stcurve — Plot the survivor or related function after streg, stcox, and more

Description

`stcurve` plots the survivor, failure, hazard, or cumulative hazard function after `stcox`, `streg`, `stintreg`, `stintcox`, `stmgintcox`, `mestreg`, `xtstreg`, `lasso cox`, or `elasticnet cox`. `stcurve` also plots the cumulative subhazard or cumulative incidence function (CIF) after `stcrreg`.

Quick start

Plot the survivor function with covariates at their means after `stcox`, `streg`, `stintreg`, `stintcox`, `stmgintcox`, `mestreg`, `xtstreg`, `lasso cox`, or `elasticnet cox`

```
stcurve, survival
```

Same as above, but plot separate survivor functions for covariate `x` set to 1, 2, and 3

```
stcurve, survival at1(x=1) at2(x=2) at3(x=3)
```

Same as above, but specify a numlist for `x` in `at()`

```
stcurve, survival at(x=(1 2 3))
```

Same as above, but specify a different pattern for each line

```
stcurve, survival at(x=(1 2 3)) plot1opts(lpattern(solid)) ///
  plot2opts(lpattern(dash)) plot3opts(lpattern(dot))
```

Same as above, and save the graph as `mygraph.gph`

```
stcurve, survival at(x=(1 2 3)) saving(mygraph)
```

Plot the estimated hazard function after `stcox`, `streg`, `stintreg`, `stintcox`, `stmgintcox`, `mestreg`, `xtstreg`, `lasso cox`, or `elasticnet cox`

```
stcurve, hazard
```

Smooth the estimated hazard contributions using the Gaussian kernel function for the kernel density estimate after `stcox`, `stintcox`, or `stmgintcox`, and set `x` to 1

```
stcurve, hazard kernel(gaussian) at(x=1)
```

Plot the cumulative hazard function after `stcox`, `streg`, `stintreg`, `stintcox`, `stmgintcox`, `mestreg`, `xtstreg`, `lasso cox`, or `elasticnet cox`

```
stcurve, cumhaz
```

Plot the cumulative subhazard function after `stcrreg`

```
stcurve, cumhaz
```

Plot the cumulative incidence function after `stcrreg`

```
stcurve, cif
```

Same as above, but set `x` to 0

```
stcurve, cif at(x=0)
```

Menu

Statistics > Survival analysis > Regression models > Plot survivor or related function

Syntax

`stcurve [ , options ]`

options	Description
Main	
* survival	plot survivor function
* failure	plot failure function
* hazard	plot hazard function
* cumhaz	plot cumulative hazard function
* cif	plot cumulative incidence function
atomeans	evaluate function at overall means; the default
attmeans	evaluate function at time-specific means; available only after stintcox and stmgintcox
at(*atspec*) [at1(*atspec1*) [at2(*atspec2*) [...]]]	values of the specified covariates and means of unspecified covariates
atframe(*framename*)	use covariate values from frame; available only after stintcox and stmgintcox
events(*evlist*)	plot functions for specified events; default is all events; available only after stmgintcox
sepevents	show event-specific curves on separate graphs; default is to show event-specific curves as subgraphs on one graph; available only after stmgintcox
Options	
alpha1	conditional frailty model
fixedonly	set all random effects to zero
unconditional	unconditional frailty model or random-effects model
marginal	synonym for unconditional
range(# #)	range of analysis time
outfile(*filename* [, replace])	save values used to plot the curves
width(#)	override "optimal" width; use with hazard
kernel(*kernel*)	kernel function; use with hazard
noboundary	no boundary correction; use with hazard
name(*namespec*, ...)	specify names of graphs
saving(*filespec*, ...)	save graphs in files
Plot	
connect_options	affect rendition of plotted survivor, failure, hazard, or cumulative hazard function
plot#opts(*connect_options*)	affect rendition of the #th plot
atplot#opts(*connect_options*)	affect rendition of the #th at-plot
event#opts(*connect_options*)	affect rendition of plots for the #th event; available only after stmgintcox
graph#opts(*twoway_opts*)	control the look of the #th graph; only allowed with sepevents after stmgintcox

Add plots	
`addplot(`*plot*`)`	add other plots to the generated graph
Y axis, X axis, Titles, Legend, Overall	
twoway_opts	control the look of all graphs; any options other than `by()`, `name()`, or `saving()` documented in [G-3] ***twoway_options***
By options	
`byopts(`*byopts*`)`	how subgraphs created by `event()` are combined, labeled, etc.; allowed only after `stmgintcox`, but not allowed with `sepevents`

*One of `survival`, `failure`, `hazard`, `cumhaz`, or `cif` must be specified.

`survival`, `failure`, and `hazard` are not allowed after estimation with `stcrreg`; see [ST] **stcrreg**

`cif` is allowed only after estimation with `stcrreg`; see [ST] **stcrreg**.

`stcurve` is not supported after stratified estimation.

For the `stcurve` syntax following `lasso cox` and `elasticnet cox`, see [LASSO] **lasso postestimation**.

Options

Main

`survival` specifies that the survivor function be plotted. `survival` is not allowed after estimation with `stcrreg`.

`failure` specifies that the failure function be plotted. `failure` is not allowed after estimation with `stcrreg`.

`hazard` specifies that the hazard function be plotted. `hazard` is not allowed after estimation with `stcrreg`.

`cumhaz` specifies that the cumulative hazard function be plotted when used after `stcox`, `streg`, `stintreg`, `stintcox`, `stmgintcox`, `mestreg`, or `xtstreg` and specifies that the cumulative subhazard function be plotted when used after `stcrreg`.

`cif` specifies that the cumulative incidence function be plotted. This option is available only after estimation with `stcrreg`.

`atomeans` specifies that the estimates of the survivor or other function be evaluated at the overall means of covariates. This is the default. For functions after `stmgintcox`, `atomeans` specifies that those estimates be evaluated at the overall means of covariates for each specified events.

`attmeans` is supported after `stintcox` and `stmgintcox` in a multiple-record-per-subject format. It specifies that the estimates of the survivor or other function be evaluated at the time-specific means of covariates (for each specified event after `stmgintcox`). This option is useful to incorporate time profiles for time-varying covariates present in the dataset. Also see the `atframe()` option.

`at(`*atspec*`)` specifies that the estimates of the survivor or other function be evaluated at specific covariate values. By default, `stcurve` evaluates the function by setting each covariate to its overall mean value. This option causes the function to be evaluated at the values of the covariates listed in `at()` and at the overall means of all unlisted covariates. If the `attmeans` option is also specified, the unlisted covariates are evaluated at time-specific means. This option can be repeated to produce multiple curves (up to 20), or you can specify multiple values for a set of covariates in one at() option; see *Syntax of at()* in [ST] ***adjustfor_option*** for details. at() may not be combined with at1(), at2(), and so on.

at1(*atspec1*), at2(*atspec2*), ..., at20(*atspec20*) are the alternatives to the repeated use of at(). They specify that multiple curves (up to 20) be plotted on the same graph. at1(), at2(), ..., at20() work similarly to the at() option. at1() specifies the values of the covariates for the first curve, at2() specifies the values of the covariates for the second curve, and so on. But, unlike at(), at#() cannot be repeated and may not be combined with at(). *atspec1*, *atspec2*, and so on follow the same syntax as *atspec*, except they do not allow *numlist*s or multiple values for the same covariate.

atframe(*framename*) is supported after stintcox in a multiple-record-per-subject format and after stmgintcox with multiple-record-per-event data. It specifies that the estimates of the survivor or other function be evaluated using the values of variables specified in the *framename* frame. The frame must contain a time variable with the same name as the examination time variable specified in the time() option with stintcox or stmgintcox. It must also include at least one covariate as specified with stintcox or stmgintcox or in their tvc() option. atframe() may not be combined with the at() option.

events(*evlist*) specifies that the function be plotted only for the specified events. The default is events(_all), which means the function is plotted for all events. The events() option is available only after stmgintcox.

evlist may be _all (indicating all events), a numlist with values of the event variable, a list of labels from the value label for the event variable, or a list such as #1, #2, ..., with #1 meaning the first event, #2 meaning the second event, etc. For example, suppose the event variable contains values 1, 2, 3 with corresponding labels “event1”, “event2”, and “event3” defined in its value label. If we would like to plot the cumulative hazard functions for the first two events, we can specify stcurve, cumhaz with one of the following options: events(1 2), events("event1" "event2"), or events(#1 #2).

sepevents is meaningful only after stmgintcox. By default, the plots for each event are combined as subgraphs on one graph. sepevents requests that the plots for each event be placed on separate graphs.

Options

alpha1, when used after fitting a frailty model, plots curves that are conditional on a frailty value of one. This is the default for shared-frailty models.

fixedonly specifies that all random effects be set to zero, which is equivalent to using only the fixed portion of the model, when plotting results for random-effects models. This option is allowed only after xtstreg or mestreg; it is the default after xtstreg.

unconditional and marginal, when used after fitting a frailty model or a random-effects model, plot curves that are unconditional on the frailty or on the random effects. That is, the curve is “averaged” over the frailty distribution or over the random-effects distributions. This is the default for unshared-frailty models and for random-effects models. This option is not allowed after stintreg, stintcox, stmgintcox, or xtstreg.

range(# #) specifies the range of the time axis to be plotted. If this option is not specified, stcurve plots the desired curve on an interval expanding from the earliest to the latest time in the data.

outfile(*filename* [, replace]) saves in *filename*.dta the values used to plot the curve(s).

width(#) is for use with hazard and is for use only after stcox, stintcox, or stmgintcox. width() is used to specify the bandwidth to be used in the kernel smooth used to plot the estimated hazard function. If left unspecified, a default bandwidth is used, as described in [R] **kdensity**.

kernel(*kernel*) is for use with hazard and is for use only after stcox, stintcox, or stmgintcox because, for Cox regression, an estimate of the hazard function is obtained by smoothing the estimated hazard contributions. kernel() specifies the kernel function for use in calculating the weighted kernel-density estimate required to produce a smoothed hazard-function estimator. The default is kernel(epanechnikov), yet *kernel* may be any of the kernels supported by kdensity; see [R] **kdensity**.

noboundary is for use with hazard and applies only to the plotting of smoothed hazard functions after stcox, stintcox, or stmgintcox. 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; see [ST] **sts graph**. If the epan2, biweight, or rectangular kernel is used after estimation using stcox, 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 inside of one bandwidth from each endpoint. For these other kernels, specifying noboundary merely removes this range restriction. After estimation using stintcox or stmgintcox, the boundary adjustments correspond to simply restricting the plotted range of the function for all kernels.

name(*namespec*[, replace]) specifies the name of the graph or multiple graphs. For a single graph, see [G-3] ***name_option***. If multiple graphs are produced, then the argument of name() is either a list of names or a *stub*, in which case graphs are named *stub*1, *stub*2, and so on. replace causes existing graphs with the specified name or names to be replaced.

saving(*filespec*[, replace]) specifies the filename or filenames to use to save the graph or multiple graphs to disk. For a single graph, see [G-3] ***saving_option***. If multiple graphs are produced, then the argument of saving() is either a list of filenames or a *stub*, in which case graphs are saved with filenames *stub*1, *stub*2, and so on. replace specifies that the file (or files) be replaced if it already exists.

Plot

connect_options affect the rendition of all plotted survivor, failure, hazard, or cumulative hazard functions; see [G-3] ***connect_options***. They may be overridden for specific plots by using plot#opts(), at#plotopts(), or event#opts().

plot#opts(*connect_options*) affect the rendition of the #th plotted function. When multiple options apply to the same plot, the *connect_options* specified with plot#opts() will override those specified with atplot#opts() and event#opts().

atplot#opts(*connect_options*) affect the rendition of the #th at-plot function created by at() or at#(). When you plot functions for multiple events after stmgintcox, the default is to use the same line color and pattern for the #th at-plot function for all events. When multiple options apply to the same plot, the *connect_options* specified with plot#opts() will override those specified with atplot#opts(), and the options specified with atplot#opts() will override those specified with event#opts().

event#opts(*connect_options*) affect the rendition of the plotted function for the #th event after stmgintcox. When multiple options apply to the same plot, the *connect_options* specified with plot#opts() will override those specified with atplot#opts(), and the options specified with atplot#opts() will override those specified with event#opts().

`graph#opts(`*twoway_opts*`)` affects the appearance of the #th graph when `sepevents` is specified after `stmgintcox`. *twoway_opts* are any of the options documented in [G-3] ***twoway_options***, excluding `by()`, `name()`, and `saving()`.

Add plots

`addplot(`*plot*`)` provides a way to add other plots to the generated graph; see [G-3] ***addplot_option***. `addplot()` is not allowed when the graph contains subgraphs.

Y axis, X axis, Titles, Legend, Overall

twoway_opts control the appearance of all graphs; they are any of the options documented in [G-3] ***twoway_options***, excluding `by()`, `name()`, and `saving()`. These include options for titling the graph (see [G-3] ***title_options***) and for specifying legends (see [G-3] ***legend_options***). They may be overridden for specific graphs by using the `graph#opts()` option.

By options

`byopts(`*byopts*`)` affects the appearance of the combined subgraphs on one graph when functions are plotted for multiple events after `stmgintcox`; `byopts()` is not allowed in combination with `sepevents`.

byopts may be any of the suboptions of `by()` documented in [G-3] ***by_option***, except for `total`, `missing`, and *legend_options*.

Remarks and examples

Remarks are presented under the following headings:

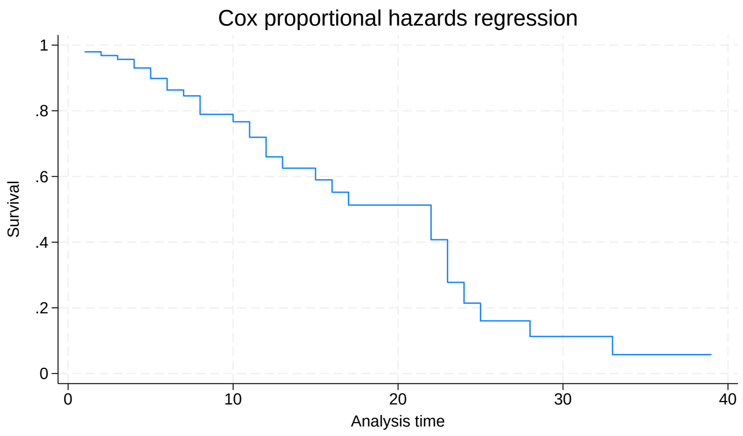
For examples of `stcurve` after `xtstreg` and `mestreg`, see [XT] **xtstreg postestimation** and [ME] **mestreg postestimation**, respectively.

stcurve after stcox

After fitting a Cox model, `stcurve` can be used to plot the estimated survivor, failure, hazard, or cumulative hazard function.

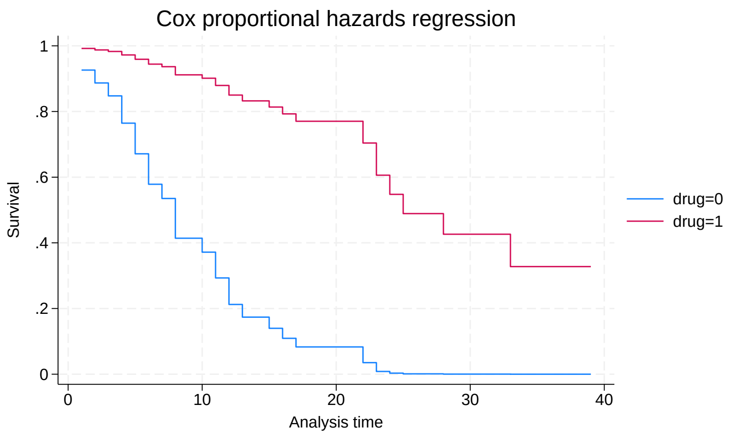
▷ Example 1

```
. use https://www.stata-press.com/data/r19/drugtr
(Patient survival in drug trial)
. stcox age drug
  (output omitted)
. stcurve, survival
note: function evaluated at overall means of covariates.
```



By default, the curve is evaluated at the mean values of all the predictors, but we can specify other values if we wish.

```
. stcurve, survival at1(drug=0) at2(drug=1)
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any).
```



Alternatively, you can obtain the same plot by typing the following:

```
. stcurve, survival at(drug=(0 1))
```

In this example, we asked for two plots, one for the placebo group and one for the treatment group. For both groups, the value of age was held at its mean value for the overall estimation sample.

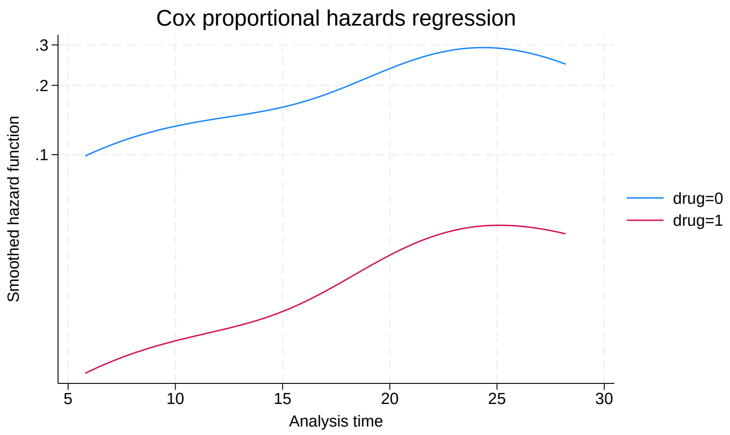
See Cefalu (2011) for a Stata command to plot the survivor or cumulative hazard function with pointwise confidence intervals.

◁

▷ Example 2

stcurve can also be used to plot estimated hazard functions. The hazard function is estimated by a kernel smooth of the estimated hazard contributions; see [ST] **sts graph** for details. We can thus customize the smooth as we would any other; see [R] **kdensity** for details.

```
. stcurve, hazard at(drug=(0 1)) kernel(gauss) yscale(log)
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any).
```



For the hazard plot, we plotted on a log scale to demonstrate the proportionality of hazards under this model; see the technical note below on smoothed hazards.

◁

❑ Technical note

For survivor or cumulative hazard estimation, stcurve works by first estimating the baseline function and then modifying it to adhere to the specified (or by default, mean) covariate patterns. As mentioned previously, *baseline* (when all covariates are equal to zero) must correspond to something that is meaningful and preferably in the range of your data. Otherwise, stcurve could encounter numerical difficulties. We ignored our own advice above and left age unchanged. Had we encountered numerical problems, or funny-looking graphs, we would have known to try shifting age so that age==0 was in the range of our data.

For hazard estimation, stcurve works by first transforming the estimated hazard contributions to adhere to the necessary covariate pattern and then applying the smooth. When you plot multiple curves, each is smoothed independently, although the same bandwidth is used for each.

The smoothing takes place in the hazard scale and not in the log hazard-scale. As a result, the resulting curves will look nearly, but not exactly, parallel when plotted on a log scale. This inexactitude is a product of the smoothing and should not be interpreted as a deviation from the proportional-hazards assumption; stcurve (after stcox) assumes proportionality of hazards and will reflect this in the produced plots. If smoothing were a perfect science, the curves would be parallel when plotted on a log scale. If you encounter estimated hazards exhibiting severe disproportionality, this may signal a numerical problem as described above. Try recentering your covariates so that baseline is more reasonable.

❑

stcurve after streg

stcurve is used after streg to plot the fitted survivor, failure, hazard, or cumulative hazard function. By default, stcurve computes the means of the covariates and evaluates the functions at each time in the data, censored or uncensored. The resulting plot is therefore the survival experience of a subject with a covariate pattern equal to the average covariate pattern in the study. You can produce the plot at other values of the covariates by using the at() option or specify a time range by using the range() option.

▷ Example 3

We pick up where example 6 of [ST] **streg** left off. The cancer dataset we are using has three values for variable drug: 1 corresponds to placebo, and 2 and 3 correspond to two alternative treatments. Using the cancer data with drug remapped to form an indicator of treatment, let's fit a loglogistic regression model and plot its survival curves. We can perform a loglogistic regression by issuing the following commands:

```
. use https://www.stata-press.com/data/r19/cancer
(Patient survival in drug trial)

. replace drug = drug==2 | drug==3                // 0, placebo : 1, nonplacebo
(48 real changes made)

. stset studytime, failure(died)
  (output omitted)

. streg age drug, distribution(llogistic) nolog

        Failure _d: died
  Analysis time _t: studytime

Loglogistic AFT regression

No. of subjects =  48                                  Number of obs =     48
No. of failures =  31
Time at risk    = 744
                                                       LR chi2(2)    =  35.14
Log likelihood = -43.21698                             Prob > chi2   = 0.0000

------------------------------------------------------------------------------
          _t | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |  -.0803289   .0221598    -3.62   0.000    -.1237614   -.0368964
        drug |   1.420237   .2502148     5.68   0.000     .9298251    1.910649
       _cons |   6.446711   1.231914     5.23   0.000     4.032204    8.861218
-------------+----------------------------------------------------------------
    /lngamma |  -.8456552   .1479337    -5.72   0.000      -1.1356   -.5557105
-------------+----------------------------------------------------------------
       gamma |    .429276   .0635044                      .3212293    .5736646
------------------------------------------------------------------------------
```

Now, we wish to plot the survivor and the hazard functions:

```
. stcurve, survival ylabels(0 .5 1)
note: function evaluated at overall means of covariates.
```

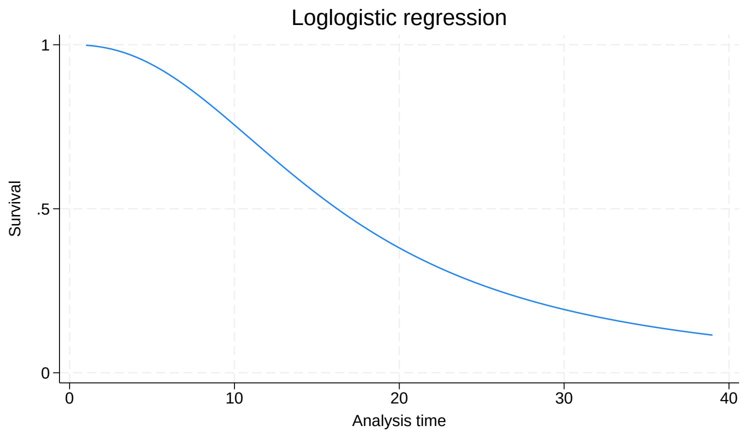


Figure 3. Loglogistic survival distribution at mean value of all covariates

```
. stcurve, hazard
note: function evaluated at overall means of covariates.
```

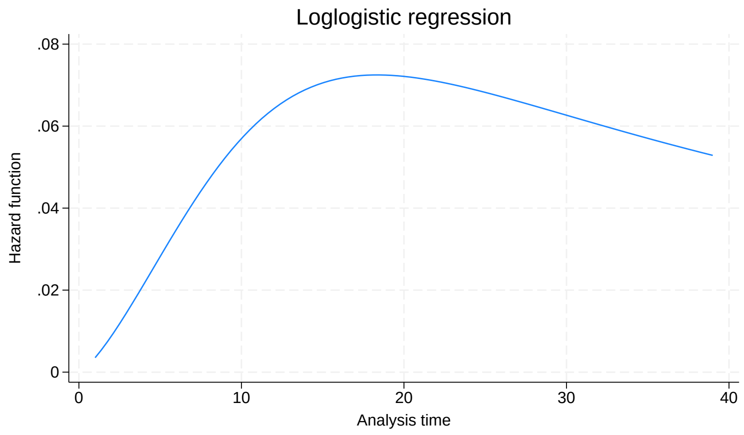


Figure 4. Loglogistic hazard distribution at mean value of all covariates

These plots show the fitted survivor and hazard functions evaluated for a cancer patient of average age receiving the average drug. Of course, the "average drug" has no meaning here because `drug` is an indicator variable. It makes more sense to plot the curves at a fixed value (level) of the drug. We can do this with the `at` option. For example, we may want to compare the average-age patient's survival curve under placebo (`drug==0`) and under treatment (`drug==1`).

We can plot both curves on the same graph:

```
. stcurve, survival at(drug=(0 1)) ylabels(0 .5 1)
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any).
```

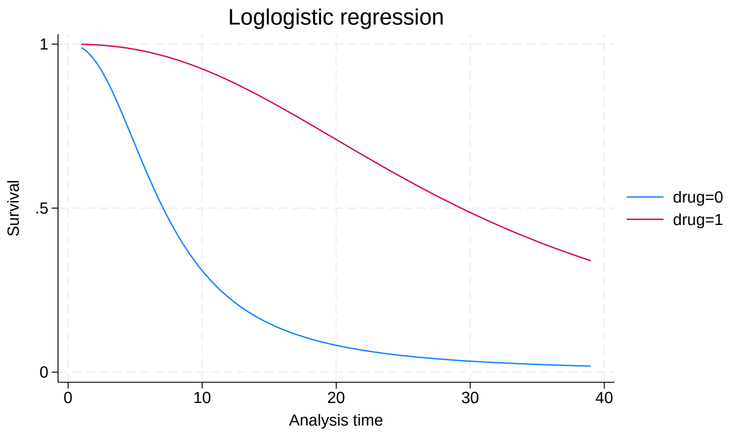


Figure 5. Loglogistic survival distribution at mean age for placebo

In the plot, we can see from the loglogistic model that the survival experience of an average-age patient receiving the placebo is worse than the survival experience of that same patient receiving treatment. We can also see the accelerated-failure-time feature of the loglogistic model. The survivor function for treatment is a time-decelerated (stretched-out) version of the survivor function for placebo.

◁

▷ Example 4

In our discussion of frailty models in [ST] **streg**, we emphasize the distinction between the individual hazard (or survivor) function and the hazard (survivor) function for the population. When significant frailty is present, the population hazard will tend to begin falling past a certain point, regardless of the shape of the individual hazard. This is due to the frailty effect—as time passes, the frailer individuals will fail, leaving a more homogeneous population comprising only the most robust individuals.

The frailty effect may be demonstrated using `stcurve` to plot the estimated hazard (both individual and population) after fitting a frailty model. Use the `alpha1` option to specify the individual hazard ($\alpha = 1$) and the `unconditional` option to specify the population hazard. Applying this to the Weibull/inverse-Gaussian shared-frailty model on the kidney data of example 11 of [ST] **streg**,

```
. use https://www.stata-press.com/data/r19/catheter, clear
(Kidney data, McGilchrist and Aisbett, Biometrics, 1991)
. stset time infect
  (output omitted)
. quietly streg age female, distribution(weibull) frailty(invgauss) shared(patient)
. stcurve, hazard at(female=1) alpha1
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any).
```

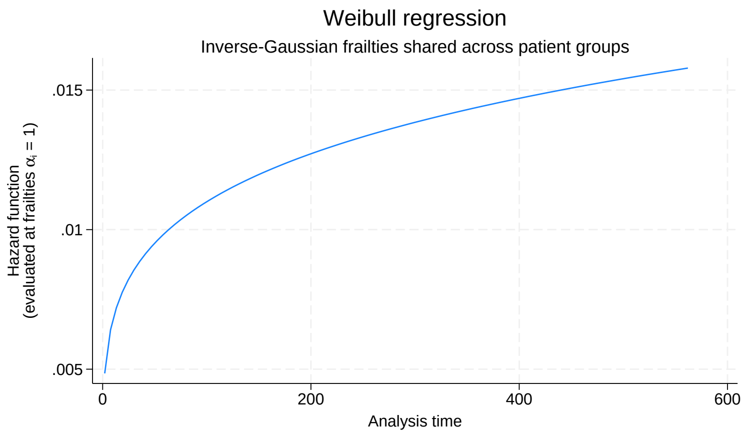


Figure 6. Individual hazard for females at mean age

Compare with

```
. stcurve, hazard at(female=1) unconditional
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any).
```

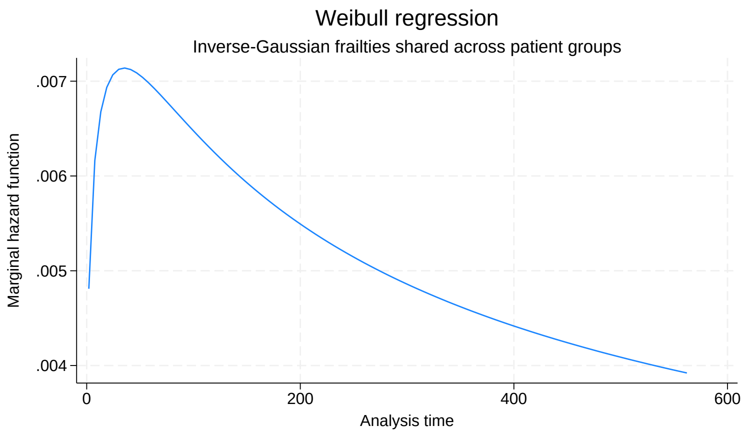


Figure 7. Population hazard for females at mean age

◁

stcurve after stcrreg

▷ Example 5

In [ST] **stcrreg**, we analyzed data from 109 patients with primary cervical cancer, treated at a cancer center between 1994 and 2000. We fit a competing-risks regression model where local relapse was the failure event of interest (`failtype == 1`), distant relapse with no local relapse was the competing risk event (`failtype == 2`), and we were interested primarily in the effect of interstitial fluid pressure (`ifp`) while controlling for tumor size and pelvic node involvement.

After fitting the competing-risks regression model, we can use `stcurve` to plot the estimated cumulative incidence of local relapses in the presence of the competing risk. We wish to compare the cumulative incidence curves for `ifp == 5` versus `ifp == 20`, assuming positive pelvic node involvement (`pelnode == 0`) and a tumor size that is the average over the data.

```
. use https://www.stata-press.com/data/r19/hypoxia
(Hypoxia study)
. stset dftime, fail(failtype==1)
  (output omitted)
. stcrreg ifp tumsize pelnode, compete(failtype==2)
  (output omitted)
. stcurve, cif at(ifp=(5 20) pelnode=0)
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any).
```

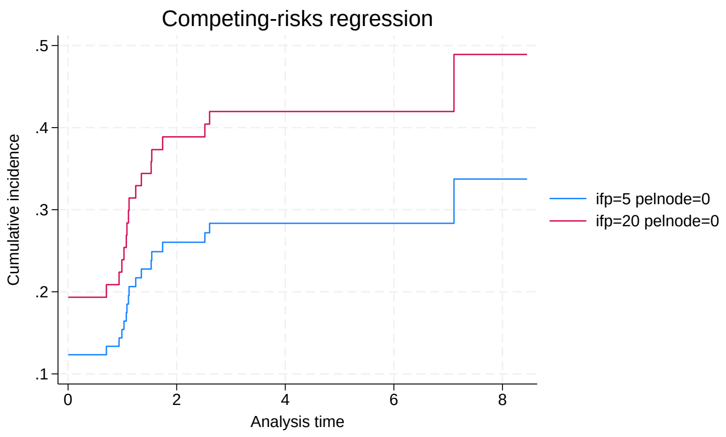


Figure 8. Comparative cumulative incidence functions

Specifying `ifp=(5 20)` in the `at()` option is the same as specifying the following `at#()` options:

```
. stcurve, cif at1(ifp=5 pelnode=0) at2(ifp=20 pelnode=0)
```

◁

stcurve after stintreg and stintcox

`stcurve` can be used after `stintreg` or `stintcox` to plot the fitted survivor, failure, hazard, or cumulative hazard function. For single-record interval-censored data, these functions can be evaluated at a lower or upper time endpoint of time intervals. `stcurve` after `stintreg` uses the lower and upper time endpoints to determine the range for the plotted functions. `stcurve` after `stintcox` plots the functions at the distinct time points formed by combining the lower and upper time endpoints. By default, without the at() option, stcurve computes the overall means of the covariates and evaluates the function at the overall means and at each time in the data, censored or uncensored. The resulting plot is therefore the survival experience of a subject with a covariate pattern equal to the average covariate pattern in the study. You can produce the plot at other values of the covariates by using the at() option or specify a time range by using the range() option. stcurve after stintcox can also be used to plot functions that allow covariates to vary over time; see *Remarks and examples* in [ST] **stintcox postestimation**.

▷ Example 6

We continue with example 1 of [ST] **stintreg**, which studies the effect of treatment on breast retraction for breast cancer patients. In that example, we compared the cosmetic effects of two cancer treatments, radiotherapy alone versus radiotherapy plus adjuvant chemotherapy, by fitting a Weibull proportional hazards model:

```
. use https://www.stata-press.com/data/r19/cosmesis, clear
(Cosmetic deterioration of breast cancer patients)
. stintreg i.treat, interval(ltime rtime) distribution(weibull)
  (iteration log omitted)
Weibull PH regression                              Number of obs      =     94
                                                         Uncensored =      0
                                                      Left-censored =      5
                                                     Right-censored =     38
                                                     Interval-cens. =     51

                                                   LR chi2(1)         =  10.93
Log likelihood = -143.19228                        Prob > chi2        = 0.0009

------------------------------------------------------------------------------
             |  Haz. ratio   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       treat |
 Radio+Chemo |   2.498526    .7069467     3.24   0.001     1.434961    4.350383
       _cons |   .0018503    .0013452    -8.66   0.000      .000445     .007693
-------------+----------------------------------------------------------------
       /ln_p |   .4785787    .1198973     3.99   0.000     .2435843     .713573
-------------+----------------------------------------------------------------
           p |   1.613779    .1934877                      1.275814    2.041272
         1/p |   .6196635     .074296                      .4898907    .7838134
------------------------------------------------------------------------------
Note: _cons estimates baseline hazard.
```

Now, we wish to compare the average patient's survival curve under radiotherapy only (`treat == 0`) and under radiotherapy plus chemotherapy (`treat == 1`):

```
. stcurve, survival at(treat=(0 1))
note: function evaluated at specified covariate values.
```

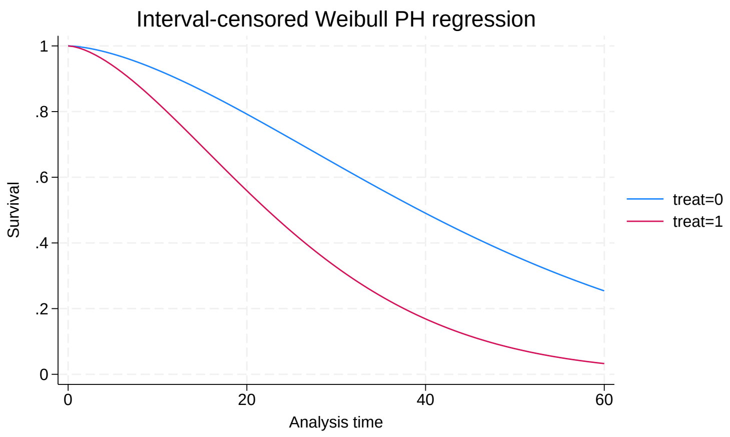


Figure 9. Treatment-specific survivor functions for Weibull proportional hazards model

From figure 9, we see that the risk of developing breast retraction for an average patient receiving the radiotherapy-plus-chemotherapy treatment is higher than that for the same patient receiving radiotherapy-only treatment. In other words, the adjuvant chemotherapy increases the risk of breast retraction.

Let's now use `stintcox` to fit a semiparametric Cox model that relaxes the distributional assumption about the event-time distribution. To speed up execution, we will use the `favorspeed` option in this demonstration.

```
. stintcox i.treat, interval(ltime rtime) favorspeed
note: using fixed step size with a multiplier of 5 to compute derivatives.
note: using EM and VCE tolerances of 0.0001.
note: option noemhsgtolerance assumed.
Performing EM optimization (showing every 100 iterations):
Iteration 0:   Log likelihood = -150.52924
Iteration 36:  Log likelihood = -133.02071
Computing standard errors: ... done
Interval-censored Cox regression                     Number of obs     =     94
Baseline hazard: Reduced intervals                          Uncensored =      0
                                                        Left-censored =      5
Event-time interval:                                   Right-censored =     38
  Lower endpoint: ltime                                Interval-cens. =     51
  Upper endpoint: rtime
                                                     Wald chi2(1)      =   8.34
Log likelihood = -133.02071                          Prob > chi2       = 0.0039

------------------------------------------------------------------------------
             |               OPG
             | Haz. ratio   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       treat |
 Radio+Chemo |   2.229089   .6188939     2.89   0.004     1.293589    3.841127
------------------------------------------------------------------------------
Note: Standard error estimates may be more variable for small datasets and
      datasets with low proportions of interval-censored observations.
```

And we now compare the survivor functions of the two treatment groups:

```
. stcurve, survival at(treat=(0 1))
note: function evaluated at specified covariate values.
```

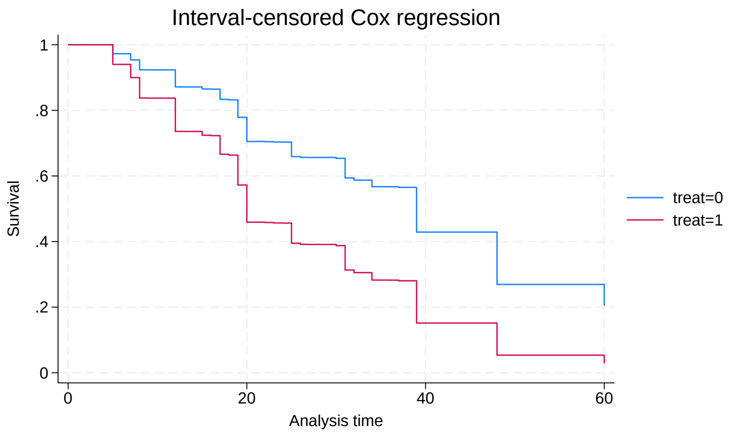


Figure 10. Treatment-specific survivor functions for Cox proportional hazards model

The survivor functions for the semiparametric Cox model are step functions but they look similar to the Weibull survivor functions from figure 9.

◁

stcurve after stmgintcox

`stcurve` can be used after `stmgintcox` to plot the fitted survivor, failure, hazard, or cumulative hazard function for all events in the data. You can also use the `events()` option to plot the functions only for the specified events. For each event, if the `at()` option is not specified, `stcurve` computes the overall means of the covariates for the specified event and evaluates the function at those means and at each time point for that event, whether censored or uncensored. You can produce the plots at other values of the covariates by using the `at()`, `at#()`, or `atframe()` option. You can also specify a time range using the `range()` option. By default, `stcurve` plots the estimated function for each event as a subgraph and places the subgraphs on a single graph. You can use the `sepevents` option to place the plot for each event on a separate graph.

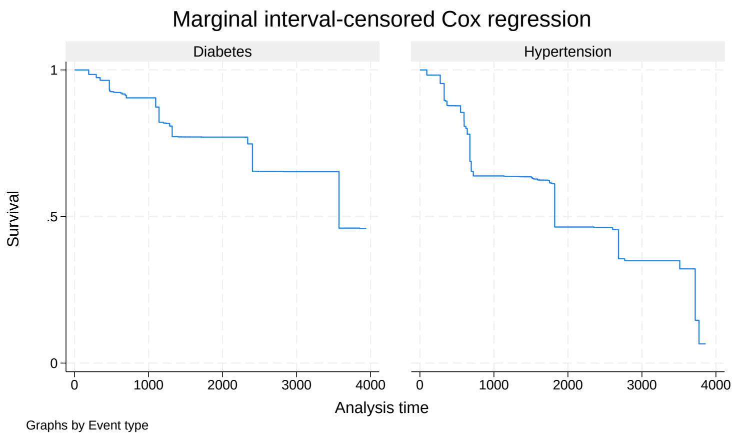
▷ Example 7

We continue with example 2 from [ST] **stmgintcox**, which examines the factors influencing the time to onset of diabetes and hypertension in different communities. In that example, we found that the key risk factors for diabetes differ from those for hypertension. Therefore, we can use different sets of covariates to model the two events. Additionally, we use the `favorspeed` option for the command to run faster purely for the purpose of demonstration. We also specify the `nolog` option to suppress the iteration log.

```
. use https://www.stata-press.com/data/r19/aric
(Simulated ARIC data)

. stmgintcox ("Diabetes": age i.male i.community i.race bmi glucose)
>  ("Hypertension": age i.male i.community i.race sysbp diabp),
>  id(id) event(event) interval(ltime rtime) nolog favorspeed
note: using fixed step size with a multiplier of 5 to compute derivatives.
note: using EM and VCE tolerances of 0.0001.
note: option noemhsgtolerance assumed.

Marginal interval-censored Cox regression          Number of events   =      2
Baseline hazard: Reduced intervals                 Number of subjects =    200
                                                   Number of obs      =    400
ID variable: id                                            Uncensored =      0
Event variable: event                                   Left-censored =     47
Event-time interval:                                   Right-censored =    240
  Lower endpoint: ltime                                Interval-cens. =    113
  Upper endpoint: rtime
                                                   Wald chi2(16)      =  77.01
Log pseudolikelihood = -272.76543                  Prob > chi2        = 0.0000

------------------------------------------------------------------------------
             |               Robust
             | Haz. ratio   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Diabetes     |
         age |   .9693495   .0293552    -1.03   0.304     .9134885    1.028626
             |
        male |
        Yes  |   .8021755   .2273265    -0.78   0.437     .4603091    1.397942
             |
   community |
    Jackson  |   1.549902   .6274179     1.08   0.279     .7010166    3.426733
Minneapolis  |   .9649113   .3361108    -0.10   0.918     .4875122    1.909806
 Washington  |    1.36829   .5112313     0.84   0.401     .6578786    2.845842
             |
        race |
      White  |   .4412767    .135994    -2.65   0.008     .2412044    .8073037
         bmi |   1.112781   .0314166     3.79   0.000     1.052878    1.176092
     glucose |   1.141379   .0304922     4.95   0.000     1.083153    1.202735
-------------+----------------------------------------------------------------
Hypertension |
         age |   .9945906   .0220662    -0.24   0.807     .9522686    1.038794
             |
        male |
        Yes  |   .6229044   .1403048    -2.10   0.036     .4005846    .9686091
             |
   community |
    Jackson  |    .606375   .1824113    -1.66   0.096     .3362643    1.093457
Minneapolis  |   .8873364   .2642854    -0.40   0.688     .4949546    1.590784
 Washington  |   .6548935   .1999546    -1.39   0.166     .3599802    1.191414
             |
        race |
      White  |    1.26674   .4058107     0.74   0.460     .6760798    2.373433
       sysbp |   1.072573   .0149785     5.02   0.000     1.043614    1.102336
       diabp |   1.025091   .0138294     1.84   0.066     .9983414    1.052558
------------------------------------------------------------------------------
Note: Standard error estimates may be more variable for small datasets and
      datasets with low proportions of interval-censored observations.
```

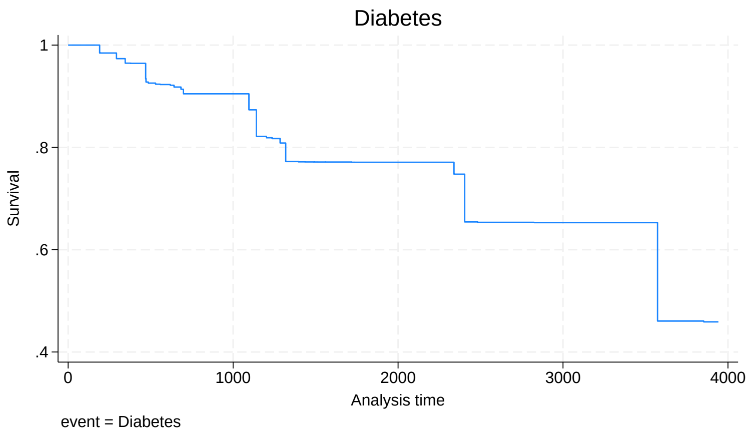
After fitting the marginal Cox model, we can use `stcurve` to plot the estimated survivor functions. By default, `stcurve` plots the survivor functions for both events as subgraphs within a single graph.

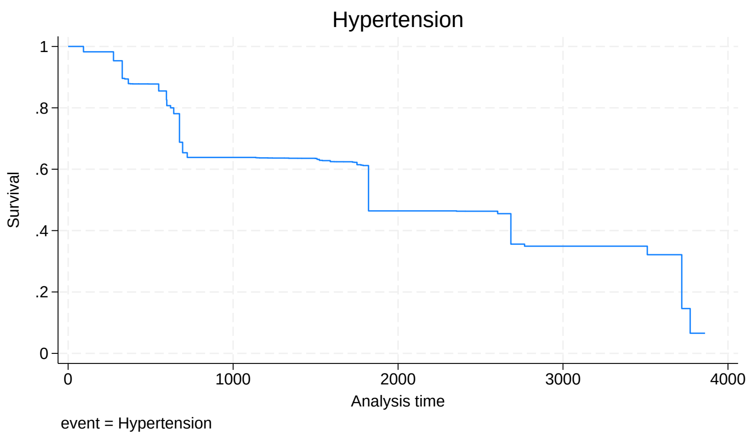
```
. stcurve, survival
note: function evaluated at overall means of covariates for specified events.
```



Because there are different covariates for each event, we do not want to compare those survivor functions directly. By adding the `sepevents` option, we can request that the estimated survivor function for each event be displayed on a separate graph. Additionally, we can use the `graph#opts()` options to modify the title of each graph.

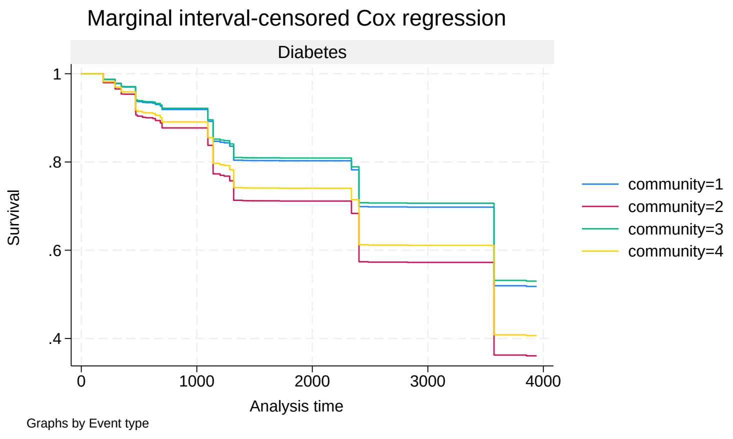
```
. stcurve, survival sepevents graph1opts(title("Diabetes"))
> graph2opts(title("Hypertension"))
note: function evaluated at overall means of covariates for specified events.
```





If we wish to compare the survival curve of an average person with diabetes across different communities, we can specify multiple values for `community` using the `at()` option and specify the event value label `"Diabetes"` in the `events()` option. Alternatively, we can specify the numerical value for diabetes (1) or its position (#1) in the `events()` option.

```
. stcurve, surv at(community=(1(1)4)) event("Diabetes")
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any) for specified event.
```



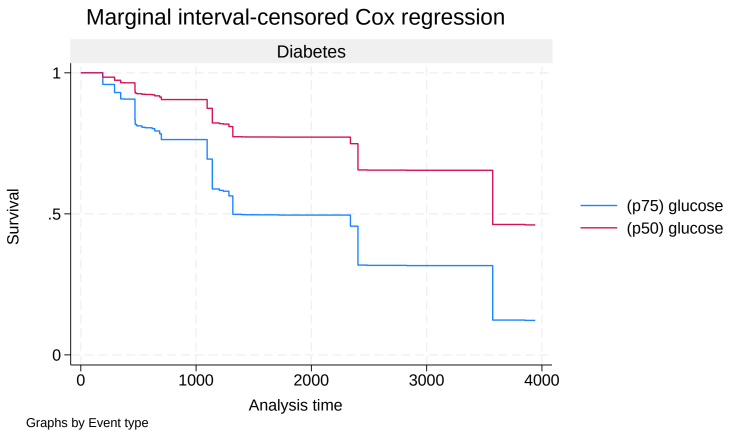
The above survival curves show that an average person in Forsyth (blue line) and one in Minneapolis (green line) have a similar higher risk of developing diabetes, whereas an average person in Washington (yellow line) and one in Jackson (red line) have a lower risk of developing diabetes.

Suppose we want to compare the survival curves for people whose glucose levels are at the 75th percentile with those for people whose levels are at the 50th percentile. The following `stcurve` command will result in an error because `glucose` is not a covariate in the hypertension model, but it is a covariate in the diabetes model.

```
. rcof "noi stcurve, surv at((p75) glucose) at((p50) glucose)" == 322
variable glucose not found in list of covariates
in option at() for event = 2
```

In this case, we need to use the events() option to specify for which event we would like to create a plot.

```
. stcurve, surv at((p75) glucose) at((p50) glucose) event(1)
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any) for specified event.
```



◁

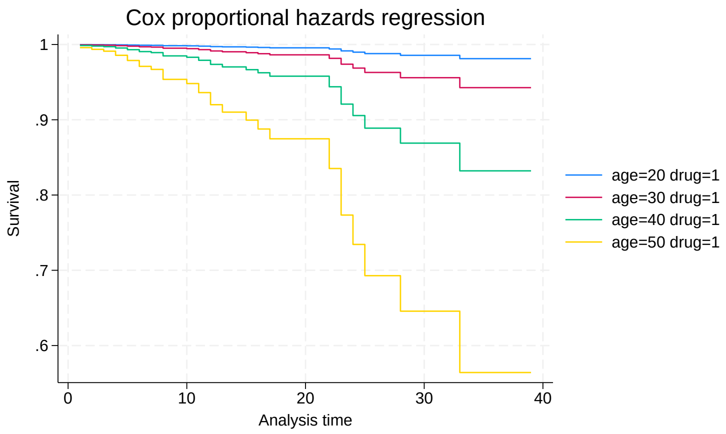
Using at() with stcurve

stcurve, by default, evaluates the function by setting each covariate to its mean value. The at() option specifies that the function be evaluated at the values of the covariates listed in at() and at the means of all unlisted covariates. You can repeat the at() option to produce multiple curves corresponding to different sets of covariate values.

▷ Example 8

Let's return to example 1. Suppose that we want to compare the survival curves for patients at ages 20, 30, 40, and 50 of the treatment group. The easiest way to do this is to specify multiple values for age in `at(numlist)`.

```
. use https://www.stata-press.com/data/r19/drugtr
(Patient survival in drug trial)
. quietly stcox age drug
. stcurve, survival at(age=(20(10)50) drug=1)
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any).
```

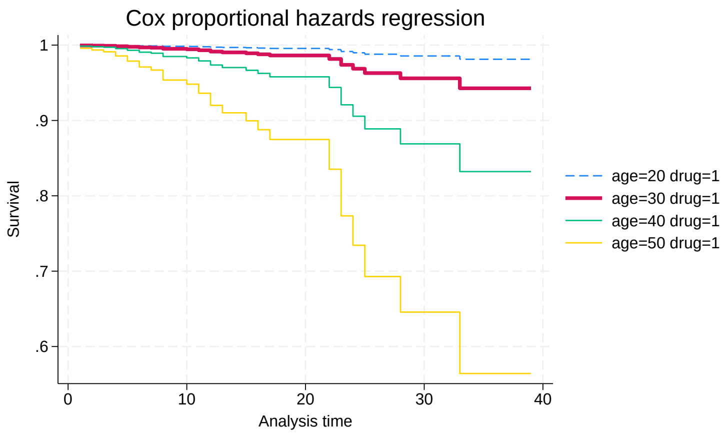


We could have obtained the same plot by specifying the `at#()` options but with more typing:

```
stcurve, survival at1(age=20 drug=1) at2(age=30 drug=1) ///
    at3(age=40 drug=1) at4(age=50 drug=1)
```

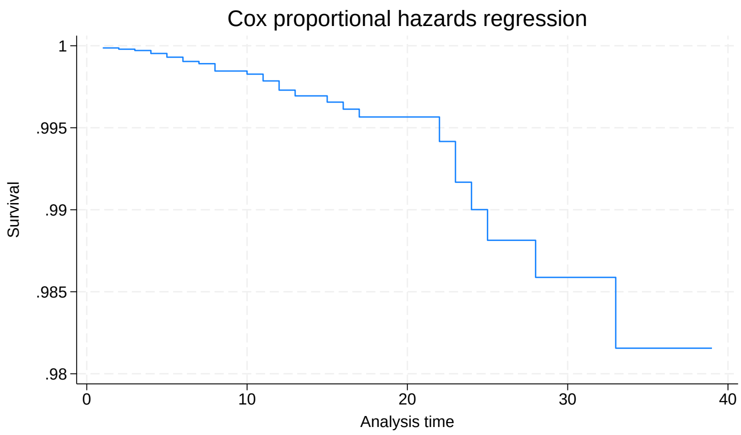
We can change the look of an individual at() plot by using atplot#opts():

```
. stcurve, survival at(age=(20(10)50) drug=1)
> atplot1opts(lpattern(dash)) atplot2opts(lwidth(thick))
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any).
```



The at() option provides many other flexible specifications. For example, if we would like to plot the baseline survivor function, we do not need to set every covariate to zero in at(). We can set all covariates to zero at once as follows:

```
. stcurve, survival at((zero) _all)
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any).
```



For details about the at() option, see *Syntax of at()* in [ST] ***adjustfor_option***.

◁

References

Cefalu, M. S. 2011. Pointwise confidence intervals for the covariate-adjusted survivor function in the Cox model. *Stata Journal* 11: 64–81.

Ruhe, C. 2016. Estimating survival functions after stcox with time-varying coefficients. *Stata Journal* 16: 867–879.

Also see

[ST] **stcox** — Cox proportional hazards model

[ST] **stcox postestimation** — Postestimation tools for stcox

[ST] **stcrreg** — Competing-risks regression

[ST] **stcrreg postestimation** — Postestimation tools for stcrreg

[ST] **stintcox** — Cox proportional hazards model for interval-censored survival-time data

[ST] **stintcox postestimation** — Postestimation tools for stintcox

[ST] **stintreg** — Parametric models for interval-censored survival-time data

[ST] **stintreg postestimation** — Postestimation tools for stintreg

[ST] **stmgintcox** — Marginal Cox PH model for interval-censored multiple-event data

[ST] **stmgintcox postestimation** — Postestimation tools for stmgintcox

[ST] **streg** — Parametric survival models

[ST] **streg postestimation** — Postestimation tools for streg

[ST] **sts** — Generate, graph, list, and test the survivor and related functions

[ST] **stset** — Declare data to be survival-time data

[ST] ***adjustfor_option*** — Adjust survivor and related functions for covariates at specific values

[ME] **mestreg** — Multilevel mixed-effects parametric survival models

[ME] **mestreg postestimation** — Postestimation tools for mestreg

[XT] **xtstreg** — Random-effects parametric survival models

[XT] **xtstreg postestimation** — Postestimation tools for xtstreg