

Postestimation commands

The following postestimation commands are of special interest after `stintcox`:

Command	Description
<code>estat gofplot</code>	produce goodness-of-fit plot
<code>stcurve</code>	plot the survivor, failure, hazard, or cumulative hazard function
<code>stintcoxnp</code>	plot nonparametric curves and Cox predicted curves
<code>stintphplot</code>	plot $-\ln\{-\ln(\text{survival})\}$ curves

The following standard postestimation commands are also available:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of parameters
<code>estat ic</code>	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>hausman</code>	Hausman's specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of parameters
<code>lrtest</code>	likelihood-ratio test
<code>margins</code>	marginal means, predictive margins, marginal effects, and average marginal effects
<code>marginsplot</code>	graph the results from margins (profile plots, interaction plots, etc.)
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
<code>predict</code>	hazard ratios, survivor functions, influence statistics, residuals, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of parameters
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates new variables containing predictions such as hazard ratios, linear predictions, standard errors, and baseline survivor and baseline cumulative hazard functions.

Menu for predict

Statistics > Postestimation

Syntax for predict

Single-record-per-subject interval-censored data with baseline covariates

```
predict [type] newvar [if] [in] [ , statistic]

predict [type] { stub* | newvarl newvaru } [if] [in] , statistic2
```

Single-record-per-subject interval-censored data with time-varying covariates

```
predict [type] newvar [if] [in] [ , mgale]

predict [type] { stub* | newvarl newvaru } [if] [in] [ , statistic statistic2]
```

Multiple-record-per-subject interval-censored data

```
predict [type] newvar [if] [in] [ , statistic statistic2]
```

For single-record-per-subject data, predictions that depend on time, *statistic2*, are calculated for both the lower endpoint t_l and the upper endpoint t_u of the time interval specified in `stintcox's interval()` option and thus are stored in two new variables. This also applies to each *statistic* except *mgale* in the presence of time-varying covariates. For multiple-record-per-subject data, predictions are calculated for the examination times specified in `stintcox's time()` option and are always stored in a single variable.

statistic	Description
Main	
hr	predicted hazard ratio, also known as the relative hazard; the default
xb	linear prediction $\mathbf{x}_j\hat{\beta}$
stdp	standard error of the linear prediction, $SE(\mathbf{x}_j\hat{\beta})$
* mgale	martingale-like residuals
statistic2	Description
Main	
* basesurv	baseline survivor function
* basechazard	baseline cumulative hazard function
* basehc	baseline hazard contributions
* csnell	Cox–Snell-like residuals

Unstarred statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample.

Options for predict

Main

`hr`, the default, calculates the relative hazard (hazard ratio), that is, the exponentiated linear prediction, $\exp(\mathbf{x}_j\hat{\boldsymbol{\beta}})$. After `stintcox` with the `tvc()` option, it calculates $\exp\{\mathbf{x}_j\hat{\boldsymbol{\beta}} + g(t)\mathbf{z}_j\hat{\boldsymbol{\gamma}}\}$, where $g(t)$ is a function of time as specified in `stintcox`'s option `texp()`.

`xb` calculates the linear prediction from the fitted model. That is, you fit the model by estimating a set of parameters, $\beta_1, \beta_2, \dots, \beta_p$, and the linear prediction is $\hat{\beta}_1 x_{1j} + \hat{\beta}_2 x_{2j} + \dots + \hat{\beta}_p x_{pj}$, often written in matrix notation as $\mathbf{x}_j\hat{\boldsymbol{\beta}}$.

The $x_{1j}, x_{2j}, \dots, x_{pj}$ used in the calculation are obtained from the data currently in memory and need not correspond to the data on the independent variables used in estimating $\boldsymbol{\beta}$.

After `stintcox` with `tvc()`, it calculates $\mathbf{x}_j\hat{\boldsymbol{\beta}} + g(t)\mathbf{z}_j\hat{\boldsymbol{\gamma}}$, where $g(t)$ is a function of time as specified in `stintcox`'s `texp()` option.

`stdp` calculates the standard error of the linear prediction `xb`.

`basesurv` calculates the baseline survivor function.

`basechazard` calculates the baseline cumulative hazard.

`basehc` calculates the baseline hazard contributions.

`mgale` calculates interval-censored martingale-like residuals, which are an interval-censored version of martingale residuals for right-censored data.

`csnell` calculates the Cox–Snell-like residuals, which are the estimates of the cumulative hazard function obtained from the fitted model.

margins

Description for margins

margins estimates margins of response for hazard ratios and linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

```
margins [marginlist] [ , options ]
margins [marginlist] , predict(statistic ...) [predict(statistic ...) ...] [options]
```

statistic	Description
hr	hazard ratio, also known as the relative hazard
xb	linear prediction
stdp	not allowed with margins
basesurv	not allowed with margins
basechazard	not allowed with margins
basehc	not allowed with margins
mgale	not allowed with margins
csnell	not allowed with margins

Statistics not allowed with margins are functions of stochastic quantities other than $e(b)$.

For the full syntax, see [R] margins.

Remarks and examples

Remarks are presented under the following headings:

- Baseline functions
- Residuals and diagnostic measures
- Postestimation after stintcox with the tvc() option
- Survivor curves for multiple-record-per-subject data with time-varying covariates

Baseline functions

predict after stintcox is used to generate a new variable or variables containing predicted values or residuals. predict can generate predicted hazard ratios, linear predictions, and standard errors of the linear predictions. It can also predict the baseline survivor function, baseline cumulative hazard function, or baseline hazard contributions. Baseline functions refer to the values of the functions when all covariates are set to 0. If the dataset is single-record-per-subject data, it calculates those statistics for both interval endpoints t_l and t_u specified in the interval() option with stintcox.

► Example 1: Baseline survivor function

We use data from [example 1](#) of [\[ST\] stintcox](#) and fit the Cox proportional hazards model in which the time to HIV infection depends on centered age variable (`age_mean`) and whether a subject has been in jail at the time of recruitment (`jail`):

```
. use https://www.stata-press.com/data/r19/idu
(Modified Bangkok IDU Preparatory Study)

. stintcox age_mean i.jail, interval(ltime rtime)
note: using adaptive step size to compute derivatives.

Performing EM optimization (showing every 100 iterations):
Iteration 0:   Log likelihood = -1086.2564
Iteration 100: Log likelihood = -600.55818
Iteration 200: Log likelihood = -600.47678
Iteration 295: Log likelihood = -600.46551

Computing standard errors: ..... done

Interval-censored Cox regression      Number of obs      =   1,124
Baseline hazard: Reduced intervals          Uncensored      =         0
                                           Left-censored    =         41
                                           Right-censored   =        991
                                           Interval-cens.   =         92

Event-time interval:
  Lower endpoint: ltime
  Upper endpoint: rtime
                                           Wald chi2(2)      =   14.07
                                           Prob > chi2       =   0.0009

Log likelihood = -600.46551
```

	Haz. ratio	OPG std. err.	z	P> z	[95% conf. interval]	
age_mean	.9647236	.0122201	-2.84	0.005	.9410676	.9889743
jail Yes	1.819721	.3804566	2.86	0.004	1.207927	2.741379

Note: Standard error estimates may be more variable for small datasets and datasets with low proportions of interval-censored observations.

We can now estimate, for instance, the baseline survivor function. For this dataset, estimates of baseline survivor function, as well as baseline cumulative hazard and baseline hazard contributions, are intervals. So, to compute these statistics, we can either specify two new variable names with `predict` or specify a stub *stub** that will automatically create two new variable names *stub1* and *stub2*; the first one will contain statistics computed using the lower time endpoint, and the second one will contain statistics computed using the upper time endpoint.

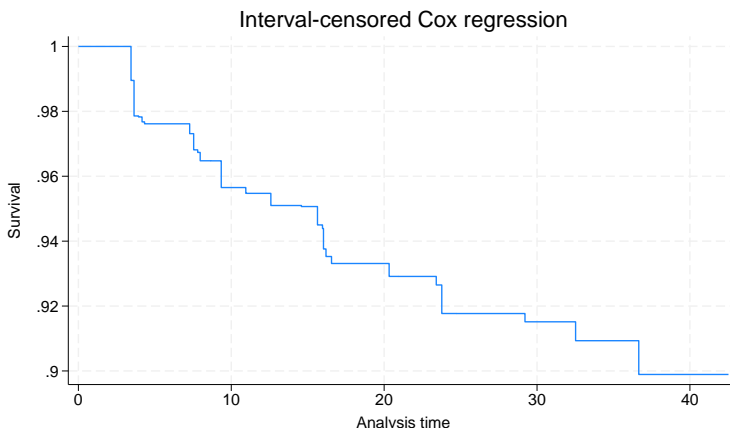
```
. predict bs_l bs_u, basesurv
. list bs_l bs_u ltime rtime age jail in 300/310
```

	bs_l	bs_u	ltime	rtime	age	jail
300.	.8989196	0	40.00	.	36	Yes
301.	.9547279	.9450011	11.97	15.84	21	Yes
302.	.9647679	.9506636	8.20	15.18	36	Yes
303.	.8989196	0	39.93	.	40	Yes
304.	.8989196	0	39.48	.	25	Yes
305.	.8989196	0	36.72	.	40	Yes
306.	.8989196	0	39.93	.	40	Yes
307.	.9767599	0	4.30	.	34	Yes
308.	.8989196	0	39.74	.	42	No
309.	.8989196	0	37.61	.	30	Yes
310.	.8989196	0	39.97	.	28	Yes

Listed above are the baseline survivor functions, `bs_l` and `bs_u`, evaluated at the lower and upper time endpoints `ltime` and `rtime`, for subjects 300 to 310.

To graph the baseline survival curve, we can use `stcurve` with all covariates set to 0. We can also graph survivor functions after `stintcox` with covariates set to any value. See [\[ST\] stcurve](#) for more information.

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



Residuals and diagnostic measures

For right-censored data, several types of residuals have been introduced to assess the appropriateness of the Cox proportional hazards model; see *Residuals and diagnostic measures* in [ST] [stcox postestimation](#) for details.

Farrington (2000) proposed extensions of those residuals for interval-censored data; see the reference for applications and a discussion of limitations of the residuals for interval-censored data. Here we offer martingale-like residuals and Cox–Snell-like residuals for visual model checking. Martingale-like residuals are useful in determining the functional form of covariates to be included in the model. They are also useful in assessing whether some covariates are needed in the model and for identifying outliers. Cox–Snell-like residuals are useful in assessing the overall model fit. If the model fits the data, those residuals should approximate an interval-censored sample from the standard exponential distribution. These residuals can be plotted by using `estat gofplot`.

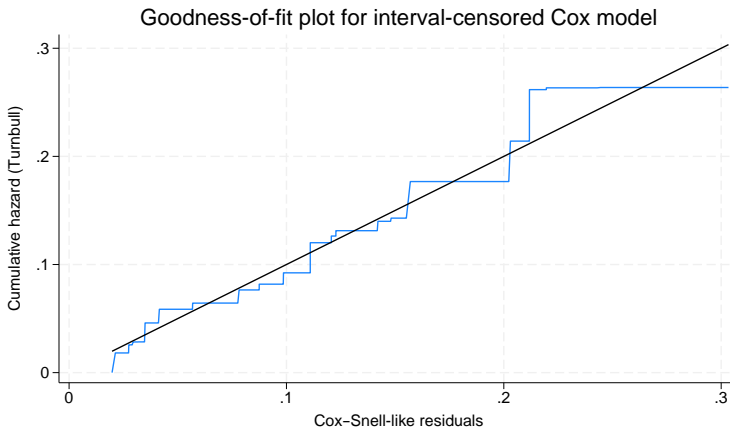
► Example 2: Assess overall model fit

To visually assess the overall model fit, we can use the Cox–Snell-like residuals. For right-censored data, Cox and Snell (1968) argued that if the correct model has been fit to the data, these residuals should have a censored standard exponential distribution. With interval-censored, Cox–Snell-like residuals approximate an interval-censored sample from this distribution.

`estat gofplot` calculates an empirical estimate of the cumulative hazard function based on the Cox–Snell-like residuals and plots the resulting cumulative hazard rate against the residuals themselves. If the model fits the data, the plot is expected to approximate a straight line with slope 1.

Continuing with [example 1](#), let's produce the goodness-of-fit plot.

```
. estat gofplot
```



The goodness-of-fit plot shows that the jagged line stays very close to the 45° reference line. The Cox proportional hazards model appears to fit the data well.

➤ Example 3: Martingale-like residuals

Martingale-like residuals may be used as a diagnostic tool to assess the functional form of a covariate to be included in a Cox model. To find the appropriate functional form of a variable, we fit a Cox model without the variable of interest and then plot a lowess smooth of the martingale-like residuals against some transformation of that variable. If the transformation is appropriate, then the smooth should be approximately linear.

Continuing with [example 1](#), we can apply this procedure to check whether the functional form of the covariate `age_mean` is appropriate. First, we refit the Cox proportional hazards model on whether a subject has been in jail at the time of recruitment (`jail`). Then, we obtain the martingale-like residuals `mg` using `predict` with the `mgale` option. Last, we plot a lowess smooth of `mg` against variable `age_mean`.

```
. stintcox i.jail, interval(ltime rtime) nolog
note: using adaptive step size to compute derivatives.

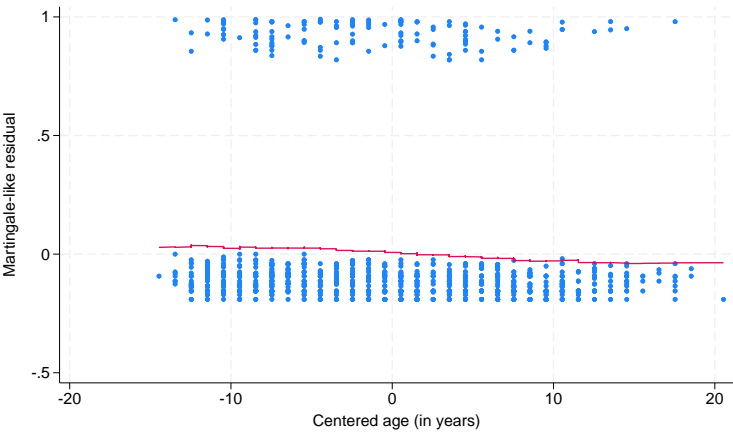
Interval-censored Cox regression      Number of obs      =   1,124
Baseline hazard: Reduced intervals      Uncensored      =       0
                                      Left-censored     =       41
Event-time interval:                  Right-censored    =     991
  Lower endpoint: ltime                  Interval-cens.    =       92
  Upper endpoint: rtime

                                      Wald chi2(1)        =     6.24
Log likelihood = -605.20197              Prob > chi2        =     0.0125
```

	OPG					
	Haz. ratio	std. err.	z	P> z	[95% conf. interval]	
jail						
Yes	1.675924	.346496	2.50	0.013	1.117556	2.513273

Note: Standard error estimates may be more variable for small datasets and datasets with low proportions of interval-censored observations.

```
. predict mg, mgale
. lowess mg age_mean, mean noweight title("") note("") m(o)
```



We used the `lowess` command with the `mean` and `noweight` options to obtain a plot of the running-mean smoother to ease interpretation. The smooth appears nearly linear, supporting the inclusion of the untransformed version of `age_mean` in our Cox model. Had the smooth not been linear, we would have tried smoothing the martingale residuals against various transformations of `age_mean` until we found one that produced a near-linear smooth.



Postestimation after stintcox with the `tvc()` option

When the `tvc()` option is used with `stintcox`, the model includes the time-varying covariates formed by covariates in the `tvc()` option multiplied by the deterministic function of time specified in the `texp()` option. In this case, predictions that depend on the covariate pattern, such as a linear predictor and relative hazard, also depend on time. For instance, for single-record-per-subject data, `predict`, `xb` will now produce two new variables: one containing the linear predictor computed using the lower endpoint of the time interval and the other one computed using the upper endpoint.

After `stintcox`, `tvc()`, postestimation features such as `predict` and `stcurve` incorporate both fixed and time-varying covariates. For instance, the linear predictor will include the effects of both baseline covariates and time-varying covariates produced by the `tvc()` option.

Predictions of baseline functions and martingale-like residuals do not depend on covariates and are thus unaffected by `stintcox`, `tvc()`.

▷ Example 4: Predictions after stintcox with `tvc()`

In *Testing the proportional-hazards assumption using the `tvc()` option* in [ST] `stintcox`, we discovered that `jail` was one of the variables that did not satisfy the proportional-hazards assumption.

Let's fit the following model with a time-varying covariate `jail × t`:

$$h(t) = h_0(t) \exp\{\beta_1 \times \text{age_mean} + \beta_2 \times \text{jail} + \gamma_1 \times (\text{jail} \times t)\}$$

We use the `tvc()` option to create the time-varying covariate `jail × t`:

```
. stintcox age_mean i.jail, interval(ltime rtime) tvc(i.jail) nohr
note: using adaptive step size to compute derivatives.
Performing EM optimization (showing every 100 iterations):
Iteration 0:   Log likelihood = -1086.2564
Iteration 100: Log likelihood = -597.6072
Iteration 200: Log likelihood = -597.52358
Iteration 295: Log likelihood = -597.51247

Computing standard errors: ..... done

Interval-censored Cox regression      Number of obs      =   1,124
Baseline hazard: Reduced intervals      Uncensored         =     0
                                         Left-censored      =    41
Event-time interval:                  Right-censored     =   991
  Lower endpoint: ltime                Interval-cens.    =    92
  Upper endpoint: rtime

                                         Wald chi2(3)       =   16.87
Log likelihood = -597.51247            Prob > chi2        =  0.0008
```

		OPG				
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
main						
age_mean	-.0360376	.0126271	-2.85	0.004	-.0607864	-.0112889
jail						
Yes	-.1593675	.3777957	-0.42	0.673	-.8998335	.5810986
tvc						
jail						
Yes	.0732193	.0342545	2.14	0.033	.0060818	.1403569

Notes: Standard error estimates may be more variable for small datasets and datasets with low proportions of interval-censored observations.
Variables in `tvc` equation interacted with `_t`.

Wald test that `[tvc] = 0`: `chi2(1) = 4.5690` `Prob > chi2 = 0.0326`

Let's use `predict`, `xb` to calculate the linear prediction. Because the linear prediction now depends on time and there are two time interval endpoints in a single-record-per-subject data, `predict` requires two new variable names. Below, we use a stub notation `xb*` that automatically creates two new variables, `xb1` and `xb2`, to store linear predictions. To demonstrate, we list the results for subjects 30 and 95.

```
. predict xb*, xb
(991 missing values generated)
note: prediction includes time-varying covariates formed by stintcox, tvc().
. list age_mean jail ltime rtime xb1 xb2 if _n==30 | _n==95
```

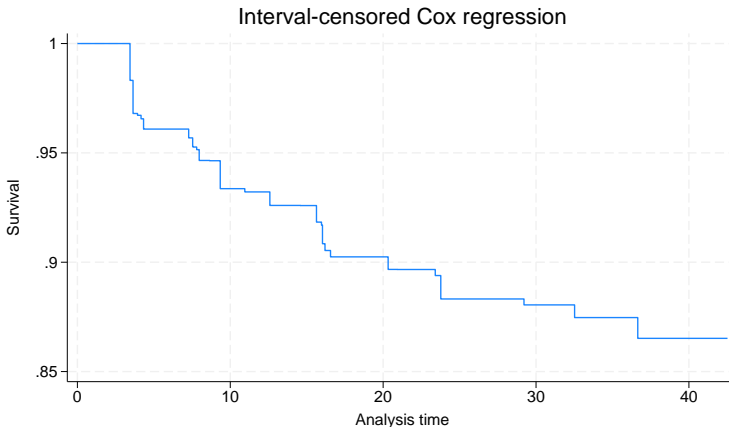
	age_mean	jail	ltime	rtime	xb1	xb2
30.	.53825623	Yes	7.147541	13.213115	.3445733	.7886905
95.	-.46174377	No	4.1311474	8.0327873	.0166402	.0166402

For subject 30, `jail` is equal to 1, and the linear predictions in `xb1` are computed as $\hat{\beta}_1 \times \text{age_mean} + \hat{\beta}_2 + \hat{\gamma}_1 \times \text{ltime}$ and in `xb2` as $\hat{\beta}_1 \times \text{age_mean} + \hat{\beta}_2 + \hat{\gamma}_1 \times \text{rtime}$. For subject 95, `jail` = 0, and thus both `xb1` and `xb2` contain $\hat{\beta}_1 \times \text{age_mean}$.

► Example 5: stcurve after stintcox with tvc()

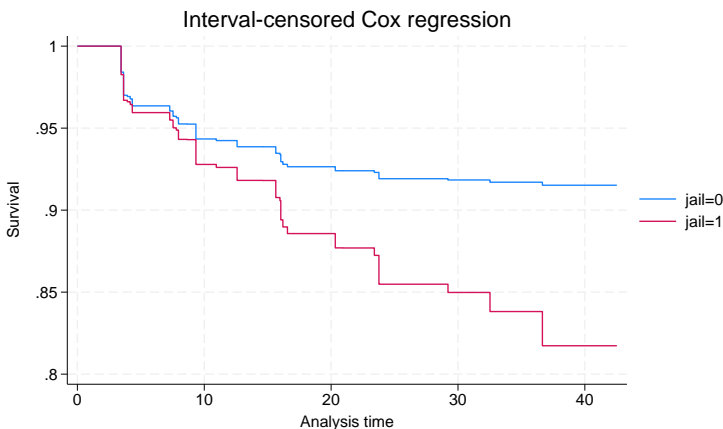
stcurve can be used after stintcox to plot the estimated survivor, failure, hazard, or cumulative hazard function. When the tvc() option is specified with stintcox, stcurve incorporates the time-varying nature of the covariates. By default, a function is evaluated at the overall means of covariates in the model. For covariates specified in the tvc() option, their overall mean values are multiplied by the function of time, as specified in stintcox's texp() option, during a function evaluation.

```
. stcurve, survival
note: function evaluated at overall means of covariates, multiplied by _t for
      covariates in stintcox's tvc() option.
```



Here jail is binary, so we may want to use the at() option to evaluate the survivor function for jail = 0 and jail = 1. The overall mean is used for any covariate not listed in at() (age_mean in our example).

```
. stcurve, survival at(jail=(0 1))
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any), multiplied by _t for
      covariates in stintcox's tvc() option.
```



The survival of those who have not been imprisoned is higher than those who have.



Survivor curves for multiple-record-per-subject data with time-varying covariates

By default, `stcurve` evaluates the survivor and other functions at the overall means of covariates. We can also use the `at()` option to evaluate functions at specific covariate values. Both of these options work well for baseline covariates. But multiple-record-per-subject data often include time-varying covariates. In that case, we may want to incorporate the time-varying nature of these covariates when plotting the survivor function or other functions. We can use the `attmeans` option to evaluate the function at time-specific means, which will vary with time for time-varying covariates. We can also specify our own time-varying values for all or some covariates by using the `atframe()` option.

In general, care should be taken when interpreting the survivor function in the presence of time-varying covariates. The interpretation is valid only for [external time-varying covariates](#), which are covariates that are external to an individual under study, such as air temperature. (Any covariate that is formed as an interaction of a baseline covariate and a deterministic function of time is an external covariate by construction.) For an [internal time-varying covariate](#), which is generated by an individual in a study, such as wellness status, the survivor function is not well defined. See, for instance, [Kalbfleisch and Prentice \(2002\)](#) for details.

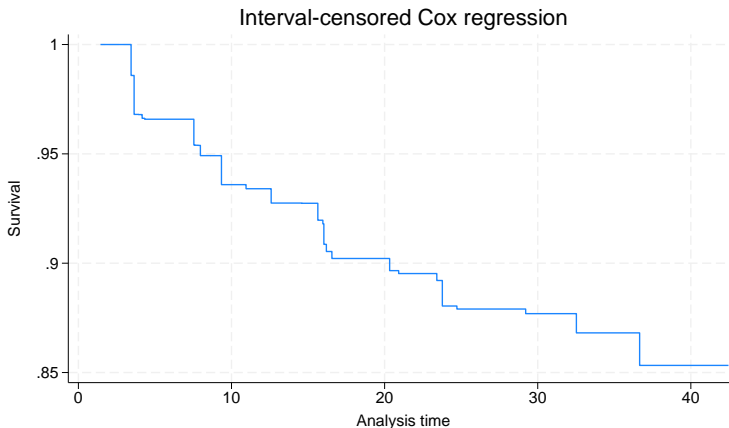
► Example 6: `stcurve` with option `attmeans`

In [example 4](#) in [\[ST\] stintcox](#), we fit a model that included the time-varying `jail_vary` variable. In this example, `jail_vary` is an internal time-varying covariate because it can be observed for a subject only until a subject “survives” and is not censored. Although it would not be sensible to proceed with this example in practice, we will use it here solely to demonstrate the software.

So, continuing with [example 4](#) in [\[ST\] stintcox](#), after fitting the model with time-varying `jail_vary`, we use `stcurve` with the `attmeans` option to plot the survivor curve that uses the time-specific mean values for covariates at each time point, hence accounting for the average changes of the covariates, specifically `jail_vary`, over time:

```
. use https://www.stata-press.com/data/r19/idu2, clear
(Modified Bangkok IDU Preparatory Study with time-varying variable jail_vary)
. stintcox age_mean i.male i.needle i.inject i.jail_vary, id(id) time(time)
> status(is_seropos)
(output omitted)
```

```
. stcurve, survival attmeans
note: function evaluated at time-specific means of covariates.
```



► Example 7: stcurve with option atframe()

We can use the `atframe()` option to specify time-varying covariate values to be used to evaluate the survivor function. We first need to create a frame that includes all or a subset of the covariates and the time variable in your model and fill in those variables with the values we want to use for prediction and the corresponding time points. If a subset of the covariates is specified in `atframe()`, the unspecified covariates are evaluated, by default, at their overall means or, if the `attmeans` option is specified, at their time-specific means.

Suppose we want to plot the survivor curve for an individual with the same covariate pattern as subject 2 in our dataset. We start by creating a new frame called `id2` and use `frame put` to copy the relevant information for subject 2 in this new frame. Then we `list` the data we just saved in frame `id2`.

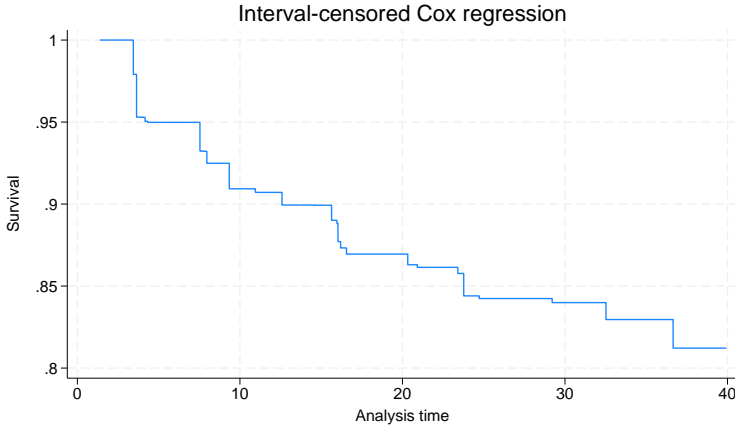
```
. frame put time age_mean male needle inject jail_vary if id==2, into(id2)
. frame id2: list
```

	time	age_mean	male	needle	inject	jail_v-y
1.	4.1311475	-6.4617438	Yes	No	Yes	Yes
2.	8.2622951	-6.4617438	Yes	No	Yes	No
3.	12.295082	-6.4617438	Yes	No	Yes	No
4.	16.065574	-6.4617438	Yes	No	Yes	No
5.	20.098361	-6.4617438	Yes	No	Yes	No
6.	24.262295	-6.4617438	Yes	No	Yes	No

In frame `id2`, six examination times are recorded, and all covariate values for each examination time are filled in. Variables `age_mean`, `male`, `needle`, and `inject` are constant, and variable `jail_vary` changes at the second examination time.

We can graph the survivor curve for this particular profile by typing

```
. stcurve, survival atframe(id2)
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any).
note: covariate values from frame id2 used to evaluate function.
```



Unless you provide covariate values for all distinct observed times used during estimation, `stcurve` will use the same imputation method as specified in the `tvcovimpute()` option with `stintcox` ([ST] [stintcox](#)) to impute the unobserved covariate values for time points that are not recorded in the frame.

◀

Methods and formulas

Methods and formulas are presented under the following headings:

Predictions for single-record interval-censored data
Predictions for multiple-record interval-censored data
Survivor curves for interval-censored data

Predictions for single-record interval-censored data

For single-record interval-censored data, each subject i contains one observation $(t_{li}, t_{ui}, \mathbf{x}_i)$, where t_{li} and t_{ui} define the observed time interval and \mathbf{x}_i is a $1 \times p$ vector of covariate values. Let $\hat{\beta}$ be a $p \times 1$ vector of estimates of regression coefficients for \mathbf{x}_i .

Case of baseline covariates. The survivor function for subject i at time t can be estimated as

$$\hat{S}_i(t) = \exp\left\{-\sum_{t_k \leq t} \hat{h}_k \exp(\mathbf{x}_i \hat{\beta})\right\}$$

where \hat{h}_k ($k = 1, \dots, m$) are the estimates of the baseline hazard jumps at t_1, \dots, t_m and $t_1 < \dots < t_m$ are the distinct time points for all $t_{li} > 0$ and $t_{ui} < \infty$.

`predict newvar`, *statistic* may be used after `stintcox` to predict various quantities, according to each following *statistic*:

`hr`:

$$newvar_i = \exp(\mathbf{x}_i \hat{\beta})$$

`xb`:

$$newvar_i = \mathbf{x}_i \hat{\beta}$$

`stdp`:

$$newvar_i = \widehat{\text{se}}(\mathbf{x}_i \hat{\beta})$$

`mgale`:

$$newvar_i = \frac{\hat{S}_i(t_{li}) \log \hat{S}_i(t_{li}) - \hat{S}_i(t_{ui}) \log \hat{S}_i(t_{ui})}{\hat{S}_i(t_{li}) - \hat{S}_i(t_{ui})}$$

For right-censored data, martingale residuals can be defined as the scores of the regression parameters. This property can carry over to the interval-censored data for martingale-like residuals `mgale`. Therefore, these residuals are expected to have mean zero and to be asymptotically uncorrelated. Furthermore, these residuals are orthogonal to variables included in the model. Thus, we can use them to assess the need to include other covariates in the model. See [Farrington \(2000\)](#) for details.

`predict newvarl newvaru`, *statistic* may be used after `stintcox` to predict a pair of quantities for each observation for both the lower and upper endpoints of the time interval (t_{li}, t_{ui}) , according to each following *statistic*:

`basehc`:

$$newvar_{li} = I(t_k = t_{li}) \hat{h}_k$$

$$newvar_{ui} = I(t_k = t_{ui}) \hat{h}_k$$

`basechazard`:

$$newvar_{li} = \hat{H}_0(t_{li}) = \sum_{k:t_k \leq t_{li}} \hat{h}_k$$

$$newvar_{ui} = \hat{H}_0(t_{ui}) = \sum_{k:t_k \leq t_{ui}} \hat{h}_k$$

`basesurv`:

$$newvar_{li} = \hat{S}_0(t_{li}) = \exp \{ -\hat{H}_0(t_{li}) \}$$

$$newvar_{ui} = \hat{S}_0(t_{ui}) = \exp \{ -\hat{H}_0(t_{ui}) \}$$

`csnell`:

$$newvar_{li} = -\log \hat{S}_i(t_{li})$$

$$newvar_{ui} = -\log \hat{S}_i(t_{ui})$$

The Cox–Snell-like residuals are the estimates of the cumulative hazard function obtained from the fitted model. They are computed separately for each of the two interval endpoints. For interval-censored data, under the correct model assumption, these residuals are expected to approximate an interval-censored sample from the standard exponential distribution. Therefore, they can be used for checking the overall model fit. Cox–Snell-like residuals can never be negative and therefore are not symmetric about zero. See [Farrington \(2000\)](#) for details.

Case of time-varying covariates with `tvvc()`. When the `tvvc()` option is specified with `stintcox`, `predict` modifies its calculations so that they include the time-varying components. Let \mathbf{z}_i denote a $1 \times q$ vector of covariates specified in the `tvvc()` option of `stintcox`, and let $g(t)$ denote the function of time specified in the `texp()` option of `stintcox`. Let a $q \times 1$ vector $\hat{\gamma}$ contain the estimates of regression coefficients for \mathbf{z}_i . The survivor function for subject i at time t can be estimated as

$$\hat{S}_i(t) = \exp \left[- \sum_{t_k \leq t} \hat{h}_k \exp \{ \mathbf{x}_i \hat{\beta} + g(t_k) \mathbf{z}_i \hat{\gamma} \} \right]$$

Also, each following *statistic* changes to predict a pair of quantities for each observation for both the lower and upper endpoints of the time interval (t_{li}, t_{ui}) :

hr:

$$newvar_{li} = \exp \{ \mathbf{x}_i \hat{\beta} + g(t_{li}) \mathbf{z}_i \hat{\gamma} \}$$

$$newvar_{ui} = \exp \{ \mathbf{x}_i \hat{\beta} + g(t_{ui}) \mathbf{z}_i \hat{\gamma} \}$$

xb:

$$newvar_{li} = \mathbf{x}_i \hat{\beta} + g(t_{li}) \mathbf{z}_i \hat{\gamma}$$

$$newvar_{ui} = \mathbf{x}_i \hat{\beta} + g(t_{ui}) \mathbf{z}_i \hat{\gamma}$$

stdp:

$$newvar_{li} = \widehat{\text{se}} \{ \mathbf{x}_i \hat{\beta} + g(t_{li}) \mathbf{z}_i \hat{\gamma} \} = \{ \mathbf{x}_i \widehat{\Sigma}_{\beta} \mathbf{x}_i' + 2g(t_{li}) \mathbf{z}_i \widehat{\Sigma}_{\beta, \gamma} \mathbf{x}_i' + g(t_{li})^2 \mathbf{z}_i \widehat{\Sigma}_{\gamma} \mathbf{z}_i' \}^{\frac{1}{2}}$$

$$newvar_{ui} = \widehat{\text{se}} \{ \mathbf{x}_i \hat{\beta} + g(t_{ui}) \mathbf{z}_i \hat{\gamma} \} = \{ \mathbf{x}_i \widehat{\Sigma}_{\beta} \mathbf{x}_i' + 2g(t_{ui}) \mathbf{z}_i \widehat{\Sigma}_{\beta, \gamma} \mathbf{x}_i' + g(t_{ui})^2 \mathbf{z}_i \widehat{\Sigma}_{\gamma} \mathbf{z}_i' \}^{\frac{1}{2}}$$

where $\widehat{\Sigma}_{\beta}$ and $\widehat{\Sigma}_{\gamma}$ are the estimated covariance matrices for $\hat{\beta}$ and $\hat{\gamma}$, respectively, and $\widehat{\Sigma}_{\beta, \gamma}$ is the estimated covariance matrix between $\hat{\beta}$ and $\hat{\gamma}$.

Predictions for multiple-record interval-censored data

For multiple-record interval-censored data, subject $i = 1, \dots, n$ may contain n_i observations. And each observation $j = 1, \dots, n_i$ for subject i consists of $(t_{ij}, \delta_{ij}, \mathbf{x}_{ij})$, where t_{ij} is the examination time, δ_{ij} denotes whether the event of interest occurs between $t_{i(j-1)}$ and t_{ij} ($t_{i(j-1)} = 0$ if $j = 1$), and \mathbf{x}_{ij} records covariate values at t_{ij} . Here \mathbf{x}_{ij} includes both baseline and time-varying covariates. For baseline covariates, $\mathbf{x}_{ij} = \mathbf{x}_i$. Without loss of generality, we assume that the examination times are sorted within the same subject. Let $t_1 < \dots < t_m$ be the distinct observed time points for all $t_{li} > 0$ and $t_{ui} < \infty$, where t_{li} and t_{ui} define the observed time interval, based on (t_{ij}, δ_{ij}) , that bracket the event of interest for subject i . The survivor function for subject i at the examination time t_{ij} can be estimated as

$$\hat{S}_i(t_{ij}) = \exp \left\{ - \sum_{t_k \leq t_{ij}} \hat{h}_k \exp(\mathbf{x}_{ik}^* \hat{\beta}) \right\}$$

where \hat{h}_k ($k = 1, \dots, m$) are the estimates of the baseline hazard jumps at t_1, \dots, t_m and \mathbf{x}_{ik}^* are the covariates' values for subject i at time t_k . For baseline covariates, $\mathbf{x}_{ik}^* = \mathbf{x}_{ij} = \mathbf{x}_i$. For time-varying covariates, if t_k happens to be t_{ij} , then $\mathbf{x}_{ik}^* = \mathbf{x}_{ij}$; otherwise, \mathbf{x}_{ik}^* is imputed using the imputation rule specified in the `tvvcovimpute()` option of `stintcox`.

`predict newvar, statistic` may be used after `stintcox` to predict various quantities for each observation, according to each following *statistic*:

hr:

$$newvar_{ij} = \exp(\mathbf{x}_{ij}\widehat{\boldsymbol{\beta}})$$

xb:

$$newvar_{ij} = \mathbf{x}_{ij}\widehat{\boldsymbol{\beta}}$$

stdp:

$$newvar_{ij} = \widehat{\text{se}}(\mathbf{x}_{ij}\widehat{\boldsymbol{\beta}})$$

mgale:

$$newvar_{ij} = \frac{\widehat{S}_i(t_{i(j-1)}) \log \widehat{S}_i(t_{i(j-1)}) - \widehat{S}_i(t_{ij}) \log \widehat{S}_i(t_{ij})}{\widehat{S}_i(t_{i(j-1)}) - \widehat{S}_i(t_{ij})}$$

basehc:

$$newvar_{ij} = I(t_k = t_{ij})\widehat{h}_k$$

basechazard:

$$newvar_{ij} = \widehat{H}_0(t_{ij}) = \sum_{k:t_k \leq t_{ij}} \widehat{h}_k$$

basesurv:

$$newvar_{ij} = \widehat{S}_0(t_{ij}) = \exp\{-\widehat{H}_0(t_{ij})\}$$

csnell:

$$newvar_{ij} = -\log \widehat{S}_i(t_{ij})$$

When the `tvc()` option is specified with `stintcox`, `predict` computes the linear prediction `xb` as $\mathbf{x}_{ij}\widehat{\boldsymbol{\beta}} + g(t_{ij})\mathbf{z}_{ij}\widehat{\boldsymbol{\gamma}}$ instead of $\mathbf{x}_{ij}\widehat{\boldsymbol{\beta}}$. Here $g(t)$ denotes the function of time specified in the `texp()` option of `stintcox`, and \mathbf{z} denotes the covariates specified in the `tvc()` option of `stintcox`. The survivor function is then estimated as

$$\widehat{S}_i(t_{ij}) = \exp\left[-\sum_{t_k \leq t_{ij}} \widehat{h}_k \exp\{\mathbf{x}_{ik}^*\widehat{\boldsymbol{\beta}} + g(t_k)\mathbf{z}_{ik}^*\widehat{\boldsymbol{\gamma}}\}\right]$$

where \mathbf{z}_{ik}^* are defined similarly to \mathbf{x}_{ik}^* above but using the covariates \mathbf{z}_{ij} 's.

Survivor curves for interval-censored data

You can use the `stcurve` command to plot the survivor function after `stintcox`. `stcurve` can also plot the failure, hazard, or cumulative hazard function.

Single-record interval-censored data. By default, `stcurve` plots the estimated survivor function with covariates set to their mean values,

$$\widehat{S}(t) = \exp\left\{-\sum_{t_k \leq t} \widehat{h}_k \exp(\bar{\mathbf{x}}\widehat{\boldsymbol{\beta}})\right\} = \widehat{S}_0(t) \exp(\bar{\mathbf{x}}\widehat{\boldsymbol{\beta}})$$

where $\bar{\mathbf{x}}$ is a $1 \times p$ mean vector of \mathbf{x} . When the `tvc()` option is specified with `stintcox`, the plotted survivor function becomes

$$\hat{S}(t) = \exp \left[- \sum_{t_k \leq t} \hat{h}_k \exp \{ \bar{\mathbf{x}} \hat{\boldsymbol{\beta}} + g(t_k) \bar{\mathbf{z}} \hat{\boldsymbol{\gamma}} \} \right]$$

where $\bar{\mathbf{z}}$ is a $1 \times q$ mean vector of \mathbf{z} . Alternatively, you can specify your own values in `stcurve`'s `at()` option. For covariates that are not specified in the `at()` option, the mean values are used. For details about the `at()` option, see [Syntax of at\(\)](#) in [ST] [adjustfor_option](#).

Multiple-record interval-censored data. By default, `stcurve` plots the estimated survivor function with covariates set to their overall mean values:

$$\hat{S}(t) = \exp \left\{ - \sum_{t_k \leq t} \hat{h}_k \exp(\bar{\mathbf{x}} \hat{\boldsymbol{\beta}}) \right\} = \hat{S}_0(t) \exp(\bar{\mathbf{x}} \hat{\boldsymbol{\beta}})$$

And with the `tvc()` option of `stintcox`, the estimated survivor function is computed by default as

$$\hat{S}(t) = \exp \left[- \sum_{t_k \leq t} \hat{h}_k \exp \{ \bar{\mathbf{x}} \hat{\boldsymbol{\beta}} + g(t_k) \bar{\mathbf{z}} \hat{\boldsymbol{\gamma}} \} \right]$$

Let $\bar{\mathbf{x}}_{t_k}$ be a $1 \times p$ mean vector of \mathbf{x}_{ik}^* over all subjects at time t_k and $\bar{\mathbf{z}}_{t_k}$ be a $1 \times q$ mean vector of \mathbf{z}_{ik}^* over all subjects at time t_k (if the `tvc()` option is specified with `stintcox`). When the `attmeans` option is specified, `stcurve` plots the following estimated survivor function:

$$\hat{S}(t) = \exp \left[- \sum_{t_k \leq t} \hat{h}_k \exp \{ \bar{\mathbf{x}}_{t_k} \hat{\boldsymbol{\beta}} + g(t_k) \bar{\mathbf{z}}_{t_k} \hat{\boldsymbol{\gamma}} \} \right]$$

If `stcurve`'s `atframe()` option is specified, $\bar{\mathbf{x}}_{t_k}$ and $\bar{\mathbf{z}}_{t_k}$ in the above will be replaced with $\mathbf{x}_{t_k}^{\text{new}}$ and $\mathbf{z}_{t_k}^{\text{new}}$, where $\mathbf{x}_{t_k}^{\text{new}}$ and $\mathbf{z}_{t_k}^{\text{new}}$ are user-specified values if time t_k and the corresponding covariate values are recorded in the frame. Otherwise, $\mathbf{x}_{t_k}^{\text{new}}$ and $\mathbf{z}_{t_k}^{\text{new}}$ are imputed using the imputation rule specified in the `tvcovimpute()` option of `stintcox`. Then the survivor function can be estimated as

$$\hat{S}(t) = \exp \left[- \sum_{t_k \leq t} \hat{h}_k \exp \{ \mathbf{x}_{t_k}^{\text{new}} \hat{\boldsymbol{\beta}} + g(t_k) \mathbf{z}_{t_k}^{\text{new}} \hat{\boldsymbol{\gamma}} \} \right]$$

The failure, hazard, and cumulative hazard functions are calculated based on the survivor function above.

References

- Cox, D. R., and E. J. Snell. 1968. A general definition of residuals (with discussion). *Journal of the Royal Statistical Society, B ser.*, 30: 248–275. <https://doi.org/10.1111/j.2517-6161.1968.tb00724.x>.
- Farrington, C. P. 2000. Residuals for proportional hazards models with interval-censored survival data. *Biometrics* 56: 473–482. <https://doi.org/10.1111/j.0006-341X.2000.00473.x>.
- Kalbfleisch, J. D., and R. L. Prentice. 2002. *The Statistical Analysis of Failure Time Data*. 2nd ed. New York: Wiley.

Also see

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

[ST] [PH plots \(interval-censored\)](#) — PH-assumption plots for interval-censored data

[ST] [stcurve](#) — Plot the survivor or related function after streg, stcox, and more

[U] [20 Estimation and postestimation commands](#)

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

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

