertical space parameters.

`sts_graph_options` are (most of) the options documented in [ST] `sts` for the `sts graph` command.

`cline_options` are the options documented in [G] `connect_options`.

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

Options `xmtick()` and `ymtick()` are not allowed with `stsatrisk`. They are used to create space for the at-risk table and are not available to the user.

5 Dialog

The `stsatrisk` package includes a dialog-box program for this command, contained in the file `stsatrisk.dlg`, which is downloaded with the program. The `stsatrisk` dialog box is a modification of the dialog for `sts graph`, as seen in figure 1. The options unique to `stsatrisk` are on the **At risk options** tab.

You can launch the dialog interactively with the command `db stsatrisk` from within Stata.

(Continued on next page)

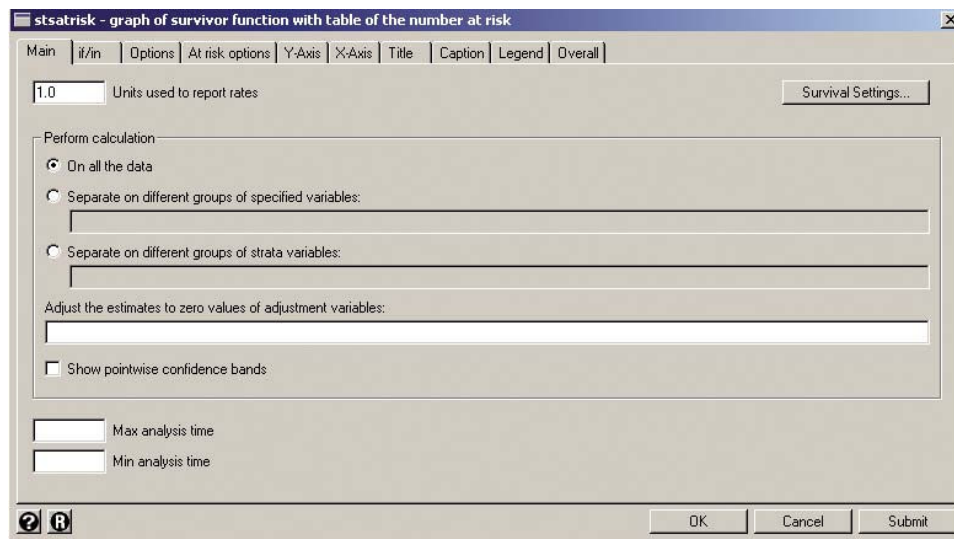


Figure 1: The `stsatrisk` dialog box is a modification of the `sts graph` dialog box.

GUI users can add `stsatrisk` permanently to their **User** menu by including the following in `profile.do`:

```
if _caller() > 7 {
    if "'c(console)'"==" {
        window menu append item "stUserGraphics"          ///
        "At risk info. on Kaplan-Meier plots (&stsatrisk)"  ///
        "db stsatrisk"
    }
}
```

6 Using `stsatrisk` to follow the recommendations

We will use a modification of the `cancer.dta` dataset and create sample graphs to demonstrate the `stsatrisk` command and show how the recommendations of Pocock, Clayton, and Altman (2002) can be implemented in Kaplan–Meier survival plots.

First, we load the `cancer.dta` dataset and modify it for our purposes:

```
. sysuse cancer, clear
. set scheme sj                      /* Stata Journal scheme */
. replace drug = (drug == 1)         /* makes two possible values for drug */
. label define drtype 0 placebo 1 active
. label val drug drtype
. expand 10                          /* 48 obs --> 480 obs */
. stset studytime, failure(died)
```

The output is omitted because it is not of interest here. This gives us a survival-time dataset, two drug types and 480 observations.

We will be comparing two or more treatment options. In both `sts graph` and `stsatrisk`, we can add an option of `by(drug)` to see two alternatives side by side.

The first recommendation of Pocock, Clayton, and Altman (2002) is that the graph be shown going up, which requires the use of the `failure` option to `sts graph`. The second recommendation is that plots should only be extended through the period of follow-up achieved by a reasonable proportion of the participants. Without splitting hairs on what a reasonable proportion means, let us take that to mean through time 20 for this dataset; we will use the `tmax(20)` option to `sts graph` to truncate our results at this time. The third recommendation is that the extent of follow-up should be explained via an at-risk table, such as the table `stsatrisk` was designed to display. This is all straightforward.

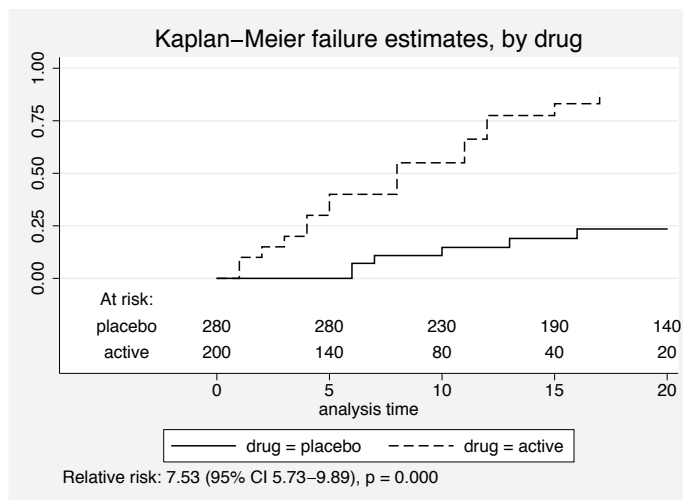
The fourth recommendation of Pocock, Clayton, and Altman (2002) is to include some measure of statistical uncertainty, such as SEs or CIs, at regular intervals or an overall estimate of the treatment difference with its 95% CI. The second of these two is the most easily implemented with `stsatrisk`. We will use `stcox` to estimate the treatment difference and then create our graph. Note that we use the options described above and also `caption()` to add the estimate of treatment difference information to our graph. The resulting graph is figure 2.

```
. stcox drug, noshow nolog
Cox regression -- Breslow method for ties
No. of subjects =          480          Number of obs   =          480
No. of failures =          310
Time at risk    =          7440
LR chi2(1)      =        238.23
Log likelihood  =   -1593.8033      Prob > chi2       =        0.0000
```

_t	Haz. Ratio	Std. Err.	z	P> z	[95% Conf. Interval]
drug	7.532496	1.047834	14.52	0.000	5.734949 9.893461

```
. stsatrisk, by(drug) failure tmax('tmax')
>      caption("Relative risk: 7.53 (95% CI 5.73-9.89), p = 0.000")
      failure _d: died
      analysis time _t: studytime
```

(Continued on next page)

Figure 2: `stsatrisk` graph by drug with statistical uncertainty caption

`sts graph` puts on CIs via the `gwood` option but separates the graphs in a `by()` graph to avoid confusing overlap. We could use `sts generate` to generate confidence intervals and then put these on the graph manually. However, this procedure is complicated and is still prone to confusing overlap. `stsatrisk` does not allow specification of `gwood` and `by()` simultaneously due to problems with conflicting and confusing information. If you want both confidence intervals and at-risk information on a `by()` graph, the best solution is to combine two graphs. Note the use of the `clabel()` option to `stsatrisk`, which customizes the labels in the at-risk table and the legend. In this case, we use it to get a more nicely labeled legend. The `title("")` option is used to suppress the titles in each of the two graphs to be combined. Since the `sts graph` is a `by()` graph, the `title("")` on this graph must be an option to `by()` in order to affect the overall title of the `sts graph`. One title is then given for the combined graph. The combined graph is figure 3, and the commands that generate it are below.

```
. stsatrisk, by(drug) failure tmax(20) clabel("Placebo" "Active")
> title("") name(stsatrisk, replace)
> caption("Relative risk 7.53 (95% CI 5.73-9.89), p = 0.000")
. sts graph, by(drug, title("")) failure tmax(20) gwood
> name(cigraph, replace)
. graph combine stsatrisk cigraph,
> title("Kaplan-Meier failure estimates, by drug")
```

(Continued on next page)

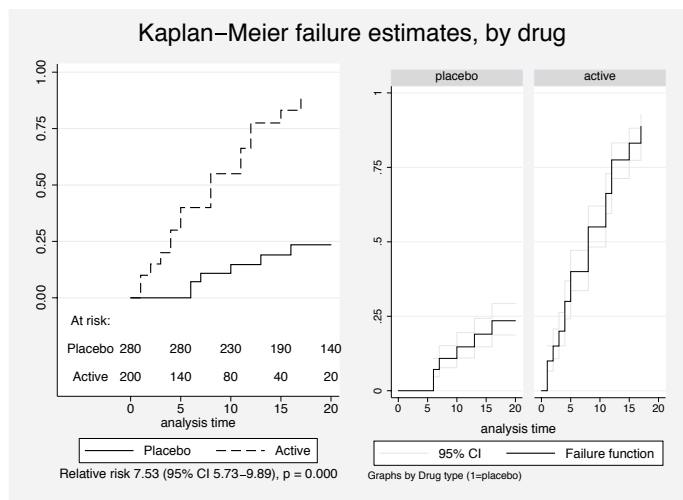


Figure 3: `stsatrisk` graph by drug and then with confidence intervals

The final recommendation of Pocock, Clayton, and Altman (2002) is a caution in interpreting the shape of the survival plots. Thus, we have used `stsatrisk` to follow all the recommendations.

7 Options to make nicer graphs

We will use the various labeling options to `stsatrisk` and one of the `textbox_options` to the `catrisk()` option to customize the at-risk table.

The `catrisk()` option lets us customize the tick marks at which we put our at-risk table. Here is the command:

```
. stsatrisk, by(drug) catrisk(0(4)20) clabel("Placebo" "Active")
> failure tmax(20)
> caption("Relative risk 7.53 (95% CI 5.73-9.89), p = 0.000")
```

(Continued on next page)

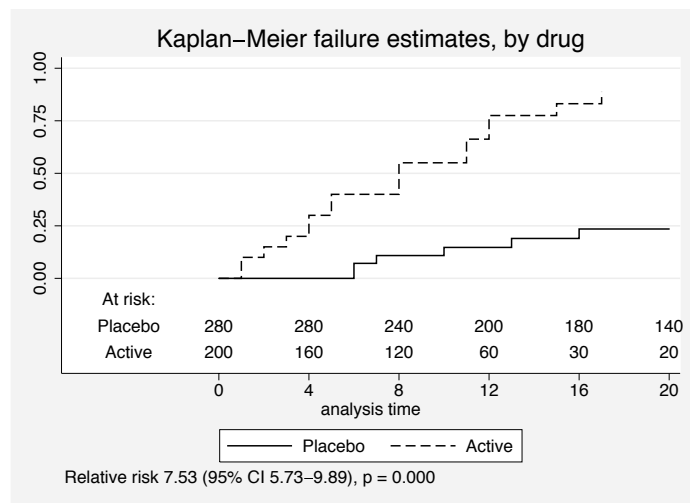


Figure 4: Customized tick marks and customized labels for the at-risk table and legend.

We can also show the number of events that occur between the time marks in parentheses with the option `showevents` and put a summary table of this information in with the legend with the `tablegend` options. Here is the command to do this and generate figure 5.

```
. stsatrisk, showevents tablegend
> by(drug) clabel("Placebo" "Active")
> failure tmax(20)
> caption("Relative risk 7.53 (95% CI 5.73–9.89), p = 0.000")
```

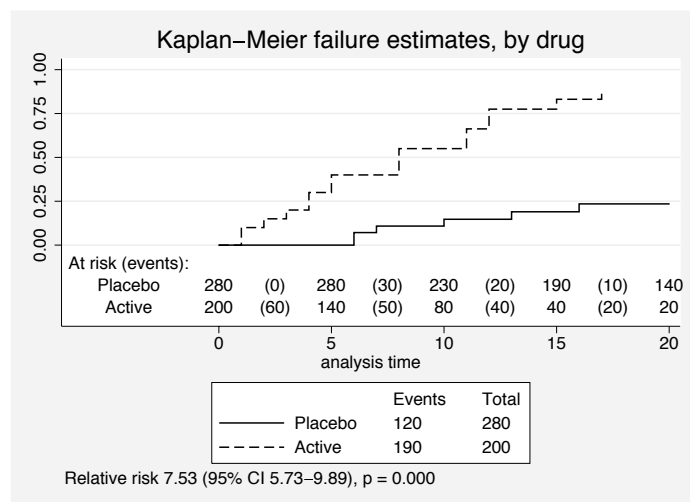


Figure 5: Graph with a legend showing the event information and a summary table.

If we wish to make the at-risk table a bit smaller, we can use the *textbox_option* `size(*#)` to the option `catrisk()`. We can, as seen below, use *textbox_options* alone, as well as with a *numlist* for `catrisk()`. We also customize the at-risk table title with `tablabel()`. The result of the following command is figure 6.

```
. stsatrisk, catrisk(, size(*.75))
>      tablabel("Number at risk (events):")
>      showevents tablelegend by(drug) clabel("Placebo" "Active")
>      failure tmax(20)
>      caption("Relative risk 7.53 (95% CI 5.73-9.89), p = 0.000")
```

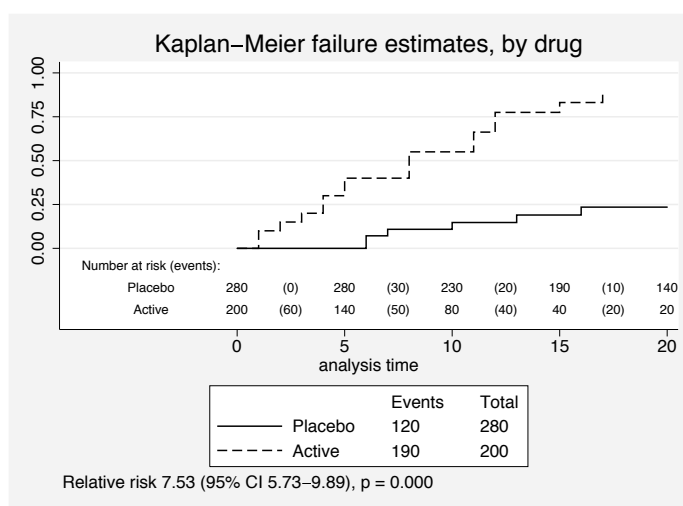


Figure 6: Graph with a smaller the at-risk table and a customized table title.

8 Additional notes

Because `stsatrisk` uses an ad hoc calculation to create space for the at-risk table and makes repeated use of the `addtext()` option to create the table, problems can occur in certain circumstances.

If labels are too long, they can overlap other parts of the graph. Generally, this can be remedied by adjusting the options `lspace(#)` and `vspace(#)`. These options allow the user to fine-tune the space allocation that `stsatrisk` does to make room for the at-risk table. These options multiply the space allocation and default to one, so space can be increased with values greater than one or decreased with values less than one.

An error message of “too many options” can also be generated. This indicates that the number of `addtext()` options generated automatically by the code is too high. The only remedy for this is to reduce the number of `by()` or `strata()` groups or to reduce the number of tick marks at which at-risk information is to be given. The number of tick marks can be modified through the `catrisk()` option.

`stsatrisk.ado` also contains many comments to assist user-programmers and non-programmers alike in making changes, although most problems can be fixed by modifications to the call, as described above.

9 Acknowledgments

Patrick Royston and Matthew Sydes of the MRC Clinical Trials Unit provided extensive feedback on the `stsatrisk` program. Jens Lauritsen of Odense University Hospital contributed to the `stsatrisk` code. Vince Wiggins of StataCorp provided extensive advice on graphics for the original implementation of `stsatrisk`.

10 References

Pocock, S. J., T. C. Clayton, and D. G. Altman. 2002. Survival plots of time-to-event outcomes in clinical trials: good practice and pitfalls. *The Lancet* 359: 1686–1689.

About the Authors

Jean Marie Linhart is Senior Mathematician at StataCorp.

Jeffrey S. Pitblado is Senior Statistician at StataCorp.

James Hassell is a Technical Services Representative at StataCorp.