Title

stcox postestimation — Postestimation tools for stcox

Description

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

command	Description
`estat concordance`	compute the concordance probability
`stcurve`	plot the survivor, hazard, and cumulative hazard functions

`estat concordance` is not appropriate after estimation with `svy`.

For information on `estat concordance`, see below. For information on `stcurve`, see [ST] **stcurve**.

The following standard postestimation commands are also available:

command	Description
`estat`	AIC, BIC, VCE, and estimation sample summary
`estat` (svy)	postestimation statistics for survey data
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`linktest`	link test for model specification
`lrtest`[1]	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

[1] `lrtest` is not appropriate with `svy` estimation results.

See the corresponding entries in the *Base Reference Manual* for details, but see [SVY] **estat** for details about `estat` (svy).

Special-interest postestimation commands

`estat concordance` calculates the concordance probability, which is defined as the probability that predictions and outcomes are concordant. `estat concordance` provides two measures of the concordance probability: Harrell's C and Gönen and Heller's K concordance coefficients. `estat concordance` also reports the Somers' D rank correlation, which is obtained by calculating $2C - 1$ or $2K - 1$.

Syntax for predict

```
predict [type] newvar [if] [in] [, sv_statistic nooffset partial]

predict [type] {stub* | newvarlist} [if] [in], mv_statistic [ partial]
```

sv_statistic	description
Main	
`hr`	predicted hazard ratio, also known as the relative hazard; the default
`xb`	linear prediction $\mathbf{x}_j\boldsymbol{\beta}$
`stdp`	standard error of the linear prediction; SE($\mathbf{x}_j\boldsymbol{\beta}$)
* `basesurv`	baseline survivor function
* `basechazard`	baseline cumulative hazard function
* `basehc`	baseline hazard contributions
* `mgale`	martingale residuals
* `csnell`	Cox–Snell residuals
* `deviance`	deviance residuals
* `ldisplace`	likelihood displacement values
* `lmax`	LMAX measures of influence
* `effects`	log frailties

mv_statistic	description
Main	
* `scores`	efficient score residuals
* `esr`	synonym for `scores`
* `dfbeta`	DFBETA measures of influence
* `schoenfeld`	Schoenfeld residuals
* `scaledsch`	scaled Schoenfeld 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, even when `e(sample)` is not specified. `nooffset` is allowed only with unstarred statistics.

`mgale`, `csnell`, `deviance`, `ldisplace`, `lmax`, `dfbeta`, `schoenfeld`, and `scaledsch` are not allowed with `svy` estimation results.

Menu

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`hr`, the default, calculates the relative hazard (hazard ratio), that is, the exponentiated linear prediction, $\exp(\mathbf{x}_j\widehat{\boldsymbol{\beta}})$.

`xb` calculates the linear prediction from the fitted model. That is, you fit the model by estimating a set of parameters, β_0, β_1, β_2, ..., β_k, and the linear prediction is $\widehat{\beta}_1x_{1j} + \widehat{\beta}_2x_{2j} + \cdots + \widehat{\beta}_kx_{kj}$, often written in matrix notation as $\mathbf{x}_j\widehat{\boldsymbol{\beta}}$.

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

`stdp` calculates the standard error of the prediction, that is, the standard error of $\mathbf{x}_j\widehat{\boldsymbol{\beta}}$.

`basesurv` calculates the baseline survivor function. In the null model, this is equivalent to the Kaplan–Meier product-limit estimate. If `stcox`'s `strata()` option was specified, baseline survivor functions for each stratum are provided.

`basechazard` calculates the cumulative baseline hazard. If `stcox`'s `strata()` option was specified, cumulative baseline hazards for each stratum are provided.

`basehc` calculates the baseline hazard contributions. These are used to construct the product-limit type estimator for the baseline survivor function generated by `basesurv`. If `stcox`'s `strata()` option was specified, baseline hazard contributions for each stratum are provided.

`mgale` calculates the martingale residuals. For multiple-record-per-subject data, by default only one value per subject is calculated, and it is placed on the last record for the subject.

Adding the `partial` option will produce partial martingale residuals, one for each record within subject; see `partial` below. Partial martingale residuals are the additive contributions to a subject's overall martingale residual. In single-record-per-subject data, the partial martingale residuals are the martingale residuals.

`csnell` calculates the Cox–Snell generalized residuals. For multiple-record data, by default only one value per subject is calculated and, it is placed on the last record for the subject.

Adding the `partial` option will produce partial Cox–Snell residuals, one for each record within subject; see `partial` below. Partial Cox–Snell residuals are the additive contributions to a subject's overall Cox–Snell residual. In single-record data, the partial Cox–Snell residuals are the Cox–Snell residuals.

`deviance` calculates the deviance residuals. Deviance residuals are martingale residuals that have been transformed to be more symmetric about zero. For multiple-record data, by default only one value per subject is calculated, and it is placed on the last record for the subject.

Adding the `partial` option will produce partial deviance residuals, one for each record within subject; see `partial` below. Partial deviance residuals are transformed partial martingale residuals. In single-record data, the partial deviance residuals are the deviance residuals.

`ldisplace` calculates the *likelihood displacement values*. A likelihood displacement value is an influence measure of the effect of deleting a subject on the overall coefficient vector. For multiple-record data, by default only one value per subject is calculated, and it is placed on the last record for the subject.

Adding the `partial` option will produce partial likelihood displacement values, one for each record within subject; see `partial` below. Partial displacement values are interpreted as effects due to deletion of individual records rather than deletion of individual subjects. In single-record data, the partial likelihood displacement values are the likelihood displacement values.

`lmax` calculates the LMAX measures of influence. LMAX values are related to likelihood displacement values because they also measure the effect of deleting a subject on the overall coefficient vector. For multiple-record data, by default only one LMAX value per subject is calculated, and it is placed on the last record for the subject.

Adding the `partial` option will produce partial LMAX values, one for each record within subject; see `partial` below. Partial LMAX values are interpreted as effects due to deletion of individual records rather than deletion of individual subjects. In single-record data, the partial LMAX values are the LMAX values.

`effects` is for use after `stcox, shared()` and provides estimates of the log frailty for each group. The log frailties are random group-specific offsets to the linear predictor that measure the group effect on the log relative-hazard.

`scores` calculates the efficient score residuals for each regressor in the model. For multiple-record data, by default only one score per subject is calculated, and it is placed on the last record for the subject.

Adding the `partial` option will produce partial efficient score residuals, one for each record within subject; see `partial` below. Partial efficient score residuals are the additive contributions to a subject's overall efficient score residual. In single-record data, the partial efficient score residuals are the efficient score residuals.

One efficient score residual variable is created for each regressor in the model; the first new variable corresponds to the first regressor, the second to the second, and so on.

`esr` is a synonym for `scores`.

`dfbeta` calculates the DFBETA measures of influence for each regressor in the model. The DFBETA value for a subject estimates the change in the regressor's coefficient due to deletion of that subject. For multiple-record data, by default only one value per subject is calculated, and it is placed on the last record for the subject.

Adding the `partial` option will produce partial DFBETAs, one for each record within subject; see `partial` below. Partial DFBETAs are interpreted as effects due to deletion of individual records rather than deletion of individual subjects. In single-record data, the partial DFBETAs are the DFBETAs.

One DFBETA variable is created for each regressor in the model; the first new variable corresponds to the first regressor, the second to the second, and so on.

`schoenfeld` calculates the Schoenfeld residuals. This option may not be used after `stcox` with the `exactm` or `exactp` option. Schoenfeld residuals are calculated and reported only at failure times.

One Schoenfeld residual variable is created for each regressor in the model; the first new variable corresponds to the first regressor, the second to the second, and so on.

`scaledsch` calculates the scaled Schoenfeld residuals. This option may not be used after `stcox` with the `exactm` or `exactp` option. Scaled Schoenfeld residuals are calculated and reported only at failure times.

One scaled Schoenfeld residual variable is created for each regressor in the model; the first new variable corresponds to the first regressor, the second to the second, and so on.

Note: The easiest way to use the preceding four options is, for example,

```
. predict double stub*, scores
```

where *stub* is a short name of your choosing. Stata then creates variables *stub*1, *stub*2, etc. You may also specify each variable explicitly, in which case there must be as many (and no more) variables specified as there are regressors in the model.

`nooffset` is allowed only with `hr`, `xb`, and `stdp`, and is relevant only if you specified `offset(`*varname*`)` for `stcox`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_j\widehat{\boldsymbol{\beta}}$ rather than $\mathbf{x}_j\widehat{\boldsymbol{\beta}} + \text{offset}_j$.

`partial` is relevant only for multiple-record data and is valid with `mgale`, `csnell`, `deviance`, `ldisplace`, `lmax`, `scores`, `esr`, and `dfbeta`. Specifying `partial` will produce "partial" versions of these statistics, where one value is calculated for each record instead of one for each subject. The subjects are determined by the `id()` option to `stset`.

Specify partial if you wish to perform diagnostics on individual records rather than on individual subjects. For example, a partial DFBETA would be interpreted as the effect on a coefficient due to deletion of one record, rather than the effect due to deletion of all records for a given subject.

Syntax for estat concordance

```
estat concordance [if] [in] [, concordance_options]
```

concordance_options	description
Main	
harrell	compute Harrell's C coefficient; the default
gheller	compute Gönen and Heller's concordance coefficient
se	compute asymptotic standard error of Gönen and Heller's coefficient
all	compute statistic for all observations in the data
noshow	do not show st setting information

Menu

Statistics > Postestimation > Reports and statistics

Options for estat concordance

Main

harrell, the default, calculates Harrell's C coefficient, which is defined as the proportion of all usable subject pairs in which the predictions and outcomes are concordant.

gheller calculates Gönen and Heller's K concordance coefficient instead of Harrell's C coefficient. The harrell and gheller options may be specified together to obtain both concordance measures.

se calculates the smoothed version of Gönen and Heller's K concordance coefficient and its asymptotic standard error. The se option requires the gheller option.

all requests that the statistic be computed for all observations in the data. By default, estat concordance computes over the estimation subsample.

noshow prevents estat concordance from displaying the identities of the key st variables above its output.

Remarks

Remarks are presented under the following headings:

Baseline functions
Making baseline reasonable
Residuals and diagnostic measures
Multiple records per subject
Predictions after stcox with the tvc() option
Predictions after stcox with the shared() option
estat concordance

Baseline functions

`predict` after `stcox` provides estimates of the baseline survivor and baseline cumulative hazard function, among other things. Here the term *baseline* means that these are the functions when all covariates are set to zero, that is, they reflect (perhaps hypothetical) individuals who have zero-valued measurements. When you specify `predict`'s `basechazard` option, you obtain the baseline cumulative hazard. When you specify `basesurv`, you obtain the baseline survivor function. Additionally, when you specify `predict`'s `basehc` option, you obtain estimates of the baseline hazard contribution at each failure time, which are factors used to develop the product-limit estimator for the survivor function generated by `basesurv`.

Although in theory $S_0(t) = \exp\{-H_0(t)\}$, where $S_0(t)$ is the baseline survivor function and $H_0(t)$ is the baseline cumulative hazard, the estimates produced by `basechazard` and `basesurv` do not exactly correspond in this manner, although they closely do. The reason is that `predict` after `stcox` uses different estimation schemes for each; the exact formulas are given in *Methods and formulas*.

When the Cox model is fit with the `strata()` option, you obtain estimates of the baseline functions for each stratum.

▷ Example 1: Baseline survivor function

Baseline functions refer to the values of the functions when all covariates are set to 0. Let's graph the survival curve for the Stanford heart transplant model that we fit in example 3 of [ST] **stcox**, and to make the baseline curve reasonable, let's do that at `age` $= 40$ and `year` $= 70$.

Thus we will begin by creating variables that, when 0, correspond to the baseline values we desire, and then we will fit our model with these variables instead. We then predict the baseline survivor function and graph it:

```
. use http://www.stata-press.com/data/r11/stan3
(Heart transplant data)
. generate age40 = age - 40
. generate year70 = year - 70
. stcox age40 posttran surg year70, nolog
         failure _d:  died
   analysis time _t:  t1
                 id:  id
Cox regression -- Breslow method for ties
No. of subjects =          103                     Number of obs   =       172
No. of failures =           75
Time at risk    =      31938.1
                                                   LR chi2(4)      =     17.56
Log likelihood  =   -289.53378                     Prob > chi2     =    0.0015
------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       age40 |   1.030224   .0143201     2.14   0.032     1.002536    1.058677
    posttran |   .9787243   .3032597    -0.07   0.945     .5332291    1.796416
     surgery |   .3738278    .163204    -2.25   0.024     .1588759       .8796
     year70 |   .8873107    .059808    -1.77   0.076     .7775022    1.012628
------------------------------------------------------------------------------
. predict s, basesurv
```

```
. summarize s
```

Variable	Obs	Mean	Std. Dev.	Min	Max
s	172	.6291871	.2530009	.130666	.9908968

Our recentering of `age` and `year` did not affect the estimation, a fact you can verify by refitting the model with the original `age` and `year` variables.

To see how the values of the baseline survivor function are stored, we first sort according to analysis time and then list some observations.

```
. sort _t id
. list id _t0 _t _d s in 1/20
```

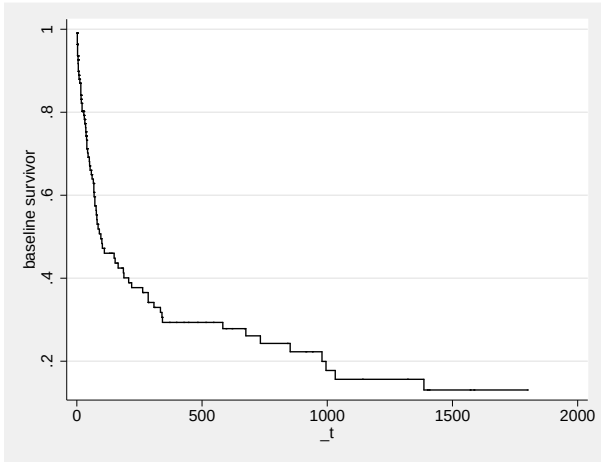
	id	_t0	_t	_d	s
1.	3	0	1	0	.9908968
2.	15	0	1	1	.9908968
3.	20	0	1	0	.9908968
4.	45	0	1	0	.9908968
5.	39	0	2	0	.9633915
6.	43	0	2	1	.9633915
7.	46	0	2	0	.9633915
8.	61	0	2	1	.9633915
9.	75	0	2	1	.9633915
10.	95	0	2	0	.9633915
11.	6	0	3	1	.9356873
12.	23	0	3	0	.9356873
13.	42	0	3	1	.9356873
14.	54	0	3	1	.9356873
15.	60	0	3	0	.9356873
16.	68	0	3	0	.9356873
17.	72	0	4	0	.9356873
18.	94	0	4	0	.9356873
19.	38	0	5	0	.9264087
20.	70	0	5	0	.9264087

At time $\_t = 2$, the baseline survivor function is 0.9634, or more precisely, $S_0(2 + \Delta t) = 0.9634$. What we mean by $S_0(t + \Delta t)$ is the probability of surviving just beyond t. This is done to clarify that the probability includes escaping failure at precisely time t.

The above also indicates that our estimate of $S_0(t)$ is a step function, and that the steps occur only at times when failure is observed—our estimated $S_0(t)$ does not change from $\_t = 3$ to $\_t = 4$ because no failure occurred at time 4. This behavior is analogous to that of the Kaplan–Meier estimate of the survivor function; see [ST] **sts**.

Here is a graph of the baseline survival curve:

```
. line s _t, sort c(J)
```



This graph was easy enough to produce because we wanted the survivor function at baseline. To graph survivor functions after `stcox` with covariates set to any value (baseline or otherwise), use `stcurve`; see [ST] **stcurve**. ◁

The similarity to Kaplan–Meier is not limited to the fact that both are step functions that change only when failure occurs. They are also calculated in much the same way, with predicting `basesurv` after `stcox` having the added benefit that the result is automatically adjusted for all the covariates in your Cox model. When you have no covariates, both methods are equivalent. If you continue from the previous example, you will find that

```
. sts generate s1 = s
```

and

```
. stcox, estimate
. predict double s2, basesurv
```

produce the identical variables `s1` and `s2`, both containing estimates of the overall survivor function, unadjusted for covariates. We used type `double` for `s2` to precisely match `sts generate`, which gives results in double precision.

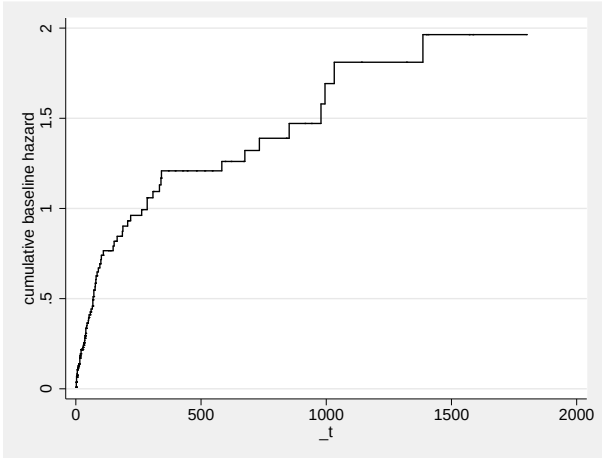
If we had fit a stratified model by using the `strata()` option, the recorded survivor-function estimate on each observation would be for the stratum of that observation. That is, what you get is one variable that holds not an overall survivor curve, but instead a set of stratum-specific curves.

▷ Example 2: Baseline cumulative hazard

Obtaining estimates of the baseline cumulative hazard, $H_0(t)$, is just as easy as obtaining the baseline survivor function. Using the same data as previously,

```
. use http://www.stata-press.com/data/r11/stan3, clear
(Heart transplant data)
. generate age40 = age - 40
. generate year70 = year - 70
```

```
. stcox age40 posttran surg year70
  (output omitted)
. predict ch, basechazard
. line ch _t, sort c(J)
```



The estimated baseline cumulative hazard is also a step function with the steps occurring at the observed times of failure. When there are no covariates in your Cox model, what you obtain is equivalent to the Nelson–Aalen estimate of the cumulative hazard (see [ST] **sts**), but using `predict, basechazard` after `stcox` allows you to also adjust for covariates.

To obtain cumulative hazard curves at values other than baseline, you could either recenter your covariates—as we did previously with `age` and `year`—so that the values in which you are interested become baseline, or simply use `stcurve`; see [ST] **stcurve**.

◁

▷ Example 3: Baseline hazard contributions

Mathematically, a baseline hazard contribution, $h_i = (1 - \alpha_i)$ (see Kalbfleisch and Prentice 2002, 115), is defined at every analytic time t_i at which a failure occurs and is undefined at other times. Stata stores h_i in observations where a failure occurred and stores missing values in the other observations.

```
. use http://www.stata-press.com/data/r11/stan3, clear
(Heart transplant data)
. generate age40 = age - 40
. generate year70 = year - 70
. stcox age40 posttran surg year70
  (output omitted)
. predict double h, basehc
(97 missing values generated)
```

```
. list id _t0 _t _d h in 1/10
```

	id	_t0	_t	_d	h
1.	1	0	50	1	.01503465
2.	2	0	6	1	.02035303
3.	3	0	1	0	.
4.	3	1	16	1	.03339642
5.	4	0	36	0	.
6.	4	36	39	1	.01365406
7.	5	0	18	1	.01167142
8.	6	0	3	1	.02875689
9.	7	0	51	0	.
10.	7	51	675	1	.06215003

At time `_t` $= 50$, the hazard contribution h_1 is 0.0150. At time `_t` $= 6$, the hazard contribution h_2 is 0.0204. In observation 3, no hazard contribution is stored. Observation 3 contains a missing value because observation 3 did not fail at time 1. We also see that values of the hazard contributions are stored only in observations that are marked as failing.

Hazard contributions by themselves have no substantive interpretation, and in particular they should *not* be interpreted as estimating the hazard function at time t. Hazard contributions are simply mass points that are used as components to calculate the survivor function; see *Methods and formulas*. You can also use hazard contributions to estimate the hazard, but because they are only mass points, they need to be smoothed first. This smoothing is done automatically with `stcurve`; see [ST] **stcurve**. In summary, hazard contributions in their raw form serve no purpose other than to help replicate calculations done by Stata, and we demonstrate this below simply for illustrative purposes.

When we created the new variable `h` for holding the hazard contributions, we used type `double` because we plan on using `h` in some further calculations below and we wish to be as precise as possible.

In contrast with the baseline hazard contributions, the baseline survivor function, $S_0(t)$, is defined at all values of t: its estimate changes its value when failures occur, and at times when no failures occur, the estimated $S_0(t)$ is equal to its value at the time of the last failure.

Continuing with our example, we now predict the baseline survivor function:

```
. predict double s, basesurv
. list id _t0 _t _d h s in 1/10
```

	id	_t0	_t	_d	h	s
1.	1	0	50	1	.01503465	.68100303
2.	2	0	6	1	.02035303	.89846438
3.	3	0	1	0	.	.99089681
4.	3	1	16	1	.03339642	.84087361
5.	4	0	36	0	.	.7527663
6.	4	36	39	1	.01365406	.73259264
7.	5	0	18	1	.01167142	.82144038
8.	6	0	3	1	.02875689	.93568733
9.	7	0	51	0	.	.6705895
10.	7	51	675	1	.06215003	.26115633

In the above, we sorted by `id`, but it is easier to see how `h` and `s` are related if we sort by `_t` and put the failures on top:

```
. gsort +_t -_d
. list id _t0 _t _d h s in 1/18
```

	id	_t0	_t	_d	h	s
1.	15	0	1	1	.00910319	.99089681
2.	20	0	1	0	.	.99089681
3.	45	0	1	0	.	.99089681
4.	3	0	1	0	.	.99089681
5.	75	0	2	1	.02775802	.96339147
6.	61	0	2	1	.02775802	.96339147
7.	43	0	2	1	.02775802	.96339147
8.	39	0	2	0	.	.96339147
9.	95	0	2	0	.	.96339147
10.	46	0	2	0	.	.96339147
11.	6	0	3	1	.02875689	.93568733
12.	42	0	3	1	.02875689	.93568733
13.	54	0	3	1	.02875689	.93568733
14.	60	0	3	0	.	.93568733
15.	23	0	3	0	.	.93568733
16.	68	0	3	0	.	.93568733
17.	72	0	4	0	.	.93568733
18.	94	0	4	0	.	.93568733

The baseline hazard contribution is stored on every failure record—if multiple failures occur at a given time, the value of the hazard contribution is repeated—and the baseline survivor is stored on every record. (More correctly, baseline values are stored on records that meet the criterion and that were used in estimation. If some observations are explicitly or implicitly excluded from the estimation, their baseline values will be set to missing, no matter what.)

With this listing, we can better understand how the hazard contributions are used to calculate the survivor function. Because the patient with `id` $= 15$ died at time $t_1 = 1$, the hazard contribution for that patient is $h_{15} = 0.00910319$. Because that was the only death at $t_1 = 1$, the estimated survivor function at this time is $S_0(1) = 1 - h_{15} = 1 - 0.00910319 = 0.99089681$. The next death occurs at time $t_1 = 2$, and the hazard contribution at this time for patient 43 (or patient 61 or patient 75, it does not matter) is $h_{43} = 0.02775802$. Multiplying the previous survivor function value by $1 - h_{43}$ gives the new survivor function at $t_1 = 2$ as $S_0(2) = 0.96339147$. The other survivor function values are then calculated in succession, using this method at each failure time. At times when no failures occur, the survivor function remains unchanged.

◁

❑ Technical note

Consider manually obtaining the estimate of $S_0(t)$ from the h_i:

```
. sort _t _d
. by _t: keep if _d & _n==_N
. generate double s2 = 1-h
. replace s2 = s2[_n-1]*s2 if _n>1
```

`s2` will be equivalent to `s` as produced above. If you had obtained stratified estimates, the code would be

```
. sort group _t _d
. by group _t: keep if _d & _n==_N
. generate double s2 = 1-h
. by group: replace s2 = s2[_n-1]*s2 if _n>1
```

❑

Making baseline reasonable

When predicting with `basesurv` or `basechazard`, for numerical accuracy reasons, the baseline functions must correspond to something reasonable in your data. Remember, the baseline functions correspond to all covariates equal to 0 in your Cox model.

Consider, for instance, a Cox model that includes the variable calendar year among the covariates. Say that `year` varies between 1980 and 1996. The baseline functions would correspond to year 0, almost 2,000 years in the past. Say that the estimated coefficient on `year` is -0.2, meaning that the hazard ratio for one year to the next is a reasonable 0.82.

Think carefully about the contribution to the predicted log cumulative hazard: it would be approximately $-0.2 \times 2{,}000 = -400$. Now $e^{-400} \approx 10^{-173}$, which on a digital computer is so close to 0 that there is simply no hope that $H_0(t)e^{-400}$ will produce an accurate estimate of $H(t)$.

Even with less extreme numbers, problems arise, even in the calculation of the baseline survivor function. Baseline hazard contributions near 1 produce baseline survivor functions with steps differing by many orders of magnitude because the calculation of the survivor function is cumulative. Producing a meaningful graph of such a survivor function is hopeless, and adjusting the survivor function to other values of the covariates is too much work.

For these reasons, covariate values of 0 must be meaningful if you are going to specify the `basechazard` or `basesurv` option. As the baseline values move to absurdity, the first problem you will encounter is a baseline survivor function that is too hard to interpret, even though the baseline hazard contributions are estimated accurately. Further out, the procedure Stata uses to estimate the baseline hazard contributions will break down—it will produce results that are exactly 1. Hazard contributions that are exactly 1 produce survivor functions that are uniformly 0, and they will remain 0 even after adjusting for covariates.

This, in fact, occurs with the Stanford heart transplant data:

```
. use http://www.stata-press.com/data/r11/stan3, clear
(Heart transplant data)
. stcox age posttran surg year
  (output omitted)
. predict ch, basechazard
. predict s, basesurv
. summarize ch s
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          ch |       172    745.1134    682.8671   11.88239   2573.637
           s |       172    1.45e-07    9.43e-07          0   6.24e-06
```

The hint that there are problems is that the values of `ch` are huge and the values of `s` are close to 0. In this dataset, `age` (which ranges from 8 to 64 with a mean value of 45) and `year` (which ranges from 67 to 74) are the problems. The baseline functions correspond to a newborn at the turn of the century on the waiting list for a heart transplant!

To obtain accurate estimates of the baseline functions, type

```
. drop ch s
. generate age40 = age - 40
. generate year70 = year - 70
. stcox age40 posttran surg year70
  (output omitted)
. predict ch, basechazard
. predict s, basesurv
. summarize ch s
```

Variable	Obs	Mean	Std. Dev.	Min	Max
ch	172	.5685743	.521076	.0090671	1.963868
s	172	.6291871	.2530009	.130666	.9908968

Adjusting the variables does not affect the coefficient (and, hence, hazard-ratio) estimates, but it changes the values at which the baseline functions are estimated to be within the range of the data.

❑ Technical note

Above we demonstrated what can happen to predicted baseline functions when baseline values represent a departure from what was observed in the data. In the above example, the Cox model fit was fine and only the baseline functions lacked accuracy. As baseline values move even further toward absurdity, the risk-set accumulations required to fit the Cox model will also break down. If you are having difficulty getting `stcox` to converge or you obtain missing coefficients, one possible solution is to recenter your covariates just as we did above.

❑

Residuals and diagnostic measures

Stata can calculate Cox–Snell residuals, martingale residuals, deviance residuals, efficient score residuals (esr), Schoenfeld residuals, scaled Schoenfeld residuals, likelihood displacement values, LMAX values, and DFBETA influence measures.

Although the uses of residuals vary and depend on the data and user preferences, traditional and suggested uses are the following: Cox–Snell residuals are useful in assessing overall model fit. Martingale residuals are useful in determining the functional form of covariates to be included in the model and are occasionally useful in identifying outliers. Deviance residuals are useful in examining model accuracy and identifying outliers. Schoenfeld and scaled Schoenfeld residuals are useful for checking and testing the proportional-hazards assumption. Likelihood displacement values and LMAX values are useful in identifying influential subjects. DFBETAs also measure influence, but they do so on a coefficient-by-coefficient basis. Likelihood displacement values, LMAX values, and DFBETAs are all based on efficient score residuals.

▷ Example 4: Cox–Snell residuals

Let's first examine the use of Cox–Snell residuals. Using the cancer data introduced in example 2 in [ST] **stcox**, we first perform a Cox regression and then `predict` the Cox–Snell residuals.

```
. use http://www.stata-press.com/data/r11/drugtr, clear
(Patient Survival in Drug Trial)
. stset studytime, failure(died)
  (output omitted)
```

```
. stcox age drug, nolog
        failure _d:  died
  analysis time _t:  studytime
Cox regression -- Breslow method for ties
No. of subjects =           48                 Number of obs   =        48
No. of failures =           31
Time at risk    =          744
                                               LR chi2(2)      =     33.18
Log likelihood  =   -83.323546                 Prob > chi2     =    0.0000
```

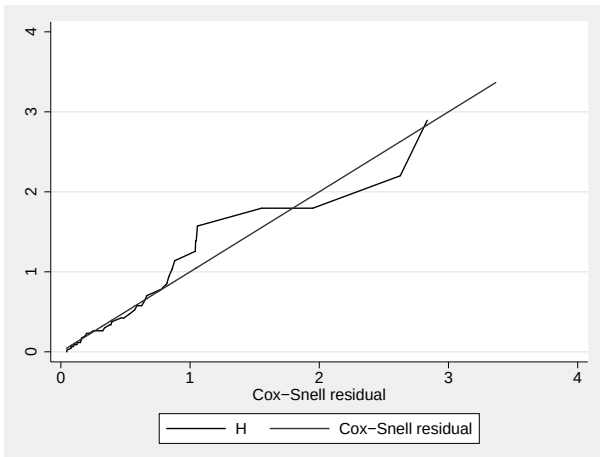
_t	Haz. Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
age	1.120325	.0417711	3.05	0.002	1.041375	1.20526
drug	.1048772	.0477017	-4.96	0.000	.0430057	.2557622

```
. predict cs, csnell
```

The `csnell` option tells `predict` to output the Cox–Snell residuals to a new variable, `cs`. If the Cox regression model fits the data, these residuals should have a standard censored exponential distribution with hazard ratio 1. We can verify the model's fit by calculating—based, for example, on the Kaplan–Meier estimated survivor function or the Nelson–Aalen estimator—an empirical estimate of the cumulative hazard function, using the Cox–Snell residuals as the time variable and the data's original censoring variable. If the model fits the data, the plot of the cumulative hazard versus `cs` should approximate a straight line with slope 1.

To do this, we first re-`stset` the data, specifying `cs` as our new failure-time variable and `died` as the failure/censoring indicator. We then use the `sts generate` command to generate the `km` variable containing the Kaplan–Meier survivor estimates. Finally, we generate the cumulative hazard, `H`, by using the relationship $\texttt{H} = -\ln(\texttt{km})$ and plot it against `cs`.

```
. stset cs, failure(died)
  (output omitted)
. sts generate km = s
. generate H = -ln(km)
(1 missing value generated)
. line H cs cs, sort ytitle("") clstyle(. refline)
```



We specified `cs` twice in the graph command above so that a reference 45° line is plotted. Comparing the jagged line with the reference line, we observe that the Cox model does not fit these data too badly.

◁

❑ Technical note

The statement that "if the Cox regression model fits the data, the Cox–Snell residuals have a standard censored exponential distribution with hazard ratio 1" holds only if the true parameters, $\boldsymbol{\beta}$, and the true cumulative baseline hazard function, $H_0(t)$, are used in calculating the residuals. Because we use estimates $\widehat{\boldsymbol{\beta}}$ and $\widehat{H}_0(t)$, deviations from the 45° line in the above plots could be due in part to uncertainty about these estimates. This is particularly important for small sample sizes and in the right-hand tail of the distribution, where the baseline hazard is more variable because of the reduced effective sample caused by prior failures and censoring.

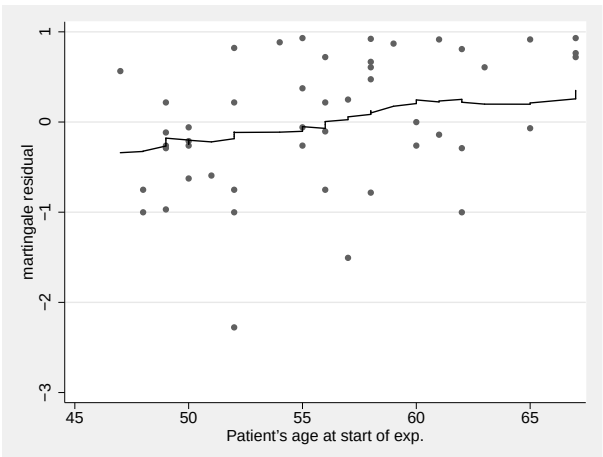
❑

▷ Example 5: Martingale residuals

Let's now examine the martingale residuals. Martingale residuals are useful in assessing the functional form of a covariate to be entered into a Cox model. Sometimes the covariate may need transforming so that the transformed variable will satisfy the assumptions of the proportional hazards model. To find the appropriate functional form of a variable, we fit a Cox model excluding the variable and then plot a `lowess` smooth of the martingale residuals against some transformation of the variable in question. If the transformation is appropriate, then the smooth should be approximately linear.

We apply this procedure to our cancer data to find an appropriate transformation of `age` (or to verify that `age` need not be transformed).

```
. use http://www.stata-press.com/data/r11/drugtr, clear
(Patient Survival in Drug Trial)
. stset studytime, failure(died)
  (output omitted)
. stcox drug
  (output omitted)
. predict mg, mgale
. lowess mg age, 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. A lowess smoother or other smoother could also be used; see [R] **lowess**. The smooth appears nearly linear, supporting the inclusion of the untransformed version of `age` in our Cox model. Had the smooth not been linear, we would have tried smoothing the martingale residuals against various transformations of `age` until we found one that produced a near-linear smooth.

◁

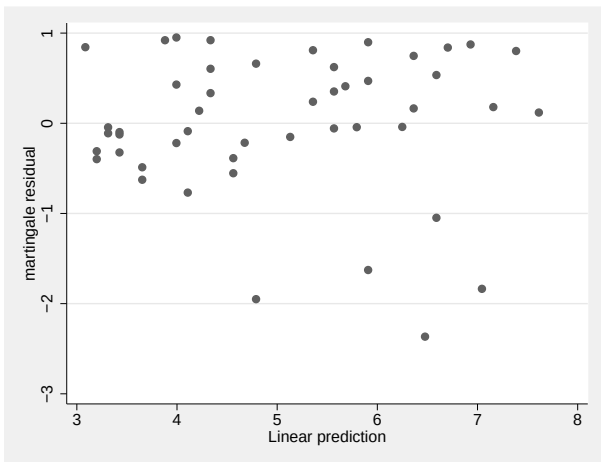
Martingale residuals can also be interpreted as the difference over time of the observed number of failures minus the difference predicted by the model. Thus a plot of the martingale residuals versus the linear predictor may be used to detect outliers.

Plots of martingale residuals are sometimes difficult to interpret, however, because these residuals are skewed, taking values in $(-\infty, 1)$. For this reason, deviance residuals are preferred for examining model accuracy and identifying outliers.

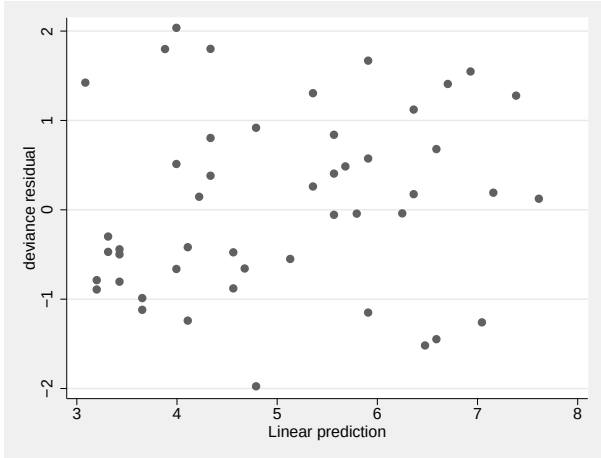
▷ Example 6: Deviance residuals

Deviance residuals are a rescaling of the martingale residuals so that they are symmetric about 0 and thus are more like residuals obtained from linear regression. Plots of these residuals against the linear predictor, survival time, rank order of survival, or observation number can be useful in identifying aberrant observations and assessing model fit. We continue from the previous example, but we need to first refit the Cox model with `age` included:

```
. drop mg
. stcox drug age
  (output omitted)
. predict mg, mgale
. predict xb, xb
. scatter mg xb
```



```
. predict dev, deviance
. scatter dev xb
```



We first plotted the martingale residuals versus the linear predictor and then plotted the deviance residuals versus the linear predictor. Given their symmetry about 0, deviance residuals are easier to interpret, although both graphs yield the same information. With uncensored data, deviance residuals should resemble white noise if the fit is adequate. Censored observations would be represented as clumps of deviance residuals near 0 (Klein and Moeschberger 2003, 381). Given what we see above, there do not appear to be any outliers.

◁

In evaluating the adequacy of the fitted model, we must determine if any one subject has a disproportionate influence on the estimated parameters. This is known as influence or leverage analysis. The preferred method of performing influence or leverage analysis is to compare the estimated parameter, $\widehat{\boldsymbol{\beta}}$, obtained from the full data, with estimated parameters $\widehat{\boldsymbol{\beta}}_i$, obtained by fitting the model to the $N-1$ subjects remaining after the ith subject is removed. If $\widehat{\boldsymbol{\beta}}-\widehat{\boldsymbol{\beta}}_i$ is close to 0, the ith subject has little influence on the estimate. The process is repeated for all subjects included in the original model. To compute these differences for a dataset with N subjects, we would have to execute `stcox` N additional times, which could be impractical for large datasets.

To avoid fitting N additional Cox models, an approximation to $\widehat{\boldsymbol{\beta}}-\widehat{\boldsymbol{\beta}}_i$ can be made based on the efficient score residuals; see *Methods and formulas*. The difference $\widehat{\boldsymbol{\beta}}-\widehat{\boldsymbol{\beta}}_i$ is commonly referred to as DFBETA in the literature; see [R] **regress postestimation**.

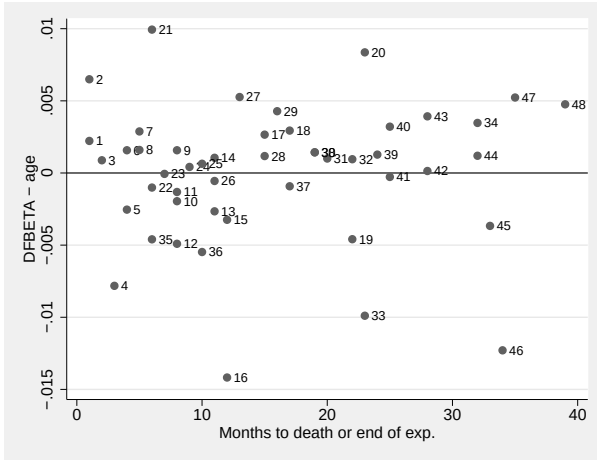
▷ Example 7: DFBETAs

You obtain DFBETAs by using `predict`'s `dfbeta` option:

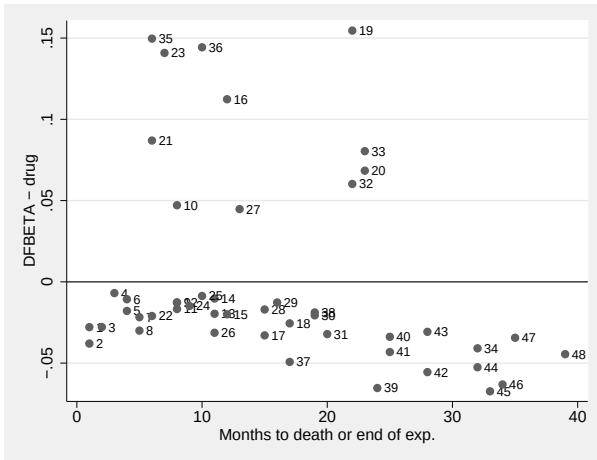
```
. use http://www.stata-press.com/data/r11/drugtr, clear
(Patient Survival in Drug Trial)
. stset studytime, failure(died)
  (output omitted)
. stcox age drug
  (output omitted)
. predict df*, dfbeta
```

The last command saves the estimates of $\text{DFBETA}_i = \widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_i$ for $i = 1, \ldots, N$ in the variables `df1` and `df2`. We can now plot these versus either time or subject (observation) number to identify subjects with disproportionate influence. To maximize the available information, we plot versus time and label the points by their subject numbers.

```
. generate obs = _n
. scatter df1 studytime, yline(0) mlabel(obs)
```




```
. scatter df2 studytime, yline(0) mlabel(obs)
```



From the second graph we see that observation 35, if removed, would decrease the coefficient on `drug` by approximately 0.15 or, equivalently, decrease the hazard ratio for `drug` by a factor of approximately $\exp(-0.15) = 0.861$.

◁

DFBETAs as measures of influence have a straightforward interpretation. Their only disadvantage is that the number of values to examine grows both with sample size and with the number of regressors.

Two alternative measures of influence are *likelihood displacement* values and LMAX values, and both measure each subject's influence on the coefficient vector as a whole. Thus, for each, you have only one value per subject regardless of the number of regressors. As was the case with DFBETAs, likelihood displacement and LMAX calculations are also based on efficient score residuals; see *Methods and formulas*.

Likelihood displacement values measure influence by approximating what happens to the model log likelihood (more precisely, twice the log likelihood) when you omit subject i. Formally, the likelihood displacement value for subject i approximates the quantity

$$2\left\{\log L\left(\widehat{\boldsymbol{\beta}}\right) - \log L\left(\widehat{\boldsymbol{\beta}}_i\right)\right\}$$

where $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\beta}}_i$ are defined as previously and $L(\cdot)$ is the partial likelihood for the Cox model estimated from all the data. In other words, when you calculate $L(\cdot)$, you use all the data, but you evaluate at the parameter estimates $\widehat{\beta}_i$ obtained by omitting the ith subject. Note that because $\widehat{\boldsymbol{\beta}}$ represents an optimal solution, likelihood displacement values will always be nonnegative.

That likelihood displacements measure influence can be seen through the following logic: if subject i is influential, then the vector $\widehat{\boldsymbol{\beta}}_i$ will differ substantially from $\widehat{\boldsymbol{\beta}}$. When that occurs, evaluating the log likelihood at such a suboptimal solution will give you a very different log likelihood.

LMAX values are closely related to likelihood displacements and are derived from an eigensystem analysis of the matrix of efficient score residuals; see *Methods and formulas* for details.

Both likelihood displacement and LMAX values measure each subject's overall influence, but they are not directly comparable with each other. Likelihood displacement values should be compared only with other likelihood displacement values, and LMAX values only with other LMAX values.

▷ Example 8: Likelihood displacement and LMAX values

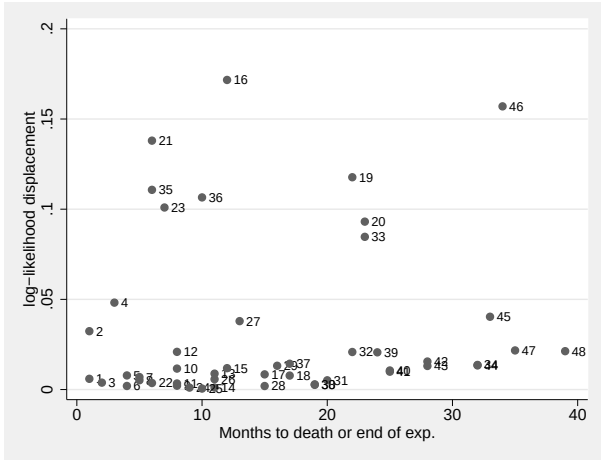
You obtain likelihood displacement values with `predict`'s `ldisplace` option, and you obtain LMAX values with the `lmax` option. Continuing from the previous example:

```
. predict ld, ldisplace
. predict lmax, lmax
. list _t0 _t _d ld lmax in 1/10
```

	_t0	_t	_d	ld	lmax
1.	0	1	1	.0059511	.0735375
2.	0	1	1	.032366	.1124505
3.	0	2	1	.0038388	.0686295
4.	0	3	1	.0481942	.0113989
5.	0	4	1	.0078195	.0331513
6.	0	4	1	.0019887	.0308102
7.	0	5	1	.0069245	.0614247
8.	0	5	1	.0051647	.0763283
9.	0	8	1	.0021315	.0353402
10.	0	8	0	.0116187	.1179539

We can plot the likelihood displacement values versus time and label the points by observation number:

```
. scatter ld studytime, mlabel(obs)
```



The above shows subjects 16 and 46 to be somewhat influential. A plot of LMAX values will show subject 16 as influential but not subject 46, a fact we leave to you to verify.

◁

Schoenfeld residuals and scaled Schoenfeld residuals are most often used to test the proportional-hazards assumption, as described in [ST] **stcox PH-assumption tests**.

Multiple records per subject

In the previous section, we analyzed data from a cancer study, and in doing so we were very loose in differentiating "observations" versus "subjects". In fact, we used both terms interchangeably. We were able to get away with that because in that dataset each subject (patient) was represented by only one observation—the subjects were the observations.

Oftentimes, however, subjects need representation by multiple observations, or records. For example, if a patient leaves the study for some time only to return later, at least one additional record will be needed to denote the subject's return to the study and the gap in their history. If the covariates of interest for a subject change during the study (for example, transitioning from smoking to nonsmoking), then this will also require representation by multiple records.

Multiple records per subject are not a problem for Stata; you simply specify an `id()` variable when `stset`ting your data, and this `id()` variable tells Stata which records belong to which subjects. The other commands in Stata's `st` suite know how to then incorporate this information into your analysis.

For `predict` after `stcox`, by default Stata handles diagnostic measures as always being at the *subject level*, regardless of whether that subject comprises one observation or multiple ones.

▷ Example 9: Stanford heart transplant data

As an example, consider, as we did previously, data from the Stanford heart transplant study:

```
. use http://www.stata-press.com/data/r11/stan3, clear
(Heart transplant data)
. stset
-> stset t1, id(id) failure(died)
                id:  id
     failure event:  died != 0 & died < .
obs. time interval:  (t1[_n-1], t1]
 exit on or before:  failure
------------------------------------------------------------------------------
       172  total obs.
         0  exclusions
------------------------------------------------------------------------------
       172  obs. remaining, representing
       103  subjects
        75  failures in single failure-per-subject data
   31938.1  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      1799
. list id _t0 _t _d age posttran surgery year in 1/10
```

	id	_t0	_t	_d	age	posttran	surgery	year
1.	1	0	50	1	30	0	0	67
2.	2	0	6	1	51	0	0	68
3.	3	0	1	0	54	0	0	68
4.	3	1	16	1	54	1	0	68
5.	4	0	36	0	40	0	0	68
6.	4	36	39	1	40	1	0	68
7.	5	0	18	1	20	0	0	68
8.	6	0	3	1	54	0	0	68
9.	7	0	51	0	50	0	0	68
10.	7	51	675	1	50	1	0	68

The data come to us already `stset`, and we type `stset` without arguments to examine the current settings. We verify that the `id` variable has been set as the patient id. We also see that we have 172 records representing 103 subjects, implying multiple records for some subjects. From our listing, we see that multiple records are necessary to accommodate changes in patients' heart-transplant status (pretransplant versus posttransplant).

Residuals and other diagnostic measures, where applicable, will by default take place at the subject level, meaning that (for example) there will be 103 likelihood displacement values for detecting influential subjects (not observations, but subjects).

```
. stcox age posttran surg year
  (output omitted)
. predict ld, ldisplace
(69 missing values generated)
```

```
. list id _t0 _t _d age posttran surgery year ld in 1/10
```

	id	_t0	_t	_d	age	posttran	surgery	year	ld
1.	1	0	50	1	30	0	0	67	.0596877
2.	2	0	6	1	51	0	0	68	.0154667
3.	3	0	1	0	54	0	0	68	.
4.	3	1	16	1	54	1	0	68	.0298421
5.	4	0	36	0	40	0	0	68	.
6.	4	36	39	1	40	1	0	68	.0359712
7.	5	0	18	1	20	0	0	68	.1260891
8.	6	0	3	1	54	0	0	68	.0199614
9.	7	0	51	0	50	0	0	68	.
10.	7	51	675	1	50	1	0	68	.0659499

Because here we are not interested in predicting any baseline functions, it is perfectly safe to leave `age` and `year` uncentered. The "(69 missing values generated)" message after `predict` tells us that only 103 out of the 172 observations of `ld` were filled in; that is, we received only one likelihood displacement per subject. Regardless of the current sorting of the data, the `ld` value for a subject is stored in the last chronological record for that subject as determined by analysis time, `_t`.

Patient 4 has two records in the data, one pretransplant and one posttransplant. As such, the `ld` value for that patient is interpreted as the change in twice the log likelihood due to deletion of both of these observations, that is, the deletion of patient 4 from the study. The interpretation is at the patient level, not the record level.

◁

If, instead, you want likelihood displacement values that you can interpret at the observation level (that is, changes in twice the log likelihood due to deleting one record), you simply add the `partial` option to the `predict` command above:

```
. predict ld, ldisplace partial
```

We do not think these kinds of observation-level diagnostics are generally what you would want, but they are available.

In the above, we discussed likelihood displacement values, but the same issue concerning subject-level versus observation-level interpretation also exists with Cox–Snell residuals, martingale residuals, deviance residuals, efficient score residuals, LMAX values, and DFBETAs. Regardless of which diagnostic you examine, this issue of interpretation is the same.

There is one situation where you do want to use the `partial` option. If you are using martingale residuals to determine functional form and the variable you are thinking of adding varies within subject, then you want to graph the partial martingale residuals against that new variable. Because the variable changes within subject, the martingale residuals should also change accordingly.

Predictions after stcox with the tvc() option

The residuals and diagnostics discussed previously are not available after estimation with `stcox` with the `tvc()` option, which is a convenience option for handling time-varying covariates:

```
. use http://www.stata-press.com/data/r11/drugtr, clear
(Patient Survival in Drug Trial)
. stcox drug age, tvc(age) nolog
         failure _d:  died
   analysis time _t:  studytime

Cox regression -- Breslow method for ties
No. of subjects =           48                     Number of obs   =        48
No. of failures =           31
Time at risk    =          744
                                                   LR chi2(3)      =     33.63
Log likelihood  =   -83.095036                     Prob > chi2     =    0.0000

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
main         |
        drug |   .1059862   .0478178    -4.97   0.000     .0437737    .2566171
         age |   1.156977     .07018     2.40   0.016     1.027288    1.303037
-------------+----------------------------------------------------------------
tvc          |
         age |   .9970966   .0042415    -0.68   0.494      .988818    1.005445
------------------------------------------------------------------------------
Note: variables in tvc equation interacted with _t
. predict dev, deviance
this prediction is not allowed after estimation with tvc();
see tvc note for an alternative to the tvc() option
r(198);
```

The above fits a Cox model to the cancer data and includes an interaction of `age` with analysis time, `_t`. Such interactions are useful for testing the proportional-hazards assumption: significant interactions are violations of the proportional-hazards assumption for the variable being interacted with analysis time (or some function of analysis time). That is not the situation here.

In any case, models with `tvc()` interactions do not allow predicting the residuals and diagnostics discussed thus far. The solution in such situations is to forgo the use of `tvc()`, expand the data, and use factor variables to specify the interaction:

```
. generate id = _n
. streset, id(id)
  (output omitted)
. stsplit, at(failures)
(21 failure times)
(534 observations (episodes) created)
```

```
. stcox drug age c.age#c._t, nolog
        failure _d:  died
  analysis time _t:  studytime
                id:  id
Cox regression -- Breslow method for ties
No. of subjects =           48                  Number of obs    =       582
No. of failures =           31
Time at risk    =          744
                                                LR chi2(3)       =     33.63
Log likelihood  =   -83.095036                  Prob > chi2      =    0.0000
```

_t	Haz. Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
drug	.1059862	.0478178	-4.97	0.000	.0437737	.2566171
age	1.156977	.07018	2.40	0.016	1.027288	1.303037
c.age#c._t	.9970966	.0042415	-0.68	0.494	.988818	1.005445

```
. predict dev, deviance
(534 missing values generated)
. summarize dev
```

Variable	Obs	Mean	Std. Dev.	Min	Max
dev	48	.0658485	1.020993	-1.804876	2.065424

We split the observations, currently one per subject, so that the interaction term is allowed to vary over time. Splitting the observations requires that we first establish a subject `id` variable. Once that is done, we split the observations with `stsplit` and the `at(failures)` option, which splits the records only at the observed failure times. This amount of splitting is the minimal amount required to reproduce our previous Cox model. We then include the interaction term `c.age#c._t` in our model, verify that our Cox model is the same as before, and obtain our 48 deviance residuals, one for each subject.

Predictions after stcox with the shared() option

A Cox shared frailty model is a Cox model with added group-level random effects such that

$$h_{ij}(t) = h_0(t)\exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \nu_i)$$

with ν_i representing the added effect due to being in group i; see *Cox regression with shared frailty* in [ST] **stcox** for more details. You fit this kind of model by specifying the `shared(`*varname*`)` option with `stcox`, where *varname* identifies the groups. `stcox` will produce an estimate of $\boldsymbol{\beta}$, its covariance matrix, and an estimate of the variance of the ν_i. What it will not produce are estimates of the ν_i themselves. These you can obtain postestimation with `predict`.

▷ Example 10: Shared frailty models

In example 9 of [ST] **stcox**, we fit a shared frailty model to data from 38 kidney dialysis patients, measuring the time to infection at the catheter insertion point. Two recurrence times (in days) were measured for each patient.

The estimated ν_i are not displayed in the `stcox` coefficient table but may be retrieved postestimation by using `predict` with the `effects` option:

```
. use http://www.stata-press.com/data/r11/catheter, clear
(Kidney data, McGilchrist and Aisbett, Biometrics, 1991)
. qui stcox age female, shared(patient)
. predict nu, effects
. sort nu
. list patient nu in 1/2
```

	patient	nu
1.	21	-2.448707
2.	21	-2.448707

```
. list patient nu in 75/L
```

	patient	nu
75.	7	.5187159
76.	7	.5187159

From the results above, we estimate that the least frail patient is patient 21, with $\widehat{\nu}_{21} = -2.45$, and that the frailest patient is patient 7, with $\widehat{\nu}_7 = 0.52$.

◁

❑ Technical note

When used with shared-frailty models, `predict`'s `basehc`, `basesurv`, and `basechazard` options produce estimates of baseline quantities that are based on the last-step penalized Cox model fit. Therefore, the term *baseline* means that not only are the covariates set to 0 but the ν_i are as well.

Other predictions, such as martingale residuals, are conditional on the estimated frailty variance being fixed and known at the onset.

❑

estat concordance

`estat concordance` calculates the concordance probability, which is defined as the probability that predictions and outcomes are concordant. `estat concordance` provides two measures of the concordance probability: Harrell's C and Gönen and Heller's K concordance coefficients. Harrell's C, which is defined as the proportion of all usable subject pairs in which the predictions and outcomes are concordant, is computed by default. Gönen and Heller (2005) propose an alternative measure of concordance, computed when the `gheller` option is specified, that is not sensitive to the degree of censoring, unlike Harrell's C coefficient. This estimator is not dependent on the observed event or the censoring time and is a function of only the regression parameters and the covariate distribution, which leads to the asymptotic unbiasedness. `estat concordance` also reports the Somers' D rank correlation, which is derived by calculating $2C - 1$ for Harrell's C and $2K - 1$ for Gönen and Heller's K.

`estat concordance` may not be used after a Cox regression model with time-varying covariates and may not be applied to weighted data or to data with delayed entries. The computation of Gönen and Heller's K coefficient is not supported for shared-frailty models, stratified estimation, or multiple-record data.

▷ Example 11: Harrell's C

Using our cancer data, we wish to evaluate the predictive value of the measurement of `drug` and `age`. After fitting a Cox regression model, we use `estat concordance` to calculate Harrell's C index.

```
. use http://www.stata-press.com/data/r11/drugtr, clear
(Patient Survival in Drug Trial)
. stcox drug age
        failure _d:  died
  analysis time _t:  studytime
Iteration 0:   log likelihood = -99.911448
Iteration 1:   log likelihood = -83.551879
Iteration 2:   log likelihood = -83.324009
Iteration 3:   log likelihood = -83.323546
Refining estimates:
Iteration 0:   log likelihood = -83.323546
Cox regression -- Breslow method for ties
No. of subjects =           48                     Number of obs   =        48
No. of failures =           31
Time at risk    =          744
                                                   LR chi2(2)      =     33.18
Log likelihood  =   -83.323546                     Prob > chi2     =    0.0000
------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        drug |   .1048772   .0477017    -4.96   0.000     .0430057    .2557622
         age |   1.120325   .0417711     3.05   0.002     1.041375     1.20526
------------------------------------------------------------------------------
. estat concordance, noshow
  Harrell's C concordance statistic
  Number of subjects (N)              =       48
  Number of comparison pairs (P)      =      849
  Number of orderings as expected (E) =      679
  Number of tied predictions (T)      =       15
          Harrell's C = (E + T/2) / P =   .8086
                      Somers' D =     .6172
```

The result of `stcox` shows that the drug results in a lower hazard and therefore a longer survival time, controlling for age and older patients being more likely to die. The value of Harrell's C is 0.8086, which indicates that we can correctly order survival times for pairs of patients 81% of the time on the basis of measurement of `drug` and `age`. See *Methods and formulas* for the full definition of concordance.

◁

❑ Technical note

`estat concordance` does not work after a Cox regression model with time-varying covariates. When the covariates are varying with time, the prognostic score, PS $= \mathbf{x}\beta$, will not capture or condense the information in given measurements, in which case it does not make sense to calculate the rank correlation between PS and survival time.

❑

▷ Example 12: Gönen and Heller's K

Alternatively, we can obtain Gönen and Heller's estimate of the concordance probability, K. To do so, we specify the `gheller` option with `estat concordance`:

```
. estat concordance, noshow gheller
  Gonen and Heller's K concordance statistic
  Number of subjects (N)       =       48
          Gonen and Heller's K =    .7748
                     Somers' D =    .5496
```

Gönen and Heller's concordance coefficient may be preferred to Harrell's C when censoring is present because Harrell's C can be biased. Because 17 of our 48 subjects are censored, we prefer Gönen and Heller's concordance to Harrell's C.

◁

Saved results

`estat concordance` saves the following in `r()`:

Scalars

`r(N)`	number of observations	`r(K)`	Gönen and Heller's K coefficient
`r(n_P)`	number of comparison pairs	`r(K_s)`	smoothed Gönen and Heller's K coefficient
`r(n_E)`	number of orderings as expected	`r(K_s_se)`	standard error of the smoothed K coefficient
`r(n_T)`	number of tied predictions	`r(D)`	Somers' D coefficient for Harrell's C
`r(C)`	Harrell's C coefficient	`r(D_K)`	Somers' D coefficient for Gönen and Heller's K

`r(n_P)`, `r(n_E)`, and `r(n_T)` are returned only when strata are not specified.

Methods and formulas

All methods presented in this entry have been implemented as ado-files that use Mata.

Let $\mathbf{x}_i$ be the row vector of covariates for the time interval $(t_{0i}, t_i\,]$ for the ith observation in the dataset ($i = 1, \ldots, N$). The Cox partial log-likelihood function, using the default Peto–Breslow method for tied failures is

$$\log L_{\text{breslow}} = \sum_{j=1}^{D} \sum_{i \in D_j} \left[w_i(\mathbf{x}_i\boldsymbol{\beta} + \text{offset}_i) - w_i \log \left\{ \sum_{\ell \in R_j} w_\ell \exp(\mathbf{x}_\ell\boldsymbol{\beta} + \text{offset}_\ell) \right\} \right]$$

where j indexes the ordered failure times t_j ($j = 1, \ldots, D$), D_j is the set of d_j observations that fail at t_j, d_j is the number of failures at t_j, and R_j is the set of observations k that are at risk at time t_j (that is, all k such that $t_{0k} < t_j \leq t_k$). w_i and offset_i are, respectively, the weight and linear offset for observation i, if specified.

If the Efron method for ties is specified at estimation, the partial log likelihood is

$$\log L_{\text{efron}} = \sum_{j=1}^{D} \sum_{i \in D_j} \left[\mathbf{x}_i\boldsymbol{\beta} + \text{offset}_i - d_j^{-1} \sum_{k=0}^{d_j-1} \log \left\{ \sum_{\ell \in R_j} \exp(\mathbf{x}_\ell\boldsymbol{\beta} + \text{offset}_\ell) - kA_j \right\} \right]$$

for $A_j = d_j^{-1} \sum_{\ell \in D_j} \exp(\mathbf{x}_\ell\boldsymbol{\beta} + \text{offset}_\ell)$. Weights are not supported with the Efron method.

At estimation, Stata also supports the exact marginal and exact partial methods for handling ties, but only the Peto–Breslow and Efron methods are supported in regard to the calculation of residuals, diagnostics, and other predictions. As such, only the partial log-likelihood formulas for those two methods are presented above, for easier reference in what follows.

If you specified `efron` at estimation, all predictions are carried out using the Efron method; that is, the handling of tied failures is done analogously to the way it was done when calculating $\log L_{\text{efron}}$. If you specified `breslow` (or nothing, because `breslow` is the default), `exactm`, or `exactp`, all predictions are carried out using the Peto–Breslow method. That is not to say that if you specify `exactm` at estimation, your predictions will be the same as if you had specified `breslow`. The formulas used will be the same, but the parameter estimates at which they are evaluated will differ because those were based on different ways of handling ties.

Define $z_i = \mathbf{x}_i\widehat{\boldsymbol{\beta}} + \text{offset}_i$. Schoenfeld residuals for the pth variable using the Peto–Breslow method are given by

$$r_{S_{pi}} = \delta_i \left(x_{pi} - a_{pi}\right)$$

where

$$a_{pi} = \frac{\sum_{\ell \in R_i} w_\ell x_{p\ell} \exp(z_\ell)}{\sum_{\ell \in R_i} w_\ell \exp(z_\ell)}$$

δ_i indicates failure for observation i, and x_{pi} is the pth element of $\mathbf{x}_i$. For the Efron method, Schoenfeld residuals are

$$r_{S_{pi}} = \delta_i \left(x_{pi} - b_{pi}\right)$$

where

$$b_{pi} = d_i^{-1} \sum_{k=0}^{d_i-1} \frac{\sum_{\ell \in R_i} x_{p\ell} \exp(z_\ell) - kd_i^{-1} \sum_{\ell \in D_i} x_{p\ell} \exp(z_\ell)}{\sum_{\ell \in R_i} \exp(z_\ell) - kd_i^{-1} \sum_{\ell \in D_i} \exp(z_\ell)}$$

Schoenfeld residuals are derived from the first derivative of the log likelihood, with

$$\left.\frac{\partial \log L}{\partial \beta_p}\right|_{\widehat{\boldsymbol{\beta}}} = \sum_{i=1}^{N} r_{S_{pi}} = 0$$

and only those observations that fail ($\delta_i = 1$) contribute a Schoenfeld residual to the derivative.

For censored observations, Stata stores a missing value for the Schoenfeld residual even though the above implies a value of 0. This is to emphasize that no calculation takes place when the observation is censored.

Scaled Schoenfeld residuals are given by

$$\mathbf{r}^*_{S_i} = \widehat{\boldsymbol{\beta}} + d \ \text{Var}(\widehat{\boldsymbol{\beta}})\mathbf{r}_{S_i}$$

where $\mathbf{r}_{S_i} = (r_{S_{1i}}, \ldots, r_{S_{mi}})'$, m is the number of regressors, and d is the total number of failures.

In what follows, we assume the Peto–Breslow method for handling ties. Formulas for the Efron method, while tedious, can be obtained by applying similar principles of averaging across risk sets, as demonstrated above with Schoenfeld residuals.

Efficient score residuals are obtained by

$$r_{E_{pi}} = r_{S_{pi}} - \exp(z_i) \sum_{j:t_{0i}<t_j\le t_i} \frac{\delta_j w_j (x_{pi} - a_{pj})}{\sum_{\ell\in R_j} w_\ell \exp(z_\ell)}$$

Like Schoenfeld residuals, efficient score residuals are also additive components of the first derivative of the log likelihood. Whereas Schoenfeld residuals are the contributions of each failure, efficient score residuals are the contributions of each observation. Censored observations contribute to the log likelihood (and its derivative) because they belong to risk sets at times when other observations fail. As such, an observation's contribution is twofold: 1) If the observation ends in failure, a risk assessment is triggered, that is, a term in the log likelihood is computed. 2) Whether failed or censored, an observation contributes to risk sets for other observations that do fail. Efficient score residuals reflect both contributions.

The above computes efficient score residuals at the observation level. If you have multiple records per subject and do not specify the `partial` option, then the efficient score residual for a given subject is calculated by summing the efficient scores over the observations within that subject.

Martingale residuals are

$$r_{M_i} = \delta_i - \exp(z_i) \sum_{j:t_{0i}<t_j\le t_i} \frac{w_j \delta_j}{\sum_{\ell\in R_j} w_\ell \exp(z_\ell)}$$

The above computes martingale residuals at the observation level. If you have multiple records per subject and do not specify the `partial` option, then the martingale residual for a given subject is calculated by summing r_{M_i} over the observations within that subject.

Martingale residuals are in the range $(-\infty, 1)$. Deviance residuals are transformations of martingale residuals designed to have a distribution that is more symmetric about zero. Deviance residuals are calculated using

$$r_{D_i} = \mathrm{sign}(r_{M_i}) \left[-2 \left\{ r_{M_i} + \delta_i \log(\delta_i - r_{M_i}) \right\} \right]^{1/2}$$

These residuals are expected to be symmetric about zero but do not necessarily sum to zero.

The above computes deviance residuals at the observation level. If you have multiple records per subject and do not specify the `partial` option, then the deviance residual for a given subject is calculated by applying the above transformation to the *subject-level* martingale residual.

The estimated baseline hazard contribution is obtained at each failure time as $h_j = 1 - \widehat{\alpha}_j$, where $\widehat{\alpha}_j$ is the solution to

$$\sum_{k\in D_j} \frac{\exp(z_k)}{1 - \widehat{\alpha}_j^{\exp(z_k)}} = \sum_{\ell\in R_j} \exp(z_\ell)$$

(Kalbfleisch and Prentice 2002, eq. 4.34, 115).

The estimated baseline survivor function is

$$\widehat{S}_0(t) = \prod_{j:t_j \le t} \widehat{\alpha}_j$$

When estimated with no covariates, $\widehat{S}_0(t)$ is the Kaplan–Meier estimate of the survivor function.

The estimated baseline cumulative hazard function, if requested, is related to the baseline survivor function calculation, yet the values of $\widehat{\alpha}_j$ are set at their starting values and are not iterated. Equivalently,

$$\widehat{H}_0(t) = \sum_{j:t_j \le t} \frac{d_j}{\sum_{\ell \in R_j} \exp(z_\ell)}$$

When estimated with no covariates, $\widehat{H}_0(t)$ is the Nelson–Aalen estimate of the cumulative hazard.

Cox–Snell residuals are calculated with

$$r_{C_i} = \delta_i - r_{M_i}$$

where r_{M_i} are the martingale residuals. Equivalently, Cox–Snell residuals can be obtained with

$$r_{C_i} = \exp(z_i)\widehat{H}_0(t_i)$$

The above computes Cox–Snell residuals at the observation level. If you have multiple records per subject and do not specify the `partial` option, then the Cox–Snell residual for a given subject is calculated by summing r_{C_i} over the observations within that subject.

DFBETAs are calculated with

$$\text{DFBETA}_i = \mathbf{r}_{E_i}\widetilde{\text{Var}}(\widehat{\boldsymbol{\beta}})$$

where $\mathbf{r}_{E_i} = (r_{E_{1i}}, \ldots, r_{E_{mi}})$ is a row vector of efficient score residuals with one entry for each regressor, and $\widetilde{\text{Var}}(\widehat{\boldsymbol{\beta}})$ is the model-based variance matrix of $\widehat{\boldsymbol{\beta}}$.

Likelihood displacement values are calculated with

$$\text{LD}_i = \mathbf{r}_{E_i}\text{Var}(\widehat{\boldsymbol{\beta}})\mathbf{r}'_{E_i}$$

(Collett 2003, 136). In both of the above, $\mathbf{r}_{E_i}$ can represent either one observation or, in multiple-record data, the cumulative efficient score for an entire subject. For the former, the interpretation is that due to deletion of one record; for the latter, the interpretation is that due to deletion of all a subject's records.

Following Collett (2003, 137), LMAX values are obtained from an eigensystem analysis of

$$\mathbf{B} = \boldsymbol{\Theta}\ \text{Var}(\widehat{\boldsymbol{\beta}})\ \boldsymbol{\Theta}'$$

where $\boldsymbol{\Theta}$ is the $N \times m$ matrix of efficient score residuals, with element (i, j) representing the jth regressor and the ith observation (or subject). LMAX values are then the absolute values of the elements of the unit-length eigenvector associated with the largest eigenvalue of the $N \times N$ matrix $\mathbf{B}$.

For shared-frailty models, the data are organized into G groups, with the ith group consisting of n_i observations, $i = 1, \ldots, G$. From Therneau and Grambsch (2000, 253–255), for fixed θ, estimates of $\boldsymbol{\beta}$ and $\nu_1, \ldots, \nu_G$ are obtained by maximizing

$$\log L(\theta) = \log L_{\text{Cox}}(\boldsymbol{\beta}, \nu_1, \ldots, \nu_G) + \sum_{i=1}^{G} \left[\frac{1}{\theta} \{\nu_i - \exp(\nu_i)\} + \left(\frac{1}{\theta} + D_i\right) \left\{1 - \log\left(\frac{1}{\theta} + D_i\right)\right\} - \frac{\log\theta}{\theta} + \log\Gamma\left(\frac{1}{\theta} + D_i\right) - \log\Gamma\left(\frac{1}{\theta}\right) \right]$$

where D_i is the number of death events in group i, and $\log L_{\text{Cox}}(\boldsymbol{\beta}, \nu_1, \ldots, \nu_G)$ is the standard Cox partial log likelihood, with the ν_i treated as the coefficients of indicator variables identifying the groups. That is, the jth observation in the ith group has log relative-hazard $\mathbf{x}_{ij}\boldsymbol{\beta} + \nu_i$.

You obtain the estimates of $\nu_1, \ldots, \nu_G$ with `predict`'s `effects` option after `stcox, shared()`.

estat concordance

Harrell's C was proposed by Harrell Jr. et al. (1982) and was developed to evaluate the results of a medical test. The C index is defined as the proportion of all usable subject pairs in which the predictions and outcomes are concordant. The C index may be applied to ordinary continuous outcomes, dichotomous diagnostic outcomes, ordinal outcomes, and censored time-until-event response variables.

In predicting the time until death, C is calculated by considering all comparable patient pairs. A pair of patients is comparable if either 1) the two have different values on the time variable, and the one with the lowest value presents a failure, or 2) the two have the same value on the time variable, and exactly one of them presents a failure. If the predicted survival time is larger for the patient who lived longer, the predictions for the pair are said to be concordant with the outcomes. From Fibrinogen Studies Collaboration (2009), Harrell's C is defined as $\sum_k (E_k + T_k/2) / \sum_k (D_k)$, where D_k is the total number of pairs usable for comparison in stratum k, E_k is the number of pairs for which the predictions are concordant with the outcomes and the predictions are not identical in stratum k, and T_k is the number of usable pairs for which the predictions are identical in stratum k. If there are no strata specified, then the formula for Harrell's C reduces to $(E + T/2)/D$.

For a Cox proportional hazards model, the probability that the patient survives past time t is given by $S_0(t)$ raised to the $\exp(\mathbf{x}\boldsymbol{\beta})$ power, where $S_0(t)$ is the baseline survivor function, $\mathbf{x}$ denotes a set of measurements for the patient, and $\boldsymbol{\beta}$ is the vector of coefficients. A Cox regression model is fit by the `stcox` command. The hazard ratio, $\exp(\mathbf{x}\boldsymbol{\beta})$, is obtained by `predict` after `stcox`. Because the predicted survivor time and the predicted survivor function are one-to-one functions of each other, the predicted survivor function can be used to calculate C instead of the predicted survival time. The predicted survivor function decreases when the predicted hazard ratio increases; therefore, Harrell's C can be calculated by computing E, T, and D, based on the observed outcomes and the predicted hazard ratios.

C takes a value between 0 and 1. A value of 0.5 indicates no predictive discrimination, and values of 0 or 1.0 indicate perfect separation of subjects with different outcomes. See Harrell Jr., Lee, and Mark (1996) for more details. Somers' D rank correlation is calculated by $2C - 1$; see Newson (2002) for a discussion of Somers' D.

In the presence of censoring, Harrell's C coefficient tends to be biased. An alternative measure of concordance that is asymptotically unbiased with censored data was proposed by Gönen and Heller (2005). This estimator does not depend on observed time directly and is a function of only the regression parameters and the covariate distribution, which leads to its asymptotic unbiasedness and thus robustness to the degree of censoring.

Let $\Delta\mathbf{x}_{ij}$ be the pairwise difference $\mathbf{x}_i - \mathbf{x}_j$. Then Gönen and Heller's concordance probability estimator is given by

$$K = K_N(\widehat{\boldsymbol{\beta}}) = \frac{2}{N(N-1)} \sum_{i<j} \sum \left\{ \frac{I(\Delta\mathbf{x}_{ji}\widehat{\boldsymbol{\beta}} < 0)}{1+\exp(\Delta\mathbf{x}_{ji}\widehat{\boldsymbol{\beta}})} + \frac{I(\Delta\mathbf{x}_{ij}\widehat{\boldsymbol{\beta}} < 0)}{1+\exp(\Delta\mathbf{x}_{ij}\widehat{\boldsymbol{\beta}})} \right\}$$

where $I(\cdot)$ is an indicator function. Somers' D rank correlation is calculated by $2K - 1$.

References

Collett, D. 2003. *Modelling Survival Data in Medical Research*. 2nd ed. London: Chapman & Hall/CRC.

Fibrinogen Studies Collaboration. 2009. Measures to assess the prognostic ability of the stratified Cox proportional hazards model. *Statistics in Medicine* 28: 389–411.

Gönen, M., and G. Heller. 2005. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika* 92: 965–970.

Harrell Jr., F. E., R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati. 1982. Evaluating the yield of medical tests. *Journal of the American Medical Association* 247: 2543–2546.

Harrell Jr., F. E., K. L. Lee, and D. B. Mark. 1996. Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine* 15: 361–387.

Kalbfleisch, J. D., and R. L. Prentice. 2002. *The Statistical Analysis of Failure Time Data*. 2nd ed. New York: Wiley.

Klein, J. P., and M. L. Moeschberger. 2003. *Survival Analysis: Techniques for Censored and Truncated Data*. 2nd ed. New York: Springer.

Newson, R. 2002. Parameters behind "nonparametric" statistics: Kendall's tau, Somers' D and median differences. *Stata Journal* 2: 45–64.

——. 2006. Confidence intervals for rank statistics: Somers' D and extensions. *Stata Journal* 6: 309–334.

Rogers, W. H. 1994. ssa4: Ex post tests and diagnostics for a proportional hazards model. *Stata Technical Bulletin* 19: 23–27. Reprinted in *Stata Technical Bulletin Reprints*, vol. 4, pp. 186–191. College Station, TX: Stata Press.

Schoenfeld, D. A. 1982. Partial residuals for the proportional hazards regression model. *Biometrika* 69: 239–241.

Therneau, T. M., and P. M. Grambsch. 2000. *Modeling Survival Data: Extending the Cox Model*. New York: Springer.

Also see

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

[ST] **stcurve** — Plot survivor, hazard, cumulative hazard, or cumulative incidence function

[U] **20 Estimation and postestimation commands**