Title

stata.com

stcrreg — Competing-risks regression

Syntax

```
stcrreg [varlist] [if] [in], compete(crvar[==numlist]) [options]
```

options	Description
Model	
* `compete(`*crvar*`[==`*numlist*`])`	specify competing-risks event(s)
`tvc(`*tvarlist*`)`	time-varying covariates
`texp(`*exp*`)`	multiplier for time-varying covariates; default is `texp(_t)`
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
`collinear`	keep collinear variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
`noadjust`	do not use standard degree-of-freedom adjustment
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`noshr`	report coefficients, not subhazard ratios
`noshow`	do not show st setting information
`noheader`	suppress header from coefficient table
`notable`	suppress coefficient table
`nodisplay`	suppress output; iteration log is still displayed
`nocnsreport`	do not display constraints
display_options	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
`coeflegend`	display legend instead of statistics

*`compete(`*crvar*`[==`*numlist*`])` is required.

You must `stset` your data before using `stcrreg`; see [ST] **stset**.

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

`bootstrap`, `by`, `fp`, `jackknife`, `mfp`, `mi estimate`, `nestreg`, `statsby`, and `stepwise` are allowed; see [U] **11.1.10 Prefix commands**.

`vce(bootstrap)` and `vce(jackknife)` are not allowed with the `mi estimate` prefix; see [MI] **mi estimate**.

Weights are not allowed with the `bootstrap` prefix; see [R] **bootstrap**.

`fweights`, `iweights`, and `pweights` may be specified using `stset`; see [ST] **stset**. In multiple-record data, weights are applied to subjects as a whole, not to individual observations. `iweights` are treated as `fweights` that can be noninteger, but not negative.

`coeflegend` does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Menu

Statistics > Survival analysis > Regression models > Competing-risks regression

Description

`stcrreg` fits, via maximum likelihood, competing-risks regression models on st data, according to the method of Fine and Gray (1999). Competing-risks regression posits a model for the subhazard function of a failure event of primary interest. In the presence of competing failure events that impede the event of interest, a standard analysis using Cox regression (see [ST] **stcox**) is able to produce incidence-rate curves that either 1) are appropriate only for a hypothetical universe where competing events do not occur or 2) are appropriate for the data at hand, yet the effects of covariates on these curves are not easily quantified. Competing-risks regression, as performed using `stcrreg`, provides an alternative model that can produce incidence curves that represent the observed data and for which describing covariate effects is straightforward.

`stcrreg` can be used with single- or multiple-record data. `stcrreg` cannot be used when you have multiple failures per subject.

Options

Model

`compete(`*crvar*`[==`*numlist*`])` is required and specifies the events that are associated with failure due to competing risks.

If `compete(`*crvar*`)` is specified, *crvar* is interpreted as an indicator variable; any nonzero, nonmissing values are interpreted as representing competing events.

If `compete(`*crvar*`==`*numlist*`)` is specified, records with *crvar* taking on any of the values in *numlist* are assumed to be competing events.

The syntax for `compete()` is the same as that for `stset`'s `failure()` option. Use `stset, failure()` to specify the failure event of interest, that is, the failure event you wish to model using `stcox`, `streg`, `stcrreg`, or whatever. Use `stcrreg, compete()` to specify the event or events that compete with the failure event of interest. Competing events, because they are not the failure event of primary interest, must be `stset` as censored.

If you have multiple records per subject, only the value of *crvar* for the last chronological record for each subject is used to determine the event type for that subject.

`tvc(`*tvarlist*`)` specifies those variables that vary continuously with respect to time, that is, time-varying covariates. These variables are multiplied by the function of time specified in `texp()`.

`texp(`*exp*`)` is used in conjunction with `tvc(`*tvarlist*`)` to specify the function of analysis time that should be multiplied by the time-varying covariates. For example, specifying `texp(ln(_t))` would cause the time-varying covariates to be multiplied by the logarithm of analysis time. If `tvc(`*tvarlist*`)` is used without `texp(`*exp*`)`, Stata understands that you mean `texp(_t)`, and thus multiplies the time-varying covariates by the analysis time.

Both `tvc(`*tvarlist*`)` and `texp(`*exp*`)` are explained more in *Option tvc() and testing the proportional-subhazards assumption* below.

`offset(`*varname*`)`, `constraints(`*constraints*`)`, `collinear`; see [R] **estimation options**.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] ***vce_option***. `vce(robust)` is the default in single-record-per-subject st data. For multiple-record st data, `vce(cluster` *idvar*`)` is the default, where *idvar* is the ID variable previously `stset`.

Standard Hessian-based standard errors—*vcetype* `oim`—are not statistically appropriate for this model and thus are not allowed.

`noadjust` is for use with `vce(robust)` or `vce(cluster` *clustvar*`)`. `noadjust` prevents the estimated variance matrix from being multiplied by $N/(N-1)$ or $g/(g-1)$, where g is the number of clusters. The default adjustment is somewhat arbitrary because it is not always clear how to count observations or clusters. In such cases, however, the adjustment is likely to be biased toward 1, so we would still recommend making it.

Reporting

`level(#)`; see [R] **estimation options**.

`noshr` specifies that coefficients be displayed rather than exponentiated coefficients or subhazard ratios. This option affects only how results are displayed and not how they are estimated. `noshr` may be specified at estimation time or when redisplaying previously estimated results (which you do by typing `stcrreg` without a variable list).

`noshow` prevents `stcrreg` from showing the key st variables. This option is seldom used because most people type `stset, show` or `stset, noshow` to set whether they want to see these variables mentioned at the top of the output of every st command; see [ST] **stset**.

`noheader` suppresses the header information from the output. The coefficient table is still displayed. `noheader` may be specified at estimation time or when redisplaying previously estimated results.

`notable` suppresses the table of coefficients from the output. The header information is still displayed. `notable` may be specified at estimation time or when redisplaying previously estimated results.

`nodisplay` suppresses the output. The iteration log is still displayed.

`nocnsreport`; see [R] **estimation options**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **maximize**. These options are seldom used.

The following option is available with stcrreg but is not shown in the dialog box:

coeflegend; see [R] **estimation options**.

Remarks and examples

stata.com

This section provides a summary of what can be done with stcrreg. For a more general tutorial on competing-risks analysis, see Cleves et al. (2010, chap. 17).

Remarks are presented under the following headings:

The case for competing-risks regression

In this section, we provide a brief history and literature review of competing-risks analysis, and provide the motivation behind the stcrreg model. If you know you want to use stcrreg and are anxious to get started, you can safely skip this section.

Based on the method of Fine and Gray (1999), competing-risks regression provides a useful alternative to Cox regression (Cox 1972) for survival data in the presence of competing risks. Consider the usual survival analysis where one measures time-to-failure as a function of experimental or observed factors. For example, we may be interested in measuring time from initial treatment to recurrence of breast cancer in relation to factors such as treatment type and smoking status. The term *competing risk* refers to the chance that instead of cancer recurrence, you will observe a *competing event*, for example, death. The competing event, death, impedes the occurrence of the *event of interest*, breast cancer. This is not to be confused with the usual right-censoring found in survival data, such as censoring due to loss to follow-up. When subjects are lost to follow-up, they are still considered at risk of recurrent breast cancer—it is just that the researcher is not in a position to record the precise time that it happens. In contrast, death is a permanent condition that prevents future breast cancer. While censoring merely obstructs you from observing the event of interest, a competing event prevents the event of interest from occurring altogether. Because competing events are distinct from standard censorings, a competing-risks analysis requires some new methodology and some caution when interpreting the results from the old methodology.

Putter, Fiocco, and Geskus (2007) and Gichangi and Vach (2005) provide excellent tutorials covering the problem of competing risks, nonparametric estimators and tests, competing-risks regression, and the more general multistate models. Textbook treatments of competing-risks analysis can be found within Andersen et al. (1993), Klein and Moeschberger (2003), Therneau and Grambsch (2000), and Marubini and Valsecchi (1997). The texts by Crowder (2001) and Pintilie (2006) are devoted entirely to the topic. In what follows, we assume that you are familiar with the basic concepts of survival analysis, for example, hazard functions and Kaplan–Meier curves. For such an introduction to survival analysis aimed at Stata users, see Cleves et al. (2010).

Without loss of generality, assume a situation where there is only one event that competes with the failure event of interest. Before analyzing the problem posed by competing-risks data—the problem `stcrreg` proposes to solve—we first formalize the mechanism behind it. Ignoring censoring for the moment, recording a failure time in a competing-risks scenario can be represented as observing the minimum of two potential failures times: the time to the event of interest, T_1, and the time to the competing event, T_2. The problem of competing risks then becomes one of understanding the nature of the bivariate distribution of (T_1, T_2), and in particular the correlation therein. Although conceptually simple, unfortunately this joint distribution cannot be identified by the data (Pepe and Mori 1993; Tsiatis 1975; Gail 1975). If you get to observe only the minimum, you are getting only half the picture.

An alternate representation of the competing-risks scenario that relies on quantities that are data-identifiable is described by Beyersman et al. (2009). In that formulation, we consider the hazard for the event of interest, $h_1(t)$, and that for the competing event, $h_2(t)$. Both hazards can be estimated from available data and when combined form a total hazard that any event will occur equal to $h(t) = h_1(t) + h_2(t)$. As risk accumulates according to $h(t)$, event times T are observed. Whether these events turn out to be failures of interest (type 1) or competing events (type 2) is determined by the two component hazards at that precise time. The event will be a failure of interest with probability $h_1(T)/\{h_1(T) + h_2(T)\}$, or a competing event with probability one minus that.

Instead of focusing on the survivor function for the event of interest, $P(T > t \text{ and event type } 1)$, when competing risks are present you want to focus on the failure function, $P(T \leq t \text{ and event type } 1)$, also known as the *cumulative incidence function* (CIF). That is because you will not know what type of event will occur until after it has occurred. It makes more sense to ask "What is the probability of breast cancer within 5 months?" than to ask "What is probability that nothing happens before 5 months, and that when something does happen, it will be breast cancer and not death?"

Much of the literature on competing risks focuses on the inadequacy of the Kaplan–Meier (1958) estimator (which we refer to as KM) as a measure of prevalence for the event of interest. Among others, Gooley et al. (1999) point out that 1−KM is a biased estimate of the CIF. The bias results from KM treating competing events as if they were censored. That is, subjects that experience competing events are treated as if they could later experience the event of interest, even though that is impossible. Although you could interpret 1−KM as the probability of a type 1 failure in a hypothetical setting where type 2 failures do not occur, this requires you to assume that $h_1(t)$ remains unchanged given that $h_2(t) = 0$, a rather strong and untestable assumption. Regardless of whether the independence assumption holds, 1−KM is still not representative of the data at hand, under which competing events do take place.

As such, 1−KM should be rejected in favor of the *cumulative incidence estimator* of the CIF; see Coviello and Boggess (2004) for a Stata-specific presentation. The cumulative incidence estimator is superior to 1−KM because it acknowledges that cumulative incidence is a function of both cause-specific hazards, $h_1(t)$ and $h_2(t)$. Conversely, 1−KM treats the CIF as a function solely of $h_1(t)$.

When you have covariates, you can use `stcox` to perform regression on $h_1(t)$ by treating failures of type 2 as censored, on $h_2(t)$ by treating failures of type 1 as censored, or on $h_1(t)$ and $h_2(t)$ simultaneously by using the method of data duplication described by Lunn and McNeil (1995) and Cleves (1999). Because cause-specific hazards are identified by the data, all three of the above analyses are suitable for estimating how covariates affect the mechanism behind a given type of failure. For example, if you are interested in how smoking affects breast cancer in general terms (competing death notwithstanding), then a Cox model for $h_1(t)$ that treats death as censored is perfectly valid; see Pintilie (2007).

If you are interested in the incidence of breast cancer, however, you want to use a Cox model that models both $h_1(t)$ and $h_2(t)$, because the CIF for breast cancer will likely depend on both. Based on the fitted model, you will have a hard time spotting the effects of covariates on cumulative incidence,

because the covariates can affect $h_1(t)$ and $h_2(t)$ differently, and the CIF is a nonlinear function of these effects and of the baseline hazards. Whether increasing a covariate increases or decreases the cumulative incidence depends on time and on the nominal value of that covariate, as well as on the values of the other covariates. There is no way to determine the full effects of the covariates by just looking at the model coefficients. You would have to estimate and graph the CIF for various sets of covariate values, and this requires a bit of programming; see example 4.

An alternative model for the CIF that does make it easy to see the effects of covariates is that due to Fine and Gray (1999). They specify a model for the *hazard of the subdistribution* (Gray 1988), formally defined for failure type 1 as

$$\overline{h}_1(t) = \lim_{\delta \to 0} \left\{ \frac{P(t < T \leq t+\delta \text{ and event type 1}) \mid T > t \text{ or } (T \leq t \text{ and not event type 1})}{\delta} \right\}$$

Less formally, think of this hazard as that which generates failure events of interest while keeping subjects who experience competing events "at risk" so that they can be adequately counted as not having any chance of failing. The advantage of modeling the subdistribution hazard, or *subhazard*, is that you can readily calculate the CIF from it;

$$\text{CIF}_1(t) = 1 - \exp\{-\overline{H}_1(t)\}$$

where $\overline{H}_1(t) = \int_0^t \overline{h}_1(t)dt$ is the *cumulative subhazard*.

Competing-risks regression performed in this manner using `stcrreg` is quite similar to Cox regression performed using `stcox`. The model is semiparametric in that the baseline subhazard $\overline{h}_{1,0}(t)$ (that for covariates set to zero) is left unspecified, while the effects of the covariates $\mathbf{x}$ are assumed to be proportional:

$$\overline{h}_1(t|\mathbf{x}) = \overline{h}_{1,0}(t)\exp(\mathbf{x}\boldsymbol{\beta})$$

Estimation with `stcrreg` will produce estimates of $\boldsymbol{\beta}$, or exponentiated coefficients known as *subhazard ratios*. A positive (negative) coefficient means that the effect of increasing that covariate is to increase (decrease) the subhazard and thus increase (decrease) the CIF across the board.

Estimates of the baseline cumulative subhazard and of the baseline CIF are available via `predict` after `stcrreg`; see [ST] **stcrreg postestimation**. Because proportionality holds for cumulative subhazards as well, adjusting the baseline cumulative hazard and baseline CIF for a given set of covariate values is quite easy and, in fact, done automatically for you by `stcurve`; see [ST] **stcurve**.

Using stcrreg

If you have used `stcox` before, `stcrreg` will look very familiar.

▷ Example 1: Cervical cancer study

Pintilie (2006, sec. 1.6.2) describes data from 109 cervical cancer patients that were treated at a cancer center between 1994 and 2000. The patients were treated and then the time in years until relapse or loss to follow-up was recorded. Relapses were recorded as either "local" if cancer relapsed in the pelvis, or "distant" if cancer recurred elsewhere but not in the pelvis. Patients who did not respond to the initial treatment were considered to have relapsed locally after one day.

```
. use http://www.stata-press.com/data/r13/hypoxia
(Hypoxia study)

. describe

Contains data from http://www.stata-press.com/data/r13/hypoxia.dta
  obs:           109                          Hypoxia study
 vars:            16                          7 Apr 2013 09:44
 size:         3,706                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
stnum	int	%8.0g		Patient ID
age	byte	%8.0g		Age (years)
hgb	int	%8.0g		Hemoglobin (g/l)
tumsize	float	%9.0g		Tumor size (cm)
ifp	float	%9.0g		Interstitial fluid pressure (marker, mmHg)
hp5	float	%9.0g		Hypoxia marker (percentage of meas. < 5 mmHg)
pelvicln	str1	%9s		Pelvic node involvement: N=Negative, E=Equivocal, Y=Positive
resp	str2	%9s		Response after treatment: CR=Complete response, NR=No response
pelrec	byte	%9.0g	yesno	Pelvic disease observed
disrec	byte	%9.0g	yesno	Distant disease observed
survtime	float	%9.0g		Time from diagnosis to death or last follow-up time (yrs)
stat	byte	%8.0g		Status at last follow-up: 0=Alive, 1=Dead
dftime	float	%9.0g		Time from diagnosis to first failure or last follow-up (yrs)
dfcens	byte	%8.0g		Censoring variable: 1=Failure, 0=Censored
failtype	byte	%8.0g		Failure type: 1 if pelrec, 2 if disrec & not pelrec, 0 otherwise
pelnode	byte	%8.0g		1 if pelvic nodes negative or equivocal

```
Sorted by:
```

The `dftime` variable records analysis time in years and the `failtype` variable records the type of event observed: 0 for loss to follow-up (censored), 1 for a local relapse, and 2 for a distant relapse. Among the covariates used in the analysis were a hypoxia marker (`hp5`) that measures the degree of oxygenation in the tumor, interstitial fluid pressure (`ifp`), tumor size (`tumsize`), and an indicator of pelvic node involvement (`pelnode == 0` if positive involvement and `pelnode == 1` otherwise). The main goal of the study was to determine whether `ifp` and `hp5` influence the outcome, controlling for the other covariates. Following Pintilie (2006), we focus on `ifp` and not on `hp5`. For more details regarding this study and the process behind the measured data, see Fyles et al. (2002) and Milosevic et al. (2001).

We wish to fit a competing-risks model that treats a local relapse as the event of interest and a distant relapse as the competing event. Although a distant relapse does not strictly prevent a future local relapse, presumably, the treatment protocol changed based on which event was first observed. As such, both events can be treated as competing with one another because the conditions of the study ended once any relapse was observed. Because no deaths occurred before first relapse, death is not considered a competing event in this analysis.

To fit the model, we first `stset` the data and specify that a local relapse, `failtype == 1`, is the event of interest. We then specify to `stcrreg` the covariates and that a distant relapse (`failtype == 2`) is a competing event.

```
. stset dftime, failure(failtype == 1)
  (output omitted)
. stcrreg ifp tumsize pelnode, compete(failtype == 2)

         failure _d:  failtype == 1
   analysis time _t:  dftime

Iteration 0:   log pseudolikelihood = -138.67925
Iteration 1:   log pseudolikelihood = -138.53082
Iteration 2:   log pseudolikelihood =  -138.5308
Iteration 3:   log pseudolikelihood =  -138.5308

Competing-risks regression                      No. of obs       =       109
                                                No. of subjects  =       109
Failure event  : failtype == 1                  No. failed       =        33
Competing event: failtype == 2                  No. competing    =        17
                                                No. censored     =        59

                                                Wald chi2(3)     =     33.21
Log pseudolikelihood =  -138.5308               Prob > chi2      =    0.0000
```

_t	SHR	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
ifp	1.033206	.0178938	1.89	0.059	.9987231	1.068879
tumsize	1.297332	.1271191	2.66	0.008	1.070646	1.572013
pelnode	.4588124	.1972067	-1.81	0.070	.1975931	1.065365

From the above we point out the following:

- When we `stset` the data, distant relapses were set as censored because they are not the event of interest and any standard, noncompeting-risks analysis would want to treat them as censored. `stcrreg` option `compete()` tells Stata which of these "censored" events are actually competing events that require special consideration in a competing-risks regression. Because competing events are not the event of interest, `stcrreg` will issue an error if competing events are not `stset` as censored.
- `stcrreg` lists the event code(s) for the event of interest under "`Failure event(s):`" and the competing event code(s) under "`Competing event(s):`". The syntax for `stset` and `stcrreg` allows you to have multiple codes for both. For competing events, multiple event codes can be devoted entirely to one competing event type, many competing event types, or some combination of both. The methodology behind `stcrreg` extends to more than one competing event type and is concerned only with whether events are competing events, not with their exact type. The focus is on the event of interest.
- We see that out of the 109 patients, 33 experienced a local relapse, 17 experienced a distant relapse, and the remaining 59 were lost to follow-up before any relapse.
- In the column labeled "SHR" are the estimated subhazard ratios, and you interpret these similarly to hazard ratios in Cox regression. Because the estimated subhazard ratio for `ifp` is greater than 1, higher interstitial fluid pressures are associated with higher incidence of local relapses controlling for tumor size, pelvic node involvement, and the fact that distant relapses can also occur. However, this effect is not highly significant.
- To see the estimated coefficients instead of subhazard ratios, use the `noshr` option either when fitting the model or when replaying results.

- Standard errors are listed as "Robust", even though we did not specify any sampling weights, `vce(robust)`, or `vce(cluster` *clustvar*`)`. As mentioned in the previous section, competing-risks regression works by keeping subjects who experience competing events at risk so that they can be adequately counted as having no chance of failing. Doing so requires a form of sample weighting that invalidates the usual model-based standard errors; see *Methods and formulas*. Robust standard errors are conventional in `stcrreg`.
- The output lists a "`log pseudolikelihood`" rather than the standard log likelihood. This is also a consequence of the inherent sample weighting explained in the previous bullet. The log pseudolikelihood is used as a maximization criterion to obtain parameter estimates, but is not representative of the distribution of the data. For this reason, likelihood-ratio (LR) tests (the `lrtest` command) are not valid after `stcrreg`. Use Wald tests (the `test` command) instead.

As mentioned above, you can use the `noshr` option to obtain coefficients instead of subhazard ratios.

```
. stcrreg, noshr
Competing-risks regression                    No. of obs       =        109
                                              No. of subjects  =        109
Failure event  : failtype == 1                No. failed       =         33
Competing event: failtype == 2                No. competing    =         17
                                              No. censored     =         59
                                              Wald chi2(3)     =      33.21
Log pseudolikelihood =  -138.5308             Prob > chi2      =     0.0000
```

_t	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
ifp	.0326664	.0173188	1.89	0.059	-.0012777	.0666105
tumsize	.2603096	.0979851	2.66	0.008	.0682623	.4523568
pelnode	-.7791139	.4298199	-1.81	0.070	-1.621545	.0633176

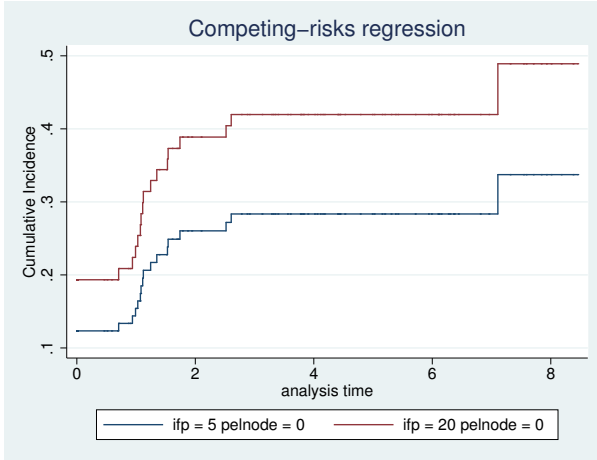
Just as with `stcox`, this model has no constant term. It is absorbed as part of the baseline subhazard, which is not directly estimated.

◁

▷ Example 2: CIF curves after stcrreg

In the above analysis, we stated that with increased interstitial fluid pressure comes an increase in the incidence of local relapses in the presence of possible distant relapses. To demonstrate this visually, we use `stcurve` to compare two CIF curves: one for `ifp == 5` and one for `ifp == 20`. For both curves, we assume positive pelvic node involvement (`pelnode==0`) and tumor size set at the mean over the data.

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



For positive pelvic node involvement and mean tumor size, the probability of local relapse within 2 years is roughly 26% when the interstitial fluid pressure is 5 mmHg and near 40% when this is increased to 20 mmHg. Both probabilities take into account the possibility that a distant relapse could occur instead.

◁

Multiple competing-event types

Competing-risks regression generalizes to the case where more than one type of event competes with the event of interest. If you have such data, after you `stset` the failure event of interest, you can lump together all competing event codes into the `compete()` option of `stcrreg`. It does not matter whether multiple codes represent the same competing-event type, or if they represent multiple types. The results will be the same.

▷ Example 3: UDCA in patients with PBC

Therneau and Grambsch (2000, sec. 8.4.3) analyze data from patients with primary biliary cirrhosis (PBC), a chronic liver disease characterized by progressive destruction of the bile ducts. Data were obtained from 170 patients in a randomized double-blind trial conducted at the Mayo Clinic from 1988 to 1992. The trial was for a new treatment, ursodeoxycholic acid (UDCA; Lindor et al. [1994]).

```
. use http://www.stata-press.com/data/r13/udca
(Randomized trial of UDCA in PBC)
. describe
Contains data from http://www.stata-press.com/data/r13/udca.dta
  obs:           188                          Randomized trial of UDCA in PBC
 vars:             8                          3 Apr 2013 09:37
 size:         5,264                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
id	int	%9.0g		Patient ID
entry	float	%td		Date of enrollment
eventtime	float	%td		Date of first event or loss to follow-up
treat	byte	%9.0g		0=placebo 1=UDCA
stage	byte	%9.0g		histologic stage: 0=stage 1/2 at entry 1=stage 3/4
lbili	float	%9.0g		log(bilirubin value)
etype	float	%9.0g	event	Event type (see notes)
wt	double	%4.2f		Observation weight

```
Sorted by:  id
```

The `etype` variable is coded as any of eight distinct event types (or no event) according to table 1.

Table 1. Event codes for the `etype` variable

Event code	Event type
0	No event (censored)
1	Death
2	Transplant
3	Histologic progression
4	Development of varices
5	Development of ascites
6	Development of encephalopathy
7	Doubling of bilirubin
8	Worsening of symptoms

Cleves (1999) analyzed these data by estimating the cause-specific hazards for each of the eight events. In the version of the data used there, the time at which any adverse event occurred was recorded, but here we record only the time of the first adverse event for each patient. We do so because we wish to perform a competing-risks analysis where we are interested in the time to the first adverse event and the type of that event. The events compete because only one can be first.

We are interested in whether treatment will decrease the incidence of histologic progression (`etype == 3`) as the first adverse outcome, in reference to treatment (`treat`), the logarithm of bilirubin level (`lbili`), and histologic stage at entry (`stage`). Because the patients entered the study at different times (`entry`), when `stset`ting the data we must specify this variable as the origin, or onset of risk.

The competing-risks analysis described above could thus proceed as follows:

```
. stset eventtime, failure(etype == 3) origin(entry)
. stcrreg treat lbili stage, compete(etype == 1 2 4 5 6 7 8)
```

except for one minor complication. Some patients experienced multiple “first events”, and thus ties exist. For example, consider patient 8 who experienced four adverse events at the same time:

```
. list if id == 8
```

	id	entry	eventtime	treat	stage	lbili	etype	wt
8.	8	25may1988	02jul1990	0	1	1.629241	ascites	0.25
9.	8	25may1988	02jul1990	0	1	1.629241	ence	0.25
10.	8	25may1988	02jul1990	0	1	1.629241	bili_2	0.25
11.	8	25may1988	02jul1990	0	1	1.629241	worse	0.25

While most patients are represented by one record each, patients with multiple first events are represented by multiple records. Rather than break ties arbitrarily, we take advantage of how importance weights (`iweights`) are handled by `stcrreg`. Importance weights are treated like frequency weights, but they are allowed to be noninteger. As such, we define the weight variable (`wt`) to equal one for single-record patients and to equal one divided by the number of tied events for multiple-record patients. In this way, each patient contributes a total weight of one observation.

The only further modification we need is to specify `vce(cluster id)` so that our standard errors account for the correlation within multiple records on the same patient.

```
. stset eventtime [iw=wt], failure(etype == 3) origin(entry)
  (output omitted)
. stcrreg treat lbili stage, compete(etype == 1 2 4 5 6 7 8) vce(cluster id)

        failure _d:  etype == 3
  analysis time _t:  (eventtime-origin)
            origin:  time entry
            weight:  [iweight=wt]

Iteration 0:   log pseudolikelihood = -62.158461
Iteration 1:   log pseudolikelihood = -61.671367
Iteration 2:   log pseudolikelihood = -61.669225
Iteration 3:   log pseudolikelihood = -61.669225

Competing-risks regression                      No. of obs       =       170
                                                No. of subjects  =       170
Failure event   : etype == 3                    No. failed       =  12.66667
Competing events: etype == 1 2 4 5 6 7 8        No. competing    =  59.33333
                                                No. censored     =        98

                                                Wald chi2(3)     =      1.89
Log pseudolikelihood = -61.669225               Prob > chi2      =    0.5955
                                  (Std. Err. adjusted for 170 clusters in id)
```

_t	SHR	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
treat	.5785214	.3238038	-0.98	0.328	.1931497	1.732786
lbili	1.012415	.367095	0.03	0.973	.4974143	2.060623
stage	.5537101	.3305371	-0.99	0.322	.1718534	1.78405

In the above, we clustered on `id` but we did not `stset` it as an `id()` variable. That was because we wanted `stcrreg` to treat each observation within patient as its own distinct spell, not as a set of overlapping spells.

Treatment with UDCA seems to decrease the incidence of histologic progression as a first adverse event. However, the effect is not significant, most likely as a result of observing so few failures.

◁

stcrreg as an alternative to stcox

In this section, we demonstrate that you may also use `stcox` to perform a cumulative-incidence analysis, and we compare that approach with one that uses `stcrreg`.

▷ Example 4: HIV and SI as competing events

Geskus (2000) and Putter, Fiocco, and Geskus (2007) analyzed data from 324 homosexual men from the Amsterdam Cohort Studies on HIV infection and AIDS. During the course of infection, the syncytium inducing (SI) HIV phenotype appeared in many of these individuals. The appearance of the SI phenotype worsens prognosis. Thus the time to SI appearance in the absence of an AIDS diagnosis is of interest. In this context, a diagnosis of AIDS acts as a competing event.

```
. use http://www.stata-press.com/data/r13/hiv_si
(HIV and SI as competing risks)
. describe
Contains data from http://www.stata-press.com/data/r13/hiv_si.dta
  obs:           324                          HIV and SI as competing risks
 vars:             4                          3 Apr 2013 13:40
 size:         2,592                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
patnr           int     %8.0g                 ID
time            float   %9.0g                 Years from HIV infection
status          byte    %10.0g     stat       1 = AIDS, 2 = SI, 0 = event-free
ccr5            byte    %9.0g      ccr5       1 if WM (deletion in C-C
                                                chemokine receptor 5 gene)
-------------------------------------------------------------------------------
Sorted by:
```

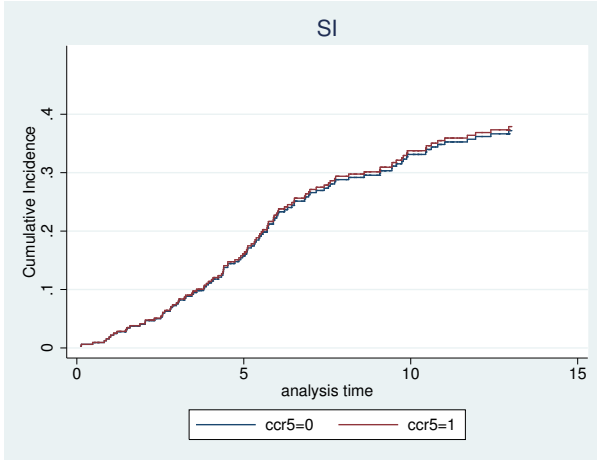
In what follows, we re-create the analysis performed by Putter, Fiocco, and Geskus (2007), treating AIDS and SI as competing events and modeling cumulative incidence in relation to covariate `ccr5`. `ccr5` equals 1 if a specific deletion in the C-C chemokine receptor 5 gene is present and equals zero otherwise (wild type).

We can model the cumulative incidence of SI on `ccr5` directly with `stcrreg`:

```
. stset time, failure(status == 2)        // SI is the event of interest
  (output omitted)
. stcrreg ccr5, compete(status == 1)      // AIDS is the competing event
  (output omitted)
Competing-risks regression                    No. of obs       =        324
                                              No. of subjects  =        324
Failure event  : status == 2                  No. failed       =        107
Competing event: status == 1                  No. competing    =        113
                                              No. censored     =        104
                                              Wald chi2(1)     =       0.01
Log pseudolikelihood = -579.06241             Prob > chi2      =     0.9172
------------------------------------------------------------------------------
             |               Robust
          _t |        SHR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        ccr5 |   1.023865   .2324119     0.10   0.917     .6561827    1.597574
------------------------------------------------------------------------------
```

It seems that this particular genetic mutation has little relation with the incidence of SI, a point we emphasize further with a graph:

```
. stcurve, cif at1(ccr5=0) at2(ccr5=1) title(SI) range(0 13) yscale(range(0 0.5))
```



The above analysis compared SI incidence curves under the assumption that the subhazard for SI, that which generates SI events in the presence of AIDS, was proportional with respect to `ccr5`. Because we modeled the subhazard and not the cause-specific hazard, obtaining estimates of cumulative incidence was straightforward and depended only on the subhazard for SI and not on that for AIDS.

As explained in *The case for competing-risks regression*, the cumulative incidence of SI is a function of both the cause-specific hazard for SI, $h_1(t)$, and that for AIDS, $h_2(t)$, because SI and AIDS are competing events. Suppose for the moment that we are not interested in the incidence of SI in the presence of AIDS, but instead in the biological mechanism that causes SI in general. We can model this mechanism with `stcox` by treating AIDS events as censored.

```
. stcox ccr5
  (output omitted)
Cox regression -- no ties
No. of subjects =          324                 Number of obs    =       324
No. of failures =          107
Time at risk    =  2261.959996
                                               LR chi2(1)       =      1.19
Log likelihood  =   -549.73443                 Prob > chi2      =    0.2748
```

_t	Haz. Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
ccr5	.7755334	.1846031	-1.07	0.286	.4863914	1.23656

Because we initially `stset` our data with SI as the event of interest, AIDS events are treated as censored by `stcox` (but not by `stcrreg`). In any case, the `ccr5` mutation somewhat decreases the risk of SI, but this effect is not significant.

We make the above interpretation with no regard to AIDS as a competing risk because we are interested only in the biological mechanism behind SI. To estimate the cumulative incidence of SI, we first need to make a choice. Either we can pretend a diagnosis of AIDS does not exist as a competing risk and use `stcurve` to plot survivor curves for SI based on the Cox model above, or we can acknowledge AIDS as a competing risk and model that cause-specific hazard also.

We choose the latter. Before fitting the model, however, we need to re-`stset` the data with AIDS as the event of interest.

```
. stset time, failure(status == 1)        // AIDS is the event of interest
  (output omitted)
. stcox ccr5
  (output omitted)
Cox regression -- Breslow method for ties
No. of subjects =          324                     Number of obs    =       324
No. of failures =          113
Time at risk    =  2261.959996
                                                   LR chi2(1)       =     21.98
Log likelihood  =   -555.37301                     Prob > chi2      =    0.0000
```

_t	Haz. Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
ccr5	.2906087	.0892503	-4.02	0.000	.1591812	.530549

Patients with the `ccr5` mutation have a significantly lower risk of AIDS.

We have now modeled both cause-specific hazards separately. Cleves (1999); Lunn and McNeil (1995); and Putter, Fiocco, and Geskus (2007) (among others) describe an approach based on data duplication where both hazards can be modeled simultaneously. Such an approach has the advantage of being able to set the effects of `ccr5` on both hazards as equal and to test that hypothesis. Also, you can model the baseline hazards as proportional rather than entirely distinct. However, for the least parsimonious model with event-specific covariate effects and event-specific baseline hazards, the data duplication method is no different than fitting separate models for each event type, just as we have done above. Because data duplication will reveal no simpler model for these data, we do not describe it further.

We can derive estimates of cumulative incidence for SI based on the above cause-specific hazard models, but the process is a bit more complicated than before. The cumulative incidence of SI (event type 1) in the presence of AIDS (event type 2) is calculated as

$$\widehat{\mathrm{CIF}}_1(t) = \sum_{j:t_j \le t} \widehat{h}_1(t_j)\widehat{S}(t_{j-1})$$

with

$$\widehat{S}(t) = \prod_{j:t_j \le t} \left\{1 - \widehat{h}_1(t_j) - \widehat{h}_2(t_j)\right\}$$

The t_j index the times at which events (of any type) occur, and $\widehat{h}_1(t_j)$ and $\widehat{h}_2(t_j)$ are the cause-specific *hazard contributions* for SI and AIDS respectively. Baseline hazard contributions can be obtained with `predict` after `stcox`, and they can be transformed to hazard contributions for any covariate pattern by multiplying them by the exponentiated linear predictor for that pattern. Hazard contributions represent the increments of the cumulative hazards at each event time. $\widehat{S}(t)$ estimates the probability that you are event free at time t.

We begin by refitting both models and predicting the hazard contributions.

```
. stset time, failure(status == 2)            // SI
  (output omitted)
. stcox ccr5
  (output omitted)
. predict h_si_0, basehc
(217 missing values generated)
. gen h_si_1 = h_si_0*exp(_b[ccr5])
(217 missing values generated)
. stset time, failure(status == 1)            // AIDS
  (output omitted)
. stcox ccr5
  (output omitted)
. predict h_aids_0, basehc
(211 missing values generated)
. gsort _t -_d
. by _t: replace h_aids_0 = . if _n > 1
(1 real change made, 1 to missing)
. gen h_aids_1 = h_aids_0*exp(_b[ccr5])
(212 missing values generated)
```

Variables `h_si_0` and `h_aids_0` hold the baseline hazard contributions, those for `ccr5 == 0`. Variables `h_si_1` and `h_aids_1` hold the hazard contributions for `ccr5 == 1`, and they were obtained by multiplying the baseline contributions by the exponentiated coefficient for `ccr5`. When we ran `stcox` with AIDS as the event of interest, the output indicated that we had tied failure times (the analysis for SI had no ties). As such, this required the extra step of setting any duplicated hazard contributions to missing. As it turned out, this affected only one observation.

Hazard contributions are generated only at times when events are observed and are set to missing otherwise. Because we will be summing and multiplying over event times, we next drop the observations that contribute nothing and then replace missing with zero for those observations that have some hazard contributions missing and some nonmissing.

```
. drop if missing(h_si_0) & missing(h_aids_0)
(105 observations deleted)
. replace h_aids_0 = 0 if missing(h_aids_0)
(107 real changes made)
. replace h_aids_1 = 0 if missing(h_aids_1)
(107 real changes made)
. replace h_si_0 = 0 if missing(h_si_0)
(112 real changes made)
. replace h_si_1 = 0 if missing(h_si_1)
(112 real changes made)
```

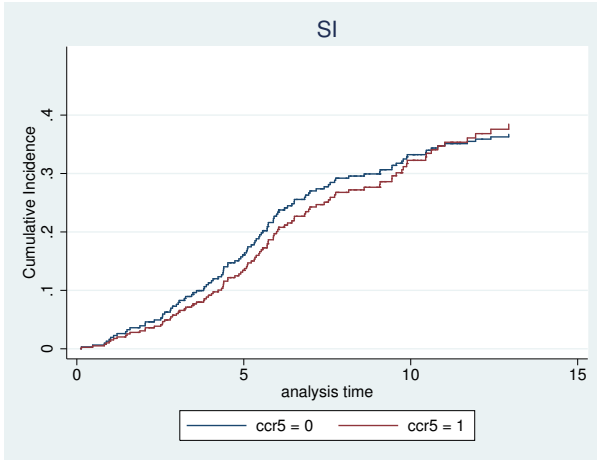
We can now sort by analysis time and calculate the estimated event-free survivor functions. Recall that you can express a product as an exponentiated sum of logarithms, which allows us to take advantage of Stata's `sum()` function for obtaining running sums.

```
. sort _t
. gen S_0 = exp(sum(log(1- h_aids_0 - h_si_0)))
. gen S_1 = exp(sum(log(1- h_aids_1 - h_si_1)))
```

Finally, we calculate the estimated CIFs and graph:

```
. gen cif_si_0 = sum(S_0[_n-1]*h_si_0)
. label var cif_si_0 "ccr5 = 0"
```

```
. gen cif_si_1 = sum(S_1[_n-1]*h_si_1)
. label var cif_si_1 "ccr5 = 1"
. twoway line cif_si* _t if _t<13, connect(J J) sort yscale(range(0 0.5))
> title(SI) ytitle(Cumulative Incidence) xtitle(analysis time)
```



This model formulation shows `ccr5` to have more of an effect on the incidence of SI, although the effect is still small. Note that under this formulation, the effect of `ccr5` is not constrained to be overall increasing or overall decreasing. In fact, when $t > 11$ years or so, those with the `ccr5` mutation actually have an increased SI incidence. That is due to time-accumulated reduced competition from AIDS, the risk of which is significantly lower when the `ccr5` mutation is present.

Putter, Fiocco, and Geskus (2007) also performed the same analysis using AIDS as the event of interest, something we leave to you as an exercise.

◁

We have described two different modeling approaches for estimating the cumulative incidence of SI. Although you may prefer the `stcrreg` approach because it is much simpler, that does not mean it is a better model than the one based on `stcox`. The better model is the one whose assumptions more closely fit the data. The `stcrreg` model assumes that the effect of `ccr5` is proportional on the subhazard for SI. The `stcox` model assumes proportionality on the cause-specific hazards for both SI and AIDS. Because our analysis uses only one binary covariate, we can compare both models with a nonparametric estimator of the CIF to see which fits the data more closely; see [ST] **stcrreg postestimation**.

Multiple records per subject

`stcrreg` can be used with data where you have multiple records per subject, as long as 1) you `stset` an ID variable that identifies the subjects and 2) you carefully consider the role played by time-varying covariates in subjects who fail because of competing events. We explain both issues below.

Stata's `st` suite of commands allows for multiple records per subject. Having multiple records allows you to record gaps in subjects' histories and to keep track of time-varying covariates. If you have multiple records per subject, you identify which records belong to which subjects by specifying an ID variable to `stset` option `id()`.

Consider the sample data listed below:

```
. list if id == 18
```

	id	_t0	_t	_d	x
1.	18	3	5	0	5.1
2.	18	5	8	0	7.8
3.	18	11	12	0	6.7
4.	18	12	20	1	8.9

These data reflect the following:

- Subject 18 first became at risk at analysis time 3 (delayed entry) with covariate value `x` equal to 5.1.
- At time 5, subject 18's `x` value changed to 7.8.
- Subject 18 left the study at time 8 only to return at time 11 (gap), with `x` equal to 6.7 at that time.
- At time 12, `x` changed to 8.9.
- Subject 18 failed at time 20 with `x` equal to 8.9 at that time.

An analysis of these data with Cox regression using `stcox` is capable of processing all this information. Intermittent records are treated as censored (`_d==0`), and either failure or censoring occurs on the last record (here failure with `_d==1`). When subjects are not under observation, they are simply not considered at risk of failure. Time-varying covariates are also processed correctly. For example, if some other subject failed at time 7, then the risk calculations would count subject 18 at risk with `x` equal to 7.8 at that time.

`stcox` will give the same results for the above data whether or not you `stset` the ID variable, `id`. Whether you treat the above data as four distinct subjects (three censored and one failed) or as one subject with a four-record history is immaterial. The only difference you may encounter concerns robust and replication-based standard errors, in which case if you `stset` an ID variable, then `stcox` will automatically cluster on this variable.

Such a distinction, however, is of vital importance to `stcrreg`. While `stcox` is concerned only about detecting one type of failure, `stcrreg` relies on precise accounting of the number of subjects who fail because of the event of interest, those who fail because of competing events, and those who are censored. In particular, the weighting mechanism behind `stcrreg` depends on an accurate estimate of the probability a subject will be censored; see *Methods and formulas*. As such, it makes a difference whether you want to treat the above as four distinct subjects or as one subject. If you have multiple records per subject, you must `stset` your ID variable before using `stcrreg`. When counting the number failed, number competing, and number censored, `stcrreg` only considers what happened at the end of a subject's history. Intermittent records are treated simply as temporary entries to and exits from the analysis, and the exits are not counted as censored in the strict sense.

Furthermore, when using `stcrreg` with covariates that change over multiple records (time-varying covariates), you need to carefully consider what happens when subjects experience competing failures. For the above sample data, subject 18 failed because of the event interest (`_d==1`). Consider, however, what would have happened had this subject failed because of a competing event instead. Competing-risks regression keeps such subjects "at risk" of failure from the event of interest even after they fail from competing events; see *Methods and formulas*. Because these subjects will be used in future risk calculations for which they have no data, `stcrreg` will use the last available covariate values for these calculations. For the above example, if subject 18 experiences a competing event at time 20, then the last available value of `x`, 8.9, will be used in all subsequent risk calculations. If the last

available values are as good a guess as any as to what future values would have been—for example, a binary covariate recording pretransplant versus posttransplant status—then this is not an issue. If, however, you have reason to believe that a subject's covariates would have been much different had the subject remained under observation, then the results from `stcrreg` could be biased.

▷ Example 5: Hospital-acquired pneumonia

Consider the following simulated data from a competing-risks analysis studying the effects of pneumonia.

```
. use http://www.stata-press.com/data/r13/pneumonia, clear
(Hospital-acquired pneumonia)
. describe
Contains data from http://www.stata-press.com/data/r13/pneumonia.dta
  obs:           957                          Hospital-acquired pneumonia
 vars:             7                          7 Apr 2013 15:35
 size:         8,613
```

variable name	storage type	display format	value label	variable label
id	int	%9.0g		Patient id
age	byte	%9.0g		Age at admission
ndays	int	%9.0g		Days in ICU
died	byte	%9.0g		1 if died
censored	byte	%9.0g		1 if alive and in ICU at the end of the study
discharged	byte	%9.0g		1 if discharged
pneumonia	byte	%9.0g		1 if pneumonia

```
Sorted by:  id
```

The above data are for 855 ICU patients. One hundred twenty-three patients contracted pneumonia, of which 21 did before admission and 102 during their stay. Those patients who contracted pneumonia during their stay are represented by two records with the time-varying covariate `pneumonia` recording the change in status.

We perform a competing-risks regression for the cumulative incidence of death during ICU stay with `age` and `pneumonia` as covariates. We also treat hospital discharge as a competing event.

```
. stset ndays, id(id) failure(died)
  (output omitted)
. stcrr age pneumonia, compete(discharged) noshow nolog
Competing-risks regression                      No. of obs       =       957
                                                No. of subjects  =       855
Failure events  : died nonzero, nonmissing      No. failed       =       178
Competing events: discharged nonzero, nonmissing No. competing   =       641
                                                No. censored     =        36
                                                Wald chi2(2)     =    121.21
Log pseudolikelihood = -1128.6096               Prob > chi2      =    0.0000
                                (Std. Err. adjusted for 855 clusters in id)
```

_t	SHR	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
age	1.021612	.0076443	2.86	0.004	1.006739	1.036705
pneumonia	5.587052	.9641271	9.97	0.000	3.983782	7.835558

Both increased age and contracting pneumonia are associated with an increased incidence of death in the ICU.

◁

Option tvc() and testing the proportional-subhazards assumption

In the previous section, we considered data with multiple records per subject. Such data makes it possible to record discrete time-varying covariates, those whose values change at discrete points in time. Each change is captured by a new record.

Consider instead what happens when you have covariates that vary continuously with respect to time. Competing-risks regression assumes the following relationship between subhazard and baseline subhazard

$$\overline{h}_1(t) = \overline{h}_{1,0}(t)\exp(\beta_1 x_1 + \cdots + \beta_k x_k)$$

where $\overline{h}_{1,0}(t)$ is the baseline subhazard function. For most purposes, this model is sufficient, but sometimes we may wish to introduce variables of the form $z_i(t) = z_i g(t)$, which vary continuously with time so that

$$\overline{h}_1(t) = \overline{h}_{1,0}(t)\exp\{\beta_1 x_1 + \cdots + \beta_k x_k + g(t)(\gamma_1 z_1 + \cdots + \gamma_m z_m)\} \tag{1}$$

where $(z_1, \ldots, z_m)$ are the time-varying covariates. Fitting this model has the net effect of estimating the regression coefficient, γ_i, for the covariate $g(t)z_i$, which is a function of analysis time.

The time-varying covariates $(z_1, \ldots, z_m)$ are specified using the `tvc(`*tvarlist*`)` option, and $g(t)$ is specified using the `texp(`*exp*`)` option, where t in $g(t)$ is analysis time. For example, if we want $g(t) = \log(t)$, we would use `texp(log(_t))` because `_t` stores the analysis time once the data are `stset`.

When subjects fail because of competing events, covariate values for these subjects continue to be used in subsequent risk calculations; see the previous section for details. When this occurs, any time-varying covariates specified using `tvc()` will continue to respect their time interactions even after these subjects fail. Because such behavior is unlikely to reflect any real data situation, we do not recommend using `tvc()` for this purpose.

We do, however, recommend using `tvc()` to model *time-varying coefficients*, because these can be used to test the proportionality assumption behind competing-risks regression. Consider a version of (1) that contains only one fixed covariate, x_1, and sets $z_1 = x_1$:

$$\overline{h}_1(t) = \overline{h}_{1,0}(t)\exp\left[\{\beta_1 + \gamma_1 g(t)\}\, x_1\right]$$

Given this new arrangement, we consider that $\beta_1 + \gamma_1 g(t)$ is a (possibly) time-varying coefficient on the covariate x_1, for some specified function of time $g(t)$. The coefficient has a time-invariant component β_1, with γ_1 determining the magnitude of the time-dependent deviations from β_1. As such, a test of $\gamma_1 = 0$ is a test of time invariance for the coefficient on x_1.

Confirming that a coefficient is time invariant is one way of testing the proportional-subhazards assumption. Proportional subhazards implies that the relative subhazard (that is, β) is fixed over time, and this assumption would be violated if a time interaction proved significant.

Example 6: Testing proportionality of subhazards

Returning to our cervical cancer study (example 1), we now include time interactions on all three covariates as a way of testing the proportional-subhazards assumption for each:

```
. use http://www.stata-press.com/data/r13/hypoxia
(Hypoxia study)
. stset dftime, failure(failtype == 1)
  (output omitted)
. stcrreg ifp tumsize pelnode, compete(failtype == 2) tvc(ifp tumsize pelnode)
> noshr
  (output omitted)
Competing-risks regression                    No. of obs       =        109
                                              No. of subjects  =        109
Failure event  : failtype == 1                No. failed       =         33
Competing event: failtype == 2                No. competing    =         17
                                              No. censored     =         59
                                              Wald chi2(6)     =      44.93
Log pseudolikelihood =    -136.79             Prob > chi2      =     0.0000
```

_t	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
main						
ifp	.0262093	.0174458	1.50	0.133	-.0079838	.0604025
tumsize	.37897	.1096628	3.46	0.001	.1640348	.5939052
pelnode	-.766362	.473674	-1.62	0.106	-1.694746	.162022
tvc						
ifp	.0055901	.0081809	0.68	0.494	-.0104441	.0216243
tumsize	-.1415204	.0908955	-1.56	0.119	-.3196722	.0366314
pelnode	.0610457	.5676173	0.11	0.914	-1.051464	1.173555

```
Note: variables in tvc equation interacted with _t
```

We used the default function of time $g(t) = t$, although we could have specified otherwise with the `texp()` option. After looking at the significance levels in the equation labeled "`tvc`", we find no indication that the proportionality assumption has been violated.

◁

When you use `tvc()` in this manner, there is no issue of postfailure covariate values for subjects who fail from competing events. The covariate values are assumed constant—the *coefficients* change with time.

Stored results

stcrreg stores the following in e():

Scalars

e(N)	number of observations
e(N_sub)	number of subjects
e(N_fail)	number of failures
e(N_compete)	number of competing events
e(N_censor)	number of censored subjects
e(k)	number of parameters
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log pseudolikelihood
e(N_clust)	number of clusters
e(chi2)	χ^2
e(p)	significance
e(rank)	rank of e(V)
e(fmult)	1 if $>$ 1 failure events, 0 otherwise
e(crmult)	1 if $>$ 1 competing events, 0 otherwise
e(fnz)	1 if nonzero indicates failure, 0 otherwise
e(crnz)	1 if nonzero indicates competing, 0 otherwise
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	stcrreg
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(mainvars)	variables in main equation
e(tvc)	time-varying covariates
e(texp)	function used for time-varying covariates
e(fevent)	failure event(s) in estimation output
e(crevent)	competing event(s) in estimation output
e(compete)	competing event(s) as typed
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(offset1)	offset
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(Cns)	constraints matrix
e(ilog)	iteration log
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample

Methods and formulas

In what follows, we assume single-record data and time-invariant covariates or coefficients. Extensions to both multiple-record data and continuous time-varying covariates are achieved by treating the mechanisms that generate censorings, competing events, and failure events of interest as counting processes; see Fine and Gray (1999) and Andersen et al. (1993) for further details.

Let $\mathbf{x}_i$ be the row vector of m covariates for the time interval $(t_{0i}, t_i\,]$ for the ith observation in the dataset ($i = 1, \ldots, n$). `stcrreg` obtains parameter estimates $\widehat{\boldsymbol{\beta}}$ by maximizing the log-pseudolikelihood function

$$\log L = \sum_{i=1}^{n} \delta_i w_i \left[\mathbf{x}_i\boldsymbol{\beta} + \text{offset}_i - \log \left\{ \sum_{j \in R_i} w_j \pi_{ji} \exp(\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j) \right\} \right]$$

where δ_i indicates a failure of interest for observation i and R_i is the set of observations, j, that are at risk at time t_i (that is, all j such that $t_{0j} < t_i \leq t_j$). w_i and offset_i are the usual observation weights and linear offsets, if specified.

The log likelihood given above is identical to that for standard Cox regression (Breslow method for ties) with the exception of the weights π_{ji}. These weights are used to keep subjects who have failed because of competing events in subsequent risk sets and to decrease their weight over time as their likelihood of being otherwise censored increases.

Formally, extend R_i above not only to include those at risk of failure at time t_i, but also to include those subjects already having experienced a competing-risks event. Also, define

$$\pi_{ji} = \frac{\widehat{S}_c(t_i)}{\widehat{S}_c\{\min(t_j, t_i)\}}$$

if subject j experiences a competing event; $\pi_{ji} = 1$ otherwise. $\widehat{S}_c(t)$ is the Kaplan–Meier estimate of the survivor function for the censoring distribution—that which treats censorings as the events of interest—evaluated at time t, and t_j is the time at which subject j experienced his or her competing-failure event. As a matter of convention, $\widehat{S}_c(t)$ is treated as the probability of being censored up to *but not including* time t.

Because of the sample weighting inherent to this estimator, the standard Hessian-based estimate of variance is not statistically appropriate and is thus rejected in favor of a robust, sandwich-type estimator, as derived by Fine and Gray (1999).

Define $z_i = \mathbf{x}_i\widehat{\boldsymbol{\beta}} + \text{offset}_i$. (Pseudo)likelihood scores are given by

$$\widehat{\mathbf{u}}_i = \widehat{\boldsymbol{\eta}}_i + \widehat{\boldsymbol{\psi}}_i$$

where $\widehat{\boldsymbol{\eta}}_i = (\widehat{\eta}_{1i}, \ldots, \widehat{\eta}_{mi})'$, and

$$\widehat{\eta}_{ki} = \delta_i \left(x_{ki} - a_{ki}\right) - \exp(z_i) \sum_{j:t_{0i}<t_j\leq t_i} \frac{\delta_j w_j \pi_{ij}(x_{ki} - a_{kj})}{\sum_{\ell\in R_j} w_\ell \pi_{\ell j} \exp(z_\ell)}$$

for

$$a_{ki} = \frac{\sum_{\ell\in R_i} w_\ell \pi_{\ell i} x_{k\ell} \exp(z_\ell)}{\sum_{\ell\in R_i} w_\ell \pi_{\ell i} \exp(z_\ell)}$$

The $\widehat{\boldsymbol{\psi}}_i$ are variance contributions due to data estimation of the weights π_{ji}, with

$$\widehat{\boldsymbol{\psi}}_i = \frac{\gamma_i \widehat{\mathbf{q}}(t_i)}{r(t_i)} - \sum_{j:t_{0i}<t_j\leq t_i} \frac{\gamma_j \widehat{h}_c(t_j)\widehat{\mathbf{q}}(t_j)}{r(t_j)}$$

γ_i indicates censoring for observation i, $r(t)$ is the number at risk of failure (or censoring) at time t,

$$\widehat{h}_c(t) = \frac{\sum_{i=1}^n \gamma_i I(t_i = t)}{r(t)}$$

and the kth component of $\widehat{\mathbf{q}}(t)$ is

$$\widehat{q}_k(t) = \sum_{i\in C(t)} w_i \exp(z_i) \sum_{j:t_{0i}<t_j\leq t_i} \frac{\delta_j w_j \pi_{ij}(x_{ki} - a_{kj})}{\sum_{\ell\in R_j} w_\ell \pi_{\ell j} \exp(z_\ell)} I(t_j \geq t)$$

where $C(t)$ is the set of observations that experienced a competing event prior to time t.

By default, `stcrreg` calculates the Huber/White/sandwich estimator of the variance and calculates its clustered version if either the `vce(cluster` *clustvar*`)` option is specified or an ID variable has been `stset`. See *Maximum likelihood estimators* and *Methods and formulas* in [P] **_robust** for details on how the pseudolikelihood scores defined above are used to calculate this variance estimator.

Acknowledgment

We thank Jason Fine of the Gillings School of Global Public Health at the University of North Carolina, Chapel Hill, for answering our technical questions.

References

Andersen, P. K., Ø. Borgan, R. D. Gill, and N. Keiding. 1993. *Statistical Models Based on Counting Processes*. New York: Springer.

Beyersman, J., A. Latouche, A. Buchholz, and M. Schumacher. 2009. Simulating competing risks data in survival analysis. *Statistics in Medicine* 28: 956–971.

Beyersman, J., and M. Schumacher. 2008. Time-dependent covariates in the proportional subdistribution hazards model for competing risks. *Biostatistics* 9: 765–776.

Cleves, M. A. 1999. ssa13: Analysis of multiple failure-time data with Stata. *Stata Technical Bulletin* 49: 30–39. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 338–349. College Station, TX: Stata Press.

Cleves, M. A., W. W. Gould, R. G. Gutierrez, and Y. V. Marchenko. 2010. *An Introduction to Survival Analysis Using Stata*. 3rd ed. College Station, TX: Stata Press.

Coviello, V., and M. M. Boggess. 2004. Cumulative incidence estimation in the presence of competing risks. *Stata Journal* 4: 103–112.

Cox, D. R. 1972. Regression models and life-tables (with discussion). *Journal of the Royal Statistical Society, Series B* 34: 187–220.

Crowder, M. J. 2001. *Classical Competing Risks*. Boca Raton, FL: Chapman & Hall/CRC.

Fine, J. P., and R. J. Gray. 1999. A proportional hazards model for the subdistribution of a competing risk. *Journal of the American Statistical Association* 94: 496–509.

Fyles, A., M. Milosevic, D. Hedley, M. Pintilie, W. Levin, L. Manchul, and R. P. Hill. 2002. Tumor hypoxia has independent predictor impact only in patients with node-negative cervix cancer. *Journal of Clinical Oncology* 20: 680–687.

Gail, M. H. 1975. A review and critique of some models used in competing risk analysis. *Biometrics* 31: 209–222.

Geskus, R. B. 2000. On the inclusion of prevalent cases in HIV/AIDS natural history studies through a marker-based estimate of time since seroconversion. *Statistics in Medicine* 19: 1753–1769.

Gichangi, A., and W. Vach. 2005. The analysis of competing risks data: A guided tour. Preprint series, Department of Statistics, University of Southern Denmark. http://www.stat.sdu.dk/publications/preprints/pp009/Anthony%20Gichangi%20Competing%20Risk%20Tutorial.pdf.

Gooley, T. A., W. Leisenring, J. Crowley, and B. E. Storer. 1999. Estimation of failure probabilities in the presence of competing risks: New representations of old estimators. *Statistics in Medicine* 18: 695–706.

Gray, R. J. 1988. A class of k-sample tests for comparing the cumulative incidence of a competing risk. *Annals of Statistics* 16: 1141–1154.

Hinchliffe, S. R., and P. C. Lambert. 2013. Extending the flexible parametric survival model for competing risks. *Stata Journal* 13: 344–355.

Hinchliffe, S. R., D. A. Scott, and P. C. Lambert. 2013. Flexible parametric illness-death models. *Stata Journal* 13: 759–775.

Kaplan, E. L., and P. Meier. 1958. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association* 53: 457–481.

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

Lambert, P. C. 2007. Modeling of the cure fraction in survival studies. *Stata Journal* 7: 351–375.

Lin, D. Y., and L. J. Wei. 1989. The robust inference for the Cox proportional hazards model. *Journal of the American Statistical Association* 84: 1074–1078.

Lindor, K. D., E. R. Dickson, W. P. Baldus, R. A. Jorgensen, J. Ludwig, P. A. Murtaugh, J. M. Harrison, R. H. Wiesner, M. L. Anderson, S. M. Lange, G. LeSage, S. S. Rossi, and A. F. Hofman. 1994. Ursodeoxycholic acid in the treatment of primary biliary cirrhosis. *Gastroenterology* 106: 1284–1290.

Lunn, M., and D. McNeil. 1995. Applying Cox regression to competing risks. *Biometrics* 51: 524–532.

Marubini, E., and M. G. Valsecchi. 1997. *Analysing Survival Data from Clinical Trials and Observational Studies*. Chichester, UK: Wiley.

Milosevic, M., A. Fyles, D. Hedley, M. Pintilie, W. Levin, L. Manchul, and R. P. Hill. 2001. Interstitial fluid pressure predicts survival in patients with cervix cancer independent of clinical prognostic factors and tumor oxygen measurements. *Cancer Research* 61: 6400–6405.

Pepe, M. S., and M. Mori. 1993. Kaplan–Meier, marginal or conditional probability curves in summarizing competing risks failure time data? *Statistics in Medicine* 12: 737–751.

Pintilie, M. 2006. *Competing Risks: A Practical Perspective*. Chichester, UK: Wiley.

——. 2007. Analysing and interpreting competing risk data. *Statistics in Medicine* 26: 1360–1367.

Putter, H., M. Fiocco, and R. B. Geskus. 2007. Tutorial in biostatistics: Competing risks and multi-state models. *Statistics in Medicine* 26: 2389–2430.

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

Tsiatis, A. A. 1975. A nonidentifiability aspect of the problem of competing risks. *Proceedings of the National Academy of Sciences* 72: 20–22.

Also see

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

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

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

[ST] **stcox PH-assumption tests** — Tests of proportional-hazards assumption

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

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

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

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

[MI] **estimation** — Estimation commands for use with mi estimate

[U] 20 Estimation and postestimation commands