stmgintcox postestimation — Postestimation tools for stmgintcox[+]

[+]Postestimation features after `stmgintcox` are part of StataNow.

Postestimation commands

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

Command	Description
`estat gofplot`	produce goodness-of-fit plot
`stcurve`	plot the survivor, failure, hazard, or cumulative hazard function
`stintcoxnp`	plot nonparametric curves and Cox predicted curves
`stintphplot`	plot $-\ln\{-\ln(\text{survival})\}$ curves
`estat common`	estimate average effects of covariates across all events

The following standard postestimation commands are also available:

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

Menu for predict

Statistics > Postestimation

Syntax for predict

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

```
predict [type] newvar [if] [in] [, statistic event(eventspec)]

predict [type] {stub* | newvar_l newvar_u} [if] [in] [, statistic2 event(eventspec)]
```

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

```
predict [type] newvar [if] [in] [, mgale event(eventspec)]

predict [type] {stub* | newvar_l newvar_u} [if] [in]
    [, [hr | xb | stdp | statistic2] event(eventspec)]
```

Multiple-record-per-event interval-censored data

```
predict [type] newvar [if] [in] [, statistic statistic2 event(eventspec)]
```

Equation-specific predictions

```
predict [type] newvar [if] [in] [, [hr | xb | stdp] equation(eqno)]
```

For single-record-per-event data, we use the interval() option with stmgintcox to specify the lower and upper endpoints for the event-time interval. With these 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 and thus are stored in two new variables. This also applies to each *statistic* in the presence of time-varying covariates, except for mgale. For multiple-record-per-event data, we use the time() option with stmgintcox to specify the examination time variable. With these data, predictions are calculated for the examination times in the time variable 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
stdp	standard error of the linear prediction
* 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 for the jth event, $\exp(\mathbf{x}_j\widehat{\boldsymbol{\beta}}_j)$, where $\mathbf{x}_j$ are the corresponding covariates for the jth event and $\widehat{\boldsymbol{\beta}}_j$ are the estimated coefficients for the jth event. After `stmgintcox` with the `tvc()` option, `hr` calculates $\exp\{\mathbf{x}_j\widehat{\boldsymbol{\beta}}_j + g_j(t)\mathbf{z}_j\widehat{\gamma}_j\}$, where $g_j(t)$ is a function of time as specified in the `texp()` option with `stmgintcox` for the jth event and $\mathbf{z}_j$ are the corresponding time-varying covariates specified in the `tvc()` option for the jth event.

`xb` calculates the linear prediction from the fitted model for the jth event, often written in matrix notation as $\mathbf{x}_j\widehat{\boldsymbol{\beta}}_j$.

After `stmgintcox` with the `tvc()` option, it calculates $\mathbf{x}_j\widehat{\boldsymbol{\beta}}_j + g_j(t)\mathbf{z}_j\widehat{\gamma}_j$, where $g_j(t)$ is a function of time as specified in `stmgintcox`'s option `texp()` for the jth event.

`stdp` calculates the standard error of the linear prediction `xb` for each event.

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

`basesurv` calculates the baseline survivor function for each event.

`basechazard` calculates the baseline cumulative hazard function for each event.

`basehc` calculates the baseline hazard contributions for each event.

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

`event(`*eventspec*`)` specifies to which event you are referring. *eventspec* is the numeric value for the event, the corresponding label, as defined in the value label for the event variable, or one of `#1`, `#2`, ... with `#1` meaning the first event, `#2` meaning the second event, etc. If *eventspec* is a label, it should be enclosed in double quotes. For example, suppose our `event` variable takes on values `1` and `2`, with corresponding labels `"diabetes"` and `"hypertension"`. We can refer to event 1 by specifying `event(1)`, `event("diabetes")`, or `event(#1)`.

When the `tvc()` option is specified for a particular event i, the prediction includes time-varying covariates formed by `tvc()`. If you do not specify `event()`, the results are the same as if you specified all events.

The following option does not appear in the dialog box:

`equation(`*eqno*`)` specifies to which equation you are referring. `equation(#1)` would mean that the calculation is to be made for the first equation, `equation(#2)` would mean the second, and so on. You could also refer to the equations by their names. `equation(diabetes)` would refer to the equation named `diabetes`, and `equation(hypertension)` to the equation named `hypertension`.

When you use `equation()`, the prediction will not include time-varying covariates formed by the `tvc()` option. For example, suppose that we are modeling one event labeled `"diabetes"` and one labeled `"hypertension"`, and that we include time-varying covariates for both events using `tvc()`. We would then have four equations: `diabetes`, `tvc_diabetes`, `hypertension`, and `tvc_hypertension`. Or, equivalently, equations `#1`, `#2`, `#3`, and `#4`. `equation()` allows you to obtain predictions specifically for any one of these equations. If you would like to obtain predictions for an event that include the time-varying covariates formed by `tvc()`, use the `event()` option.

`equation()` can be specified only with one of the following statistics: `hr`, `xb`, or `stdp`.

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`
`mgale`	not allowed with `margins`
`basesurv`	not allowed with `margins`
`basechazard`	not allowed with `margins`
`basehc`	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**.

estat

Description for estat common

`estat common` estimates the average effect of a covariate on all event times and tests whether it is zero. When the effect of a covariate is the same for different events, the estimated average effect is a common effect. When the effect of a covariate is similar for different events, the reported test is more powerful than the classic multivariate Wald test.

Menu for estat

Statistics > Postestimation

Syntax for estat common

`estat common` *varlist* [, *options*]

options	Description
`chi2`	report a one-degree χ^2 test
`level(#)`	set confidence level; default is `level(95)`
display_options	control column formats

varlist may contain factor variables; see **[U] 11.4.3 Factor variables**.

`collect` is allowed; see **[U] 11.1.10 Prefix commands**.

Options for estat common

`chi2` reports a one-degree χ^2 test for the hypothesis that the effects of a covariate on different events are jointly equal to zero. The χ^2 test is reported following the estimation table that contains a point estimate, standard error, z statistic, p-value, and confidence interval for the weighted average effect of each covariate across all events. This option is helpful for comparison of the results with those from the classic multivariate Wald χ^2 test as produced by the command `test`.

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see **[U] 20.8 Specifying the width of confidence intervals**.

display_options: `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, `noci`, and `nopvalues`; see [R] **Estimation options**.

stata.com

Remarks and examples

Remarks are presented under the following headings:

Estimating the average effect using estat common

▷ Example 1: Estimate the weighted average effects

Continuing with example 1 of [ST] **stmgintcox**, we want to test the effects of some covariates on both events. Let's first refit the model. For the purpose of demonstration, we also specify the `favorspeed` option to speed up computation. We also suppress the iteration log with the `nolog` option.

```
. use https://www.stata-press.com/data/r18/aric
(Simulated ARIC data)

. stmgintcox age i.male i.community i.race bmi glucose 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(20)      =  84.36
Log pseudolikelihood = -270.83984                  Prob > chi2        = 0.0000
```

	Haz. ratio	Robust std. err.	z	P>|z|	[95% conf.	interval]
Diabetes						
age	.9552606	.0295589	-1.48	0.139	.8990481	1.014988
male						
Yes	.8084224	.2400335	-0.72	0.474	.451755	1.446684
community						
Jackson	1.597828	.6069935	1.23	0.217	.7588748	3.364265
Minneapolis	1.028054	.342976	0.08	0.934	.5346148	1.976929
Washington	1.407869	.5192024	0.93	0.354	.6833627	2.900504
race						
White	.4289702	.1273669	-2.85	0.004	.2397145	.7676444
bmi	1.116579	.034187	3.60	0.000	1.051545	1.185636
glucose	1.139753	.0303702	4.91	0.000	1.081756	1.200859
sysbp	1.020295	.0122308	1.68	0.094	.9966021	1.04455
diabp	.9928634	.0127512	-0.56	0.577	.9681835	1.018172
Hypertension						
age	.9950085	.0225503	-0.22	0.825	.9517779	1.040203
male						
Yes	.6671401	.1599892	-1.69	0.091	.4169533	1.067448
community						
Jackson	.6085406	.1953944	-1.55	0.122	.3243246	1.141824
Minneapolis	.9040647	.2719638	-0.34	0.737	.5013468	1.630275
Washington	.674088	.2085739	-1.27	0.202	.3675707	1.23621
race						
White	1.261355	.425064	0.69	0.491	.6516152	2.441652
bmi	1.012196	.0195117	0.63	0.529	.9746672	1.05117
glucose	.989899	.0101396	-0.99	0.322	.9702238	1.009973
sysbp	1.075011	.0162901	4.77	0.000	1.043553	1.107418
diabp	1.025533	.0134835	1.92	0.055	.9994433	1.052303

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

Suppose that we want to test the hypotheses that the effects of bmi are zero for all events. We can use test to conduct a classic (two-degrees-of-freedom) Wald χ^2 test:

```
. test bmi
 ( 1)  [Diabetes]bmi = 0
 ( 2)  [Hypertension]bmi = 0
           chi2(  2) =   12.98
         Prob > chi2 =    0.0015
```

We have evidence to reject the null hypotheses that the effects of bmi are zero for all events. We can also test whether the effect of bmi is the same for the two events:

```
. test [1=2]: bmi
 ( 1)  [Diabetes]bmi - [Hypertension]bmi = 0
           chi2(  1) =    8.95
         Prob > chi2 =    0.0028
```

The above result provides evidence that the effects of bmi are different between the two events. Therefore, we now use the estat common command to compute the average effect of bmi on diabetes and hypertension; estat common will report a weighted average of the effect of bmi and a z test of the null hypothesis that the average effect is zero. When event-specific effects are similar, this test is more powerful than the multivariate Wald test reported by test.

```
. estat common bmi
    _avg_bmi: .237*[Diabetes]bmi + .763*[Hypertension]bmi
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
_avg_bmi	.0354094	.0176342	2.01	0.045	.000847	.0699718

The header above the table labels the weighted average effect as _avg_bmi and also reports the optimal weights used for computing the average effect of bmi on diabetes and hypertension. The coefficient table reports the estimated coefficient for _avg_bmi with its standard error, the z statistic, p-value, and confidence interval. The result provides strong evidence that the average effect of bmi is different from zero.

We can add the chi2 option to also report the χ^2 test for the hypothesis that the weighted average effect of BMI on the onset of diabetes and hypertension is zero.

```
. estat common bmi, chi2
    _avg_bmi: .237*[Diabetes]bmi + .763*[Hypertension]bmi
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
_avg_bmi	.0354094	.0176342	2.01	0.045	.000847	.0699718

```
 ( 1)  _avg_bmi = 0
           chi2(  1) =    4.03
         Prob > chi2 =    0.0446
```

The χ^2 test has one degree of freedom, and the result is the same as the result reported by the z-test statistic: $2.01^2 \approx 4.03$.

If we want to compute the average effects of bmi and sysbp across all events, we can type

```
. estat common bmi sysbp
    _avg_bmi: .237*[Diabetes]bmi + .763*[Hypertension]bmi
  _avg_sysbp: .617*[Diabetes]sysbp + .383*[Hypertension]sysbp
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
_avg_bmi	.0354094	.0176342	2.01	0.045	.000847	.0699718
_avg_sysbp	.0401232	.0094602	4.24	0.000	.0215815	.0586648

Looking at the table above, we have strong evidence that the average effects of bmi and sysbp are different from zero. Below, we add chi2 to obtain separate χ^2 tests for each covariate:

```
. estat common bmi sysbp, chi2
    _avg_bmi: .237*[Diabetes]bmi + .763*[Hypertension]bmi
  _avg_sysbp: .617*[Diabetes]sysbp + .383*[Hypertension]sysbp
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
_avg_bmi	.0354094	.0176342	2.01	0.045	.000847	.0699718
_avg_sysbp	.0401232	.0094602	4.24	0.000	.0215815	.0586648

```
 ( 1)  _avg_bmi = 0
 ( 2)  _avg_sysbp = 0
```

	chi2	df	p > chi2
(1)	4.03	1	0.0446
(2)	17.99	1	0.0000

The results from the one-degree-of-freedom tests shown here are identical to the results we would obtain if we issued a separate estat common command for each individual covariate.

◁

Baseline functions

predict after stmgintcox 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 for each event. It can also predict the event-specific 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 zero. If the dataset is in the single-record-per-event format, predict calculates those event-specific statistics for both interval endpoints t_l and t_u specified in the interval() option with stmgintcox. See *Predictions for single-record interval-censored data* and *Predictions for multiple-record interval-censored data* in [ST] **stintcox postestimation** for methods and formulas.

▷ Example 2: Baseline survivor function

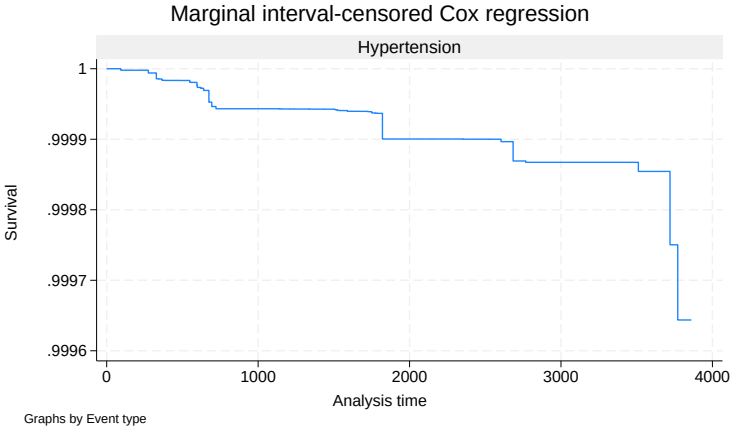
Continuing with example 1, we can estimate, for instance, the baseline survivor function for hypertension. For this dataset, estimates of the baseline survivor function, as well as baseline cumulative hazard and baseline hazard contributions, are intervals. Thus, to compute these statistics, we can either specify two new variable names with `predict` or specify a stub (*stub*`*`) that will automatically create two new variables named *stub1* and *stub2*. The first variable will contain statistics computed using the lower endpoint of the time interval, and the second variable will contain statistics computed using the upper endpoint of the time interval. Below, we also use the `event()` option to specify that we want the baseline survivor function for event 2, meaning hypertension; we can use `event(2)`, `event(#2)`, or `event("Hypertension")` to refer to event hypertension.

```
. predict double bs_l bs_r, basesurv event(2)
(200 missing values generated)
(66 missing values generated)
. list id ltime rtime bs_l bs_r if event==2 & id <=10
```

	id	ltime	rtime	bs_l	bs_r
2.	1	1389	.	.99994272	0
4.	2	2271	.	.99990037	0
6.	3	76	1751	1	.99993729
8.	4	472	1315	.99998341	.99994283
10.	5	0	722	1	.99994327
12.	6	1141	.	.99994309	0
14.	7	346	1541	.99998545	.99994071
16.	8	1317	.	.99994283	0
18.	9	530	1751	.99998339	.99993729
20.	10	1552	2767	.99994071	.99986726

The event-specific baseline survivor functions for hypertension, `bs_l` and `bs_r`, evaluated at the lower and upper endpoints of the time interval, `ltime` and `rtime`, are listed above for subjects 1 through 10. If we look at the values of `ltime` and `rtime`, we will see that these subjects represent different censoring types: subjects 1, 2, 6, and 8 are right-censored, subject 5 is left-censored, and the remaining subjects are interval-censored. To graph the baseline survival curve for hypertension, we can use the `stcurve` command with the `survival` option. We use the `events()` option to specify the event hypertension; we can identify the event using its numerical value (`2`), its position (`#2`), or the label `"Hypertension"`, which is the label corresponding to event 2 as defined by the value label for variable `event`. Additionally, we need to set all covariates to zero using the `at()` option. Alternatively, we can use the following flexible specification to obtain the baseline survival function:

```
. stcurve, surv at((zero)_c (base)_f) events(2)
note: function evaluated at specified values of selected covariates and
      overall means of other covariates (if any) for specified event.
```



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**. Farrington (2000) proposed extensions of those residuals for univariate interval-censored data; see *Residuals and diagnostic measures* in [ST] **stintcox postestimation** for details.

Here we present event-specific martingale-like residuals and Cox–Snell-like residuals for multiple-event interval-censored event-time data. 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. All the residuals are event specific and can be obtained by `predict`.

▷ Example 3: 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 data, 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 for each event and plots the resulting cumulative hazard rate against the residuals themselves. If the model fits the data, those plots are expected to approximate a straight line with slope 1.

Continuing with example 2, let's produce the goodness-of-fit plots for both events by typing

```
. estat gofplot
```

By default, `estat gofplot` displays all event-specific goodness-of-fit plots as subgraphs within a single graph. The goodness-of-fit plot on the left shows that the jagged line remains close to the 45° reference line, suggesting that the marginal Cox proportional hazards model fits the data well for diabetes. The plot on the right indicates that the marginal Cox proportional hazards model fits the data mostly well for hypertension, except for the tail, which deviates from the 45° line because of an outlier.

◁

Stored results

`estat common` stores the following in `r()`:

Matrices

`r(chi2table)`	matrix containing results of χ^2 tests
`r(weights)`	matrix of the weights for coefficients
`r(b)`	vector of weighted average of coefficients
`r(V)`	estimated variance–covariance matrix of the weighted average of coefficients
`r(table)`	matrix containing the estimates with their standard errors, test statistics, p-values, and confidence intervals

Methods and formulas

With `estat common`, we can estimate the average effect of a covariate on all events in our model. Suppose we are modeling K events; we denote the K regression parameters by η_k $(k = 1, \ldots, K)$. Let $\hat{\eta}_k$ be the estimator of η_k, and let $\widehat{\Psi} = \left\{\widehat{\psi}_{kl}; k, l = 1, \ldots, K\right\}$ be the estimated covariance matrix of $(\hat{\eta}_1, \ldots, \hat{\eta}_K)$, which is a subset of the covariance matrix for $\hat{\beta}_k$ $(k = 1, \ldots, K)$, defined in *Methods and formulas* of [ST] **stmgintcox**. Then the quadratic form $Q = (\hat{\eta}_1, \ldots, \hat{\eta}_K)\widehat{\Psi}^{-1}(\hat{\eta}_1, \ldots, \hat{\eta}_K)^T$ (the classic multivariate Wald χ^2 test) can be used to test the null hypothesis that $\eta_1 = \eta_2 = \cdots = \eta_K = 0$.

Suppose that $\eta_1 = \eta_2 = \cdots = \eta_K = \eta$. Then it is natural to estimate η by a linear combination of the $\hat{\eta}_k$'s; that is, $\hat{\eta} = \sum_{k=1}^{K} c_k \hat{\eta}_k$, with $\sum_{k=1}^{K} c_k = 1$. The following choice of weights minimizes the variance of $\hat{\eta}$,

$$c = [c_1, \ldots, c_K]^T = (e^T \widehat{\Psi}^{-1} e)^{-1} \widehat{\Psi}^{-1} e$$

where $e = [1, \ldots, 1]^T$.

For large samples, $\hat{\eta}$ is approximately normal with mean η and variance $\sum_{k=1}^{K} \sum_{l=1}^{K} c_k c_l \widehat{\psi}_{kl}$, where $\widehat{\psi}_{kl}$ is the (k, l)th element of $\widehat{\Psi}$. This distribution can be used to contruct a standard-normal test statistic to form a one-degree χ^2 test, which tends to be more powerful than Q if the η_k's are similar. It can also be used to construct a confidence interval for η. Although the η_k's may not be equal, $\hat{\eta}$ estimates an average effect.

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.

Also see

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