The Stata Journal (2016)
16, Number 4, pp. 989–1012

strcs: A command for fitting flexible parametric survival models on the log-hazard scale

Hannah Bower
Department of Medical Epidemiology and Biostatistics
Karolinska Institutet
Stockholm, Sweden
hannah.bower@ki.se

Michael J. Crowther
Department of Medical Epidemiology and Biostatistics
Karolinska Institutet
Stockholm, Sweden
and
Department of Health Sciences
University of Leicester
Leicester, UK
michael.crowther@leicester.ac.uk

Paul C. Lambert
Department of Medical Epidemiology and Biostatistics
Karolinska Institutet
Stockholm, Sweden
and
Department of Health Sciences
University of Leicester
Leicester, UK
paul.lambert@leicester.ac.uk

Abstract. In this article, we describe `strcs`, a user-written command for fitting flexible parametric survival models on the log-hazard scale. `strcs` is an extension of the user-written `stgenreg` command (Crowther and Lambert, 2013b, *Journal of Statistical Software* 53(12): 1–17), which fits general parametric models with user-defined hazard functions using numerical integration. `strcs` implements a two-step method that incorporates both analytical and numerical integration to estimate the cumulative hazard function required for the log-likelihood function. This method improves the accuracy of the fully numeric estimation implemented in `stgenreg`. Time-dependent effects can be incorporated, and excess mortality models can be fit by using the available options. We also describe some of the extensive postestimation commands that are easily implemented after fitting an `strcs` model.

Keywords: st0462, strcs, strcs postestimation, flexible parametric survival model, log-hazard scale

st0462

1 Introduction

The Cox model is the most popular method implemented to model survival data because it requires no assumptions of the shape of the baseline hazard (Cox 1972). Although the Cox model is an extremely useful tool for estimating relative effects, parametric models remain popular to obtain estimates of both absolute and relative effects. Many parametric models can be used to model hazard functions, including Poisson models using splines (Carstensen 2007) or using fractional polynomials (Lambert et al. 2005) and using the generalized F distribution (Cox 2008). Here we focus on the flexible parametric survival models of Royston and Parmar (Royston and Parmar 2002; Royston and Lambert 2011). These models are becoming more popular because they can capture simple and complex hazard functions that standard parametric models may struggle to capture. These flexible models use restricted cubic splines (Durrleman and Simon 1989) to model some transformation of the survival function, usually the log cumulative-hazard function. Fitting flexible parametric survival models on the log cumulative-hazard scale is easily implemented in Stata using the `stpm2` command (Lambert and Royston 2009). Flexible parametric survival models on the log cumulative-hazard scale have been shown to accurately capture a variety of complex hazard functions and to estimate almost identical hazard ratios as the Cox model provided that enough knots are specified for the spline function (Rutherford, Crowther, and Lambert 2015). They have also been shown to accurately capture time-dependent effects (Bower et al. 2015). Flexible parametric survival models have been extended in a variety of settings, including relative survival (Nelson et al. 2007), when modeling cure proportions (Andersson et al. 2011) and when incorporating random effects (Crowther, Look, and Riley 2014).

One can also implement flexible parametric survival models on the log-hazard scale. However, modeling on this scale requires numerical integration when complex hazards, such as splines, are used to maximize the likelihood. General models on the log-hazard scale can be implemented in Stata using the user-written `stgenreg` command (Crowther and Lambert 2013b), which fits parametric survival models for user-defined hazard functions. `stgenreg` implements a fully numeric approach to numerical integration. We present the `strcs` command, an extension of the `stgenreg` command that fits flexible parametric survival models on the log-hazard scale using a combination of analytical integration and numerical integration techniques to increase the accuracy and to reduce the number of nodes required in numerical integration (Crowther and Lambert 2014). The `strcs` command is also more user friendly and has some additional prediction tools. Here we describe the `strcs` command and its extensive postestimation predictions and illustrate their use.

2 Flexible parametric survival models

A flexible parametric survival model with time-dependent effects of covariates $\mathbf{x}$ on the log-hazard scale can be written as

$$\ln\{h(t;\mathbf{x})\} = s\left\{f(x);\boldsymbol{\gamma}_0\right\} + \mathbf{x}\boldsymbol{\beta} + \sum_{d=1}^{D} s\left\{f(x);\boldsymbol{\gamma}_d\right\}\mathbf{x}_d \qquad (1)$$

where $s\{f(x);\boldsymbol{\gamma}_0\}$ represents the restricted cubic spline function, D is the number of time-dependent effects, $s\{f(x);\boldsymbol{\gamma}_d\}$ is the spline function for the dth time-dependent effect, and $f(x) = t$ or $\ln(t)$.

Restricted cubic splines are used within flexible parametric models, usually to model the log cumulative hazard, although they can be used to model other transformations, such as the log cumulative odds (Royston and Parmar 2002; Royston and Lambert 2011). Restricted cubic splines are piecewise cubic functions joined at positions called knots. The overall function is forced to be smooth by forcing the first and second derivatives to be continuous at the knots and constraining the function to be linear before the first and after the last knot. The complexity of these functions is determined by the user-defined degrees of freedom, which is equal to the number of knots minus one. Knot positions can be user defined or chosen to be positioned at equally spaced percentiles of the observed event-time distribution.

A restricted cubic spline function, $s(a;\gamma_0)$, with $a = t$ or $\ln(t)$ and knots $k_1,\ldots,k_K$ is defined as

$$s(a;\gamma_0) = \gamma_{00} + \sum_{l=1}^{K-1} \gamma_{0l} v_l(a)$$

where the lth basis function $v_l(a)$ is defined as

$$v_l(a) = \begin{cases} a, & \text{for } l = 1 \\ (a-k_l)_+^3 - \lambda_l(a-k_1)_+^3 - (1-\lambda_l)(a-k_K)_+^3, & \text{for } l = 2,\ldots,K-1 \end{cases}$$

and where k_1 and k_K are the boundary knots, $\lambda_l = (k_K - k_l)/(k_K - k_1)$, and $u_+ = u$ if $u > 0$ and $u_+ = 0$ if $u \leq 0$. Splines can often be highly correlated; to avoid this and any computational problems that may occur because of this correlation, one can orthogonalize splines.

Restricted cubic splines within flexible parametric survival models allow both simple and complex hazard functions to be captured in situations where standard parametric models may struggle to do so (Rutherford, Crowther, and Lambert 2015). Because they model the baseline function, various postestimation predictions can be calculated, such as time-dependent hazard ratios and differences in hazards. Time-dependent effects, or nonproportional hazards, can also be incorporated easily via introducing an interaction between the covariate and a spline function. The complexity of the deviation from the baseline hazard in the time-dependent effect can also be chosen by the user by specifying separate degrees of freedom for this effect.

3 Maximum likelihood estimation

Flexible parametric survival models are fit using maximum likelihood estimation; the `ml` command is used in Stata. Consider a sample of n individuals where t_i is the exit time and d_i is the event indicator for the ith individual. Then, the log-likelihood contribution for the ith individual is

$$\log l_i = d_i \log\{h(t_i)\} + \log\{S(t_i)\} \tag{2}$$

where $h(t_i)$ is the hazard function and $S(t_i)$ is the survival function evaluated at the time of death or censoring t_i.

Thus, when one fits a flexible parametric survival model, the hazard $h(t_i)$ and the survival $S(t_i)$ are required. The survival can be written as a function of the cumulative hazard function, $H(t_i)$,

$$S(t_i) = \exp[-\{H(t_i)\}]$$

and the cumulative hazard function is the integral of the hazard function:

$$H(t_i) = \int_0^{t_i} h(u_i) du$$

Consider the flexible parametric survival model in (1), and let $\phi(t)$ equal the right-hand side of the equation. Then, we have that the hazard function

$$h(t) = \exp\{\phi(t)\}$$

The survival function can be written in the following way:

$$S(t) = \exp\left[-\int_0^{t_i} \exp\{\phi(t)\} du\right]$$

Thus, to obtain the survival function, which is part of the likelihood function, one must integrate the hazard function. In the context of flexible parametric survival models, the hazard function is a complex spline function that cannot be integrated analytically. Thus, numerical integration techniques are required.

4 Numerical integration in strcs

As described in the previous section, numerical integration techniques are required when fitting flexible parametric survival models on the log-hazard scale to maximize the likelihood function. However, numerical integration over the whole function is not necessary. The restricted cubic spline function is analytically integrable before the first knot and after the last knot because of the constraints imposed on the function at these two intervals. Even though there is often little time before the first knot, this is where numerical integration is the most inaccurate when modeling log transformed time. In comparison to modeling flexible parametric survival models on the log-hazard scale using the `stgenreg` command, performing the integration in two steps rather than over the whole function benefits from increased computational efficiency and more accurate integration.

4.1 The two-step integration approach

`strcs` uses a two-step integration approach that combines both analytical and numerical integration of the hazard function.

The cumulative hazard function can be written as the sum of three components,

$$H(t_i) = H_1 + H_2 + H_3$$

where

- H_1 is the cumulative hazard function before the first knot, k_1;
- H_2 is the cumulative hazard function between k_1 and the last knot k_K; and
- H_3 is the cumulative hazard function after k_K.

Because of the constraints imposed on restricted cubic splines, H_1 and H_3 can be calculated analytically from the corresponding hazard function; thus only numerical integration needs to be applied to obtain H_2. The number of components included in the cumulative hazard function for a particular observation depends on the value of the observed survival time. For example, if the observed survival time t_i is after the first knot but before the last knot, then the cumulative hazard function will contain only H_1 and H_2, where H_2 will correspond to integration between the time at the first knot and t_i; thus integration of the hazard function corresponding to H_3 is not required. See Crowther and Lambert (2014) for further information.

4.2 Gaussian quadrature

Gaussian quadrature is a method of numerical integration that converts an integral into a weighted summation over a set of predefined points known as nodes,

$$\int g(x)dx \approx \sum_{j=1}^{m} w_j g(x_j)$$

where m is the number of nodes and $g(x)$ can be approximated by a polynomial function. The integral can instead be estimated between a and b by the following formula:

$$\int_a^b g(x)dx \approx \frac{b-a}{2} \sum_{j=1}^{m} w_j g\left(\frac{b-a}{2}x_j + \frac{a+b}{2}\right)$$

We are interested in numerically integrating the hazard function $h(x)$ between the time at the first knot t_{k_1} and at the last knot t_{k_K}; that is, we wish to obtain part H_2 of the cumulative hazard function. The above equation then becomes

$$\int_{t_{k_1}}^{\min(t_i,t_{k_K})} h(x)dx \approx \frac{\min(t_i,t_{k_K})-t_{k_1}}{2}\sum_{j=1}^{m} w_j h \left\{\frac{\min(t_i,t_{k_K})-t_{k_1}}{2}x_j + \frac{t_{k_1}+\min(t_i,t_{k_K})}{2}\right\}$$

`strcs` implements Gauss–Legendre quadrature. The accuracy of the numerical integration depends on the number of nodes m specified; previous research has shown that 30 nodes are sufficient in most circumstances (Crowther and Lambert 2013a).

5 Excess mortality models

Excess mortality models, or relative survival models, can also be implemented in `strcs` by modeling the log excess-hazard function. Excess mortality, or relative survival, is a popular method used in population-based cancer studies. Using cause of death data can be problematic because these often ignore treatment-related deaths, and the recording of deaths can be unreliable. To avoid these problems, one can implement excess mortality models. These aim to capture the disease-related mortality by modeling the difference between the all-cause mortality in the diseased population and the all-cause mortality in the nondiseased population. Thus the total mortality rate $h_i(t)$ can be written as a function of the expected mortality rate $h_i^*(t)$ and the excess mortality rate associated with disease, $\lambda_i(t)$,

$$h_i(t) = h_i^*(t) + \lambda_i(t)$$

The expected mortality rate is usually obtained from national or regional life tables matched on age, sex, and year. The survival analogue to the excess mortality is relative survival $R_i(t)$. Relative survival is related to the expected survival $S_i^*(t)$ and the all-cause survival in the diseased population $S_i(t)$ in the following way:

$$S_i(t) = S_i^*(t)R_i(t)$$

The log likelihood described in (2) can be extended when considering a relative survival model:

$$\log l_i = d_i \log\{h^*(t_i) + \lambda(t_i)\} + \log\{R(t_i)\}$$

We have discarded $\log\{S^*(t_i)\}$ from the log likelihood because the maximum likelihood does not depend on this value.

6 The strcs command

`strcs` fits flexible parametric survival models on the log-hazard scale. Restricted cubic splines smooth the log hazard with user-specified degrees of freedom. Covariates can be included within the model and are allowed to be time dependent by specifying degrees of freedom to model the time-dependent effect. Excess hazard models can be implemented by specifying the expected hazard rate. Numerical integration of the hazard function is undertaken as a two-step process by combining analytical integration with Gauss–Legendre quadrature. Both the `rcsgen` (Lambert 2008) and `stpm2` commands are called in `strcs` to create splines and get initial values, respectively; the user must install these prior to using the `strcs` command. The log likelihood is maximized using the Newton–Raphson algorithm, via the `ml` command in Stata. The likelihood is evaluated using Mata to increase computational speed.

6.1 Syntax

`strcs` *varlist* [*if*] [*in*] , {`df(`*#*`)` | `knots(`*numlist*`)`} [`bknots(`*knots_list*`)`
`bknotstvc(`*knots_list*`)` `dftvc(`*df_list*`)` `knotstvc(`*knots_list*`)` `knscale(`*scale*`)`
`tvc(`*varlist*`)` `bhazard(`*varname*`)` `bhtime` `noconstant` `nodes(`*#*`)` `noorthog`
`offset(`*varname*`)` `reverse` `level(`*#*`)` `nohr` `verbose` `from(`*matrix*`)`
`inith(`*varlist*`)` *maximize_options*]

6.2 Options

Knot selection options

`df(`*#*`)` specifies the degrees of freedom for the restricted cubic spline function used for the baseline hazard function; the number of degrees of freedom does not include the constant term. # must be between 1 and 10. With 1 degree of freedom, a linear effect is fit. The `knots()` option is not applicable if the `df()` option is specified. The knots are placed at equally spaced centiles of the uncensored survival times or log survival-times, depending on the `bhtime` option. For example, for `df(5)` with no `bhtime` option, knots are placed at the 20th, 40th, 60th, and 80th centiles of the distribution of the uncensored log survival-times. Note that these are interior knots and that there are also boundary knots placed at the minimum and maximum of the distribution of uncensored survival times or log survival-times. `df()` or `knots()` is required.

`knots(`*numlist*`)` specifies the knot locations for the baseline hazard function, as opposed to the locations set by the `df()` option. Note that the locations of the knots are placed on the scale defined by `knscale()`. However, the scale used by the restricted cubic splines function is always log time unless the `bhtime` option is specified. Default knot locations are determined by the `df()` option. `df()` or `knots()` is required.

`bknots(`*knots_list*`)` specifies the boundary knots. By default, these are located at the minimum and maximum of the uncensored survival times, or log survival-times depending on the use of the `bhtime` option. They are specified on the scale defined by `knscale()`.

`bknotstvc(`*knots_list*`)` specifies the boundary knots for any time-dependent effects. By default, these are the same as for the `bknots` option. They are specified on the scale defined by `knscale()`. For example, `bknotstvc(x1 0.01 10 x2 0.01 8)` specifies the boundary knots for covariate `x1` are `0.01` and `10` and for covariate `x2` are `0.01` and `8`.

`dftvc(`*df_list*`)` specifies the degrees of freedom for time-dependent effects in *df_list*. If there is more than one time-dependent effect and different degrees of freedom are requested for each time-dependent effect, then use the syntax `dftvc(x1:3 x2:2 1)`. This will use 3 degrees of freedom for covariate `x1`, 2 degrees of freedom for covariate `x2`, and 1 degree of freedom for all remaining time-dependent effects.

`knotstvc(`*knots_list*`)` defines the location of the interior knots for time-dependent effects. If different knots are required for different time-dependent effects, the option is specified as, for example, `knotstvc(x1 1 2 3 x2 1.5 3.5)`.

`knscale(`*scale*`)` sets the scale on which user-defined knots are specified. `knscale(time)` denotes the original time scale, `knscale(log)` denotes the log time scale, and `knscale(centile)` specifies that the knots are taken to be centile positions in the distribution of uncensored log survival-times if the `bhtime` option is not specified. The default is `knscale(time)`.

`tvc(`*varlist*`)` gives the name of the variables that are time dependent. Time-dependent effects are fit using restricted cubic splines. The degrees of freedom are specified using the `dftvc()` option.

Estimation options

`bhazard(`*varname*`)` invokes a relative survival model where *varname* holds the expected mortality rate (hazard) at the time of death or censoring.

`bhtime` smooths the estimated log-hazard function over time using restricted cubic splines. By default, smoothing is over log time.

`noconstant` suppresses the constant term (intercept) in the model.

`nodes(`*#*`)` specifies the number of nodes to be used in Gauss–Legendre quadrature numerical integration when calculating the estimated cumulative hazard function from the estimated hazard function. The default is `nodes(30)`. Changing the nodes may be useful if there are convergence problems. Too few nodes may result in a poor approximation involved in the numerical integration sensitivity. Analyses should be performed to ensure the results are not sensitive to the number of nodes.

`noorthog` suppresses orthogonal transformation of spline variables.

`offset(`*varname*`)` specifies a variable whose coefficient is constrained to be 1.

`reverse` specifies that the splines be calculated backward. See Andersson et al. (2011) for details of the approach.

Reporting options

`level(`*#*`)` specifies the confidence level, as a percentage, for the confidence intervals (CIs). The default is `level(95)` or as set by `set level`.

`nohr` reports the coefficients instead of hazard ratios.

`verbose` details the process of the `strcs` program in its output.

Maximization options

`from(`*matrix*`)` defines the parameter matrix of initial values to be used in maximum likelihood estimation. By default, `strcs` estimates initial hazard estimates by fitting a model on the log cumulative-hazard scale using the `stpm2` command.

`inith(`*varlist*`)` defines initial hazard estimates to be used in maximum likelihood estimation. By default, `strcs` estimates initial hazard estimates by fitting a model on the log cumulative-hazard scale using the `stpm2` command.

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(`*#*`)`, [`no`]`log`, `trace`, `gradient`, `showstep`, `hessian`, `shownrtolerance`, `tolerance(`*#*`)`, `ltolerance(`*#*`)`, `gtolerance(`*#*`)`, `nrtolerance(`*#*`)`, and `nonrtolerance`; see [R] **maximize**. These options are seldom used, but the `difficult` option may be useful if there are convergence problems.

7 The strcs postestimation command

7.1 Syntax

```
predict newvar [if] [in], {survival|hazard|xb|xbnobaseline|cumhazard|
   sdiff1(varname # [varname # ...])
   sdiff2(varname # [varname # ...])|
   hdiff1(varname # [varname # ...])
   hdiff2(varname # [varname # ...])|
   hrnumerator(varname # [varname # ...])
   hrdenominator(varname # [varname # ...])}
   [at(varname # [varname # ...]) [ci|stdp] nodes(#) per(#)
   timevar(varname) zeros level(#)]
```

7.2 Options

`survival` predicts the survival function. `survival`, `hazard`, `xb`, `xbnobaseline`, `cumhazard`, `sdiff1()` (and `sdiff2()` if applicable), `hdiff1()` (and `hdiff2()` if applicable), or `hrnumerator()` (and `hrdenominator()`) is required.

`hazard` predicts the hazard function. `survival`, `hazard`, `xb`, `xbnobaseline`, `cumhazard`, `sdiff1()` (and `sdiff2()` if applicable), `hdiff1()` (and `hdiff2()` if applicable), or `hrnumerator()` (and `hrdenominator()`) is required.

`xb` predicts the linear predictor, including the spline function. `survival`, `hazard`, `xb`, `xbnobaseline`, `cumhazard`, `sdiff1()` (and `sdiff2()` if applicable), `hdiff1()` (and `hdiff2()` if applicable), or `hrnumerator()` (and `hrdenominator()`) is required.

`xbnobaseline` predicts the linear predictor, excluding the spline function, that is, only the time-fixed part of the model. `survival`, `hazard`, `xb`, `xbnobaseline`, `cumhazard`, `sdiff1()` (and `sdiff2()` if applicable), `hdiff1()` (and `hdiff2()` if applicable), or `hrnumerator()` (and `hrdenominator()`) is required.

`cumhazard` predicts the cumulative hazard function using Gauss–Legendre quadrature numerical integration. `survival`, `hazard`, `xb`, `xbnobaseline`, `cumhazard`, `sdiff1()` (and `sdiff2()` if applicable), `hdiff1()` (and `hdiff2()` if applicable), or `hrnumerator()` (and `hrdenominator()`) is required.

`sdiff1(`*varname* # [*varname* # ...]`)` and
`sdiff2(`*varname* # [*varname* # ...]`)` predict the difference in survival curves with the first survival curve defined by the covariate values listed for `sdiff1()` and the second by those listed for `sdiff2()`. By default, covariates not specified using either option are set to zero. Note that setting the remaining values of covariates to zero may not always be sensible. If # is set to `.`, then *varname* takes its

observed values in the dataset. For example, `sdiff1(hormon 1)` (without specifying `sdiff2()`) computes the difference in predicted survival curves at `hormon` = `1` compared with `hormon` = `0`. `sdiff1(hormon 0) sdiff2(hormon 1)` computes the difference in predicted survival curves at `hormon` = `0` compared with `hormon` = `1`. `sdiff1(hormon 0 age 50) sdiff2(hormon 1 age 30)` computes the difference in predicted survival curves at `hormon` = `0` and `age` = `50` compared with `hormon` = `1` and `age` = `30`. `survival`, `hazard`, `xb`, `xbnobaseline`, `cumhazard`, `sdiff1()` (and `sdiff2()` if applicable), `hdiff1()` (and `hdiff2()` if applicable), or `hrnumerator()` (and `hrdenominator()`) is required.

`hdiff1(`*varname* # [*varname* # ...]`)` and `hdiff2(`*varname* # [*varname* # ...]`)` predict the difference in hazard functions with the first hazard function defined by the covariate values listed for `hdiff1()` and the second by those listed for `hdiff2()`. By default, covariates not specified using either option are set to zero. Note that setting the remaining values of the covariates to zero may not always be sensible. If # is set to `.`, then *varname* takes its observed values in the dataset. For example, `hdiff1(hormon 1)` (without specifying `hdiff2()`) computes the difference in predicted hazard functions at `hormon` = `1` compared with `hormon` = `0`. `hdiff1(hormon 0) hdiff2(hormon 1)` computes the difference in predicted hazard functions at `hormon` = `0` compared with `hormon` = `1`. `hdiff1(hormon 0 age 50) hdiff2(hormon 1 age 30)` computes the difference in predicted hazard functions at `hormon` = `0` and `age` = `50` compared with `hormon` = `1` and `age` = `30`. `survival`, `hazard`, `xb`, `xbnobaseline`, `cumhazard`, `sdiff1()` (and `sdiff2()` if applicable), `hdiff1()` (and `hdiff2()` if applicable), or `hrnumerator()` (and `hrdenominator()`) is required.

`hrnumerator(`*varname* # [*varname* # ...]`)` predicts the (time-dependent) hazard ratio with the numerator of the hazard ratio. By default, all covariates other than *varname* and any other variables mentioned are set to zero. Note that setting the remaining values of covariates to zero may not always be sensible. If # is set to `.`, then *varname* takes its observed values in the dataset. `survival`, `hazard`, `xb`, `xbnobaseline`, `cumhazard`, `sdiff1()` (and `sdiff2()` if applicable), `hdiff1()` (and `hdiff2()` if applicable), or `hrnumerator()` (and `hrdenominator()`) is required.

`hrdenominator(`*varname* # [*varname* # ...]`)` specifies the denominator of the hazard ratio. By default, all covariates other than *varname* and any other variables mentioned are set to zero. If # is set to `.`, then *varname* takes its observed values in the dataset. `survival`, `hazard`, `xb`, `xbnobaseline`, `cumhazard`, `sdiff1()` (and `sdiff2()` if applicable), `hdiff1()` (and `hdiff2()` if applicable), or `hrnumerator()` (and `hrdenominator()`) is required.

`at(`*varname* # [*varname* # ...]`)` requests that the covariates specified by the listed *varname* be set to the listed # values. For example, `at(x1 1 x3 50)` would evaluate predictions at `x1` = `1` and `x3` = `50`. This is a useful way to obtain out-of-sample predictions. Note that if `at()` is used together with `zeros`, all covariates not listed in `at()` are set to zero. If `at()` is used without `zeros`, then all covariates not listed in `at()` are set to their sample values; see also `zeros`.

`ci` calculates a CI for the requested statistics and stores the confidence limits in *newvar*`_lci` and *newvar*`_uci`. `ci` cannot be used with the `stdp` option.

`stdp` calculates the standard error of the prediction and stores it in *newvar*`_se`. `stdp` is available only for the `xb` and `xbnobaseline` options and cannot be used with the `ci` option.

`nodes(`*#*`)` specifies the number of nodes to be used when numerically integrating the estimated hazard function using Gauss–Legendre quadrature. Numerical integration is required when predicting the cumulative hazard and survival functions. The default is `nodes(30)`.

`per(`*#*`)` expresses hazard rates and differences in hazard rates per # person years.

`timevar(`*varname*`)` defines the variable used as time in the predictions. The default is `timevar(_t)`. This is useful for large datasets where, for plotting purposes, predictions are needed, for example, only for 200 observations. Note that some caution should be taken when using this option because predictions may be made at whatever covariate values are in the first 200 rows of data. This can be avoided by using the `at()` option or the `zeros` option, or both, to define the covariate patterns for which you require the predictions.

`zeros` sets all covariates to zero (baseline prediction). For example, `predict s0, survival zeros` calculates the baseline survival function. See also `at()`.

`level(`*#*`)` specifies the confidence level as a percentage. The default is `level(95)` or as set by `set level`.

8 Examples

We illustrate `strcs` through an application to 14,423 female patients diagnosed with breast cancer in England and Wales between 1986 and 1990 (Coleman et al. 1999) and consider the variables `old` and `deprived`. We use the binary variable `old` to consider the effect of age at diagnosis on survival (`old` $= 1$ for the oldest patients, age ≥ 80 years; `old` $= 0$ for the youngest patients, age < 50 years). We use the binary variable `deprived` to consider the effect of deprivation on survival (`deprived` $= 1$ for the most deprived patients; `deprived` $= 0$ for the least deprived patients). Observations other than those in these variable groups were removed to demonstrate the specific issue of modeling multiple time-dependent effects. The data must be declared as survival-time data using `stset` to fit an `strcs` model. We follow patients up to 5-years postdiagnosis; 6,426 events were observed during this follow-up time.

```
. use ew_breast
(Ch28 Adult Breast 174, 175)
. keep if agegroup==1 | agegroup==5
(76,332 observations deleted)
. tabulate dep, generate(dep)
  (output omitted)
```

```
. keep if dep1==1 | dep5==1
(24,576 observations deleted)

. count
  14,423

. generate deprived=1 if dep==5
(9,460 missing values generated)

. replace deprived=0 if dep==1
(9,460 real changes made)

. generate old=1 if agegroup==5
(9,721 missing values generated)

. replace old=0 if agegroup==1
(9,721 real changes made)

. stset survtime, failure(dead==1) exit(time 5)

     failure event:  dead == 1
obs. time interval:  (0, survtime]
 exit on or before:  time 5

------------------------------------------------------------------------------
      14423  total observations
          0  exclusions
------------------------------------------------------------------------------
      14423  observations remaining, representing
       6426  failures in single-record/single-failure data
  52269.807  total analysis time at risk and under observation
                                                at risk from t =         0
                                     earliest observed entry t =         0
                                          last observed exit t =         5
```

8.1 Proportional hazards model

A flexible parametric proportional hazards model on the log-hazard scale adjusting for deprivation and age at diagnosis is fit and shown here using the `strcs` command:

```
. strcs deprived old, df(5) nolog

Log likelihood = -17583.118                     Number of obs     =     14,423

------------------------------------------------------------------------------
             |  Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
xb           |
    deprived |   1.188827   .0305327     6.73   0.000     1.130466    1.250202
         old |   4.242514   .1081975    56.67   0.000     4.035664    4.459967
-------------+----------------------------------------------------------------
rcs          |
        __s1 |  -.2452037   .0183438   -13.37   0.000    -.2811569   -.2092505
        __s2 |  -.1028519   .0146591    -7.02   0.000    -.1315833   -.0741205
        __s3 |  -.0095586   .0130162    -0.73   0.463    -.0350699    .0159526
        __s4 |   .0464819   .0117706     3.95   0.000      .023412    .0695518
        __s5 |  -.0374921    .010923    -3.43   0.001    -.0589007   -.0160834
       _cons |  -2.753924   .0344452   -79.95   0.000    -2.821436   -2.686413
------------------------------------------------------------------------------
 Quadrature method: Gauss-Legendre with 30 nodes
```

These results indicate that the mortality rate for those who are deprived is 19% higher than that of the affluent, and the mortality rate for the oldest patients is 4.24

times that of the youngest. The estimates for `__s1`, `__s2`, ..., `__s5`, and `_cons` are not interpretable individually but together form the log baseline hazard function. Here we specify five degrees of freedom to model the log baseline hazard function through the `df(5)` option, which is reflected in the output having five restricted cubic spline variables, `__s1`, `__s2`, ..., `__s5`. Fitting a Cox model using `stcox` gives very similar estimates to those seen when fitting flexible parametric survival models on the log-hazard scale using `strcs`:

```
. stcox deprived old, nolog
         failure _d:  dead == 1
   analysis time _t:  survtime
  exit on or before:  time 5
Cox regression -- Breslow method for ties
No. of subjects =       14,423                  Number of obs    =      14,423
No. of failures =        6,426
Time at risk    =  52269.80699
                                                LR chi2(2)       =     3285.67
Log likelihood  =   -58181.114                  Prob > chi2      =      0.0000

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    deprived |   1.188597   .0305272     6.73   0.000     1.130245     1.24996
         old |   4.240279   .1081452    56.64   0.000     4.033529    4.457627
------------------------------------------------------------------------------
```

Similarly, fitting a flexible parametric survival model on the log cumulative-hazard scale gives very similar estimates:

```
. stpm2 deprived old, df(5) scale(hazard) eform nolog
Log likelihood = -17565.021                     Number of obs     =     14,423

------------------------------------------------------------------------------
             |     exp(b)   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
xb           |
    deprived |   1.188864   .0305337     6.74   0.000       1.1305    1.250241
         old |   4.243773   .1082262    56.68   0.000     4.036868    4.461284
       _rcs1 |   2.601464   .0314216    79.16   0.000     2.540602    2.663784
       _rcs2 |   .9448331   .0075656    -7.09   0.000     .9301205    .9597785
       _rcs3 |   .9538918   .0042863   -10.51   0.000     .9455277    .9623299
       _rcs4 |    1.01948   .0028672     6.86   0.000     1.013876    1.025116
       _rcs5 |   .9978683   .0015792    -1.35   0.178      .994778    1.000968
       _cons |   .1862209   .0039616   -79.01   0.000      .178616    .1941496
------------------------------------------------------------------------------
```

The knot positions and various other stored results can be found by using the **`ereturn list`** command after fitting a model using `strcs`. It is also simple to predict the hazard function after fitting a model by using the `predict` postestimation command. We illustrate this by predicting the hazard from the previously described proportional hazards `strcs` model and creating a new variable called `pred_hazard`.

```
. predict pred_hazard, hazard
```

We then plot over time since diagnosis; see figure 1. The predicted estimates shown here are from a proportional hazards model, so the rates shown in figure 1 are perfectly proportional.

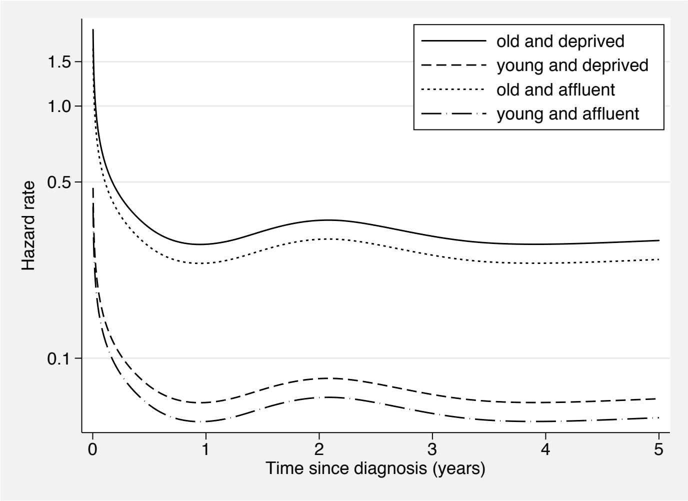

Figure 1. Predicted hazards from a proportional hazards flexible parametric model on the log-hazard scale using the `strcs` command

8.2 Nonproportional hazards model

We can fit a nonproportional hazards flexible parametric model on the log-hazard scale with time-dependent effects of deprivation and age at diagnosis using the `strcs` command:

```
. strcs deprived old, df(5) tvc(deprived old) dftvc(3) nolog
Log likelihood = -17374.818                     Number of obs     =     14,423

------------------------------------------------------------------------------
              |  Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
--------------+---------------------------------------------------------------
xb            |
     deprived |   1.078623   .0522633     1.56   0.118     .9809022    1.186079
          old |   5.124901   .2416034    34.66   0.000     4.672585    5.621001
--------------+---------------------------------------------------------------
rcs           |
         __s1 |     .00354   .0330689     0.11   0.915    -.0612739    .0683539
         __s2 |   .0433423   .0273338     1.59   0.113     -.010231    .0969155
         __s3 |   .1234013   .0211381     5.84   0.000     .0819714    .1648312
         __s4 |   .0770874    .012494     6.17   0.000     .0525996    .1015752
         __s5 |  -.0353503   .0109504    -3.23   0.001    -.0568126    -.013888
__s_deprived1 |  -.0881581   .0295574    -2.98   0.003    -.1460894   -.0302267
__s_deprived2 |   .0435977   .0258279     1.69   0.091    -.0070241    .0942195
__s_deprived3 |   .0237286   .0236893     1.00   0.317    -.0227014    .0701587
     __s_old1 |   -.233187   .0359835    -6.48   0.000    -.3037133   -.1626607
     __s_old2 |  -.2568875   .0300956    -8.54   0.000    -.3158738   -.1979012
     __s_old3 |  -.2564321   .0261969    -9.79   0.000    -.3077772   -.2050871
        _cons |  -2.838495   .0412506   -68.81   0.000    -2.919344   -2.757645
------------------------------------------------------------------------------
 Quadrature method: Gauss-Legendre with 30 nodes
```

The hazard ratios for the effect of deprivation and age at diagnosis are now allowed to vary over time through the use of the `tvc()` option. The `dftvc(3)` option specifies that three degrees of freedom, or two internal knots, should be used to model the deviations in the time-dependent effects. One can also select different degrees of freedom for each time-dependent effect using the `dftvc()` option, as illustrated here:

```
. strcs deprived old, df(5) tvc(deprived old) dftvc(deprived:2 old:3) nolog
Log likelihood = -17375.262                     Number of obs     =     14,423

------------------------------------------------------------------------------
              |  Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
--------------+---------------------------------------------------------------
xb            |
     deprived |   1.112886   .0393846     3.02   0.003      1.03831    1.192819
          old |   5.107861   .2401446    34.69   0.000     4.658221    5.600903
--------------+---------------------------------------------------------------
rcs           |
         __s1 |  -.0010421   .0327606    -0.03   0.975    -.0652517    .0631675
         __s2 |   .0477148   .0269529     1.77   0.077    -.0051118    .1005414
         __s3 |   .1304704   .0198034     6.59   0.000     .0916566    .1692843
         __s4 |   .0794761   .0122591     6.48   0.000     .0554488    .1035034
         __s5 |  -.0354069   .0109494    -3.23   0.001    -.0568673   -.0139464
__s_deprived1 |  -.0720214   .0241037    -2.99   0.003    -.1192637   -.0247791
__s_deprived2 |   .0270911   .0189817     1.43   0.154    -.0101124    .0642945
     __s_old1 |  -.2349027   .0359378    -6.54   0.000    -.3053396   -.1644659
     __s_old2 |  -.2551347   .0300374    -8.49   0.000    -.3140069   -.1962625
     __s_old3 |  -.2539953   .0260704    -9.74   0.000    -.3050923   -.2028983
        _cons |  -2.847685   .0402228   -70.80   0.000    -2.926521    -2.76885
------------------------------------------------------------------------------
 Quadrature method: Gauss-Legendre with 30 nodes
```

This specifies that the effect of deprivation is allowed to vary with time with two degrees of freedom, while the effect of age is allowed to vary with three degrees of freedom. The hazard ratios for the effects of deprivation and age at diagnosis can be predicted as follows:

```
. predict hr_deprived, hrnumerator(deprived 1) hrdenominator(deprived 0) ci
. predict hr_old, hrnumerator(old 1) hrdenominator(old 0) ci
```

The `hrnumerator()` and `hrdenominator()` options are used to define which covariate patterns are included in the numerator and the denominator of the hazard ratio, respectively. The `ci` option calculates CIs for the specified predictions and saves the lower and upper confidence limits as *varname*`_lci` and *varname*`_uci`, respectively. These predicted hazard ratios are displayed in figure 2.

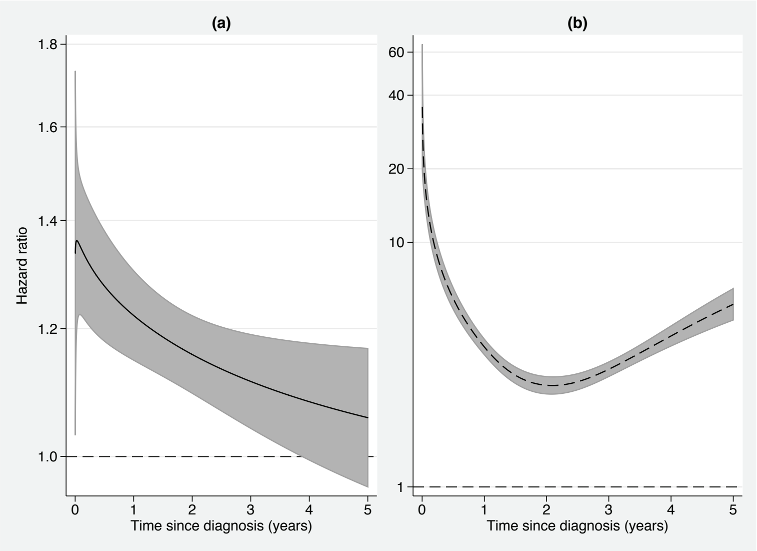

Figure 2. Predicted hazard ratios from nonproportional hazards model. (a) Hazard ratio for deprivation effect (deprived versus affluent). (b) Hazard ratio for age effect (old versus young).

Out-of-sample predictions and predictions for certain covariate patterns can be implemented using the `timevar()` and `at()` options, respectively. These are illustrated for a prediction of the survival function in addition to predictions for survival differences, the cumulative hazard function, and hazard-rate differences:

```
. range temptime 0 10 200
(14,223 missing values generated)
. predict pred_surv, surv timevar(temptime) at(deprived 1 old 1) ci
. predict pred_sdiff_age, sdiff1(old 1) sdiff2(old 0) ci
. predict pred_chazard, cumhazard
. predict pred_hdiff_age, hdiff(old 1) hdiff2(old 0) ci
```

The `range` command creates a variable named `temptime` that contains 200 observations with equally spaced values from 0 to 10. This variable is used to predict survival up to 10 years postdiagnosis, even though the analysis was based upon follow-up to 5 years. Figure 3 displays the results from the above predictions.

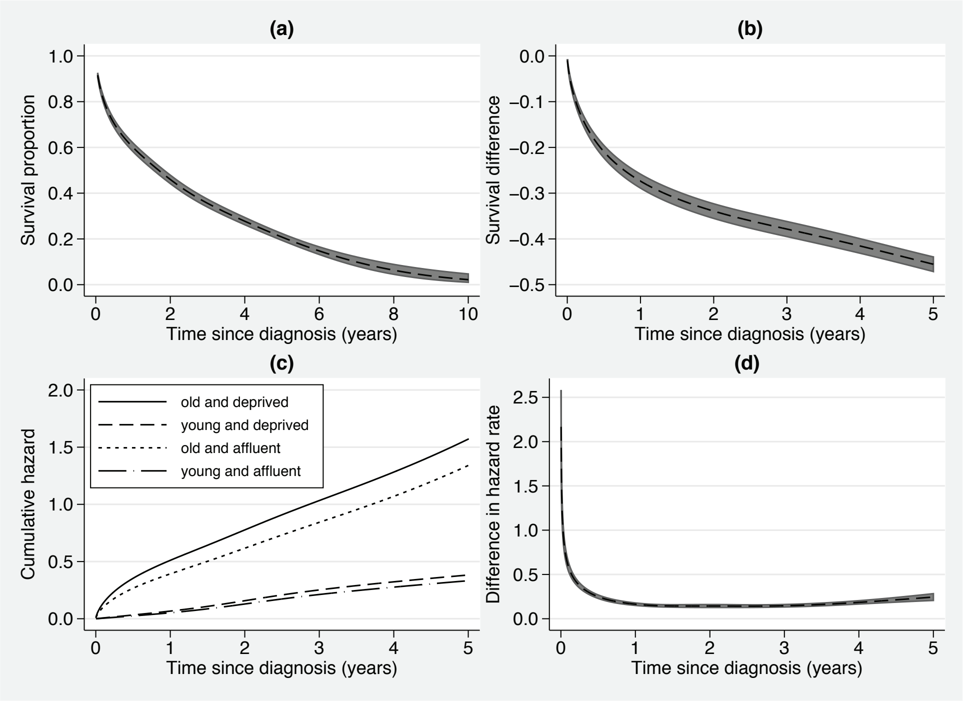

Figure 3. Predictions from flexible parametric nonproportional hazards model using `strcs`. (a) Survival function with 95% CI for the oldest patients at diagnosis who were the most deprived, up to 10 years postdiagnosis. (b) Survival difference with 95% CI between the oldest patients and the youngest patients. (c) Cumulative hazards of each deprivation group and age group. (d) Difference in hazard rates with 95% CI over time between the oldest patients and the youngest patients.

8.3 Time versus log time

It is common to model log time when fitting flexible parametric survival models. Transforming to the log time scale has been shown to generally fit better than on the untransformed time scale when using the same degrees of freedom (Royston 2000; Royston and Lambert 2011). In `strcs`, one can also model on the time scale by specifying the `bhtime` option. There are situations where one may prefer to model untransformed time, for example, when modeling with age as the time scale. Also, when modeling the log hazard with log time, one must be cautious when considering the hazard rate close to zero because at this point, the hazard rate is zero or infinity. Figure 4 shows the differences when modeling on the log-transformed time scale and the untransformed time scale. In this example, modeling on the log time scale is more stable to changes in the baseline degrees of freedom. The Akaike information criterion and Bayesian information criterion also suggest that the log-transformed time scale generally provides a better fit than the untransformed time scale when using the same degrees of freedom.

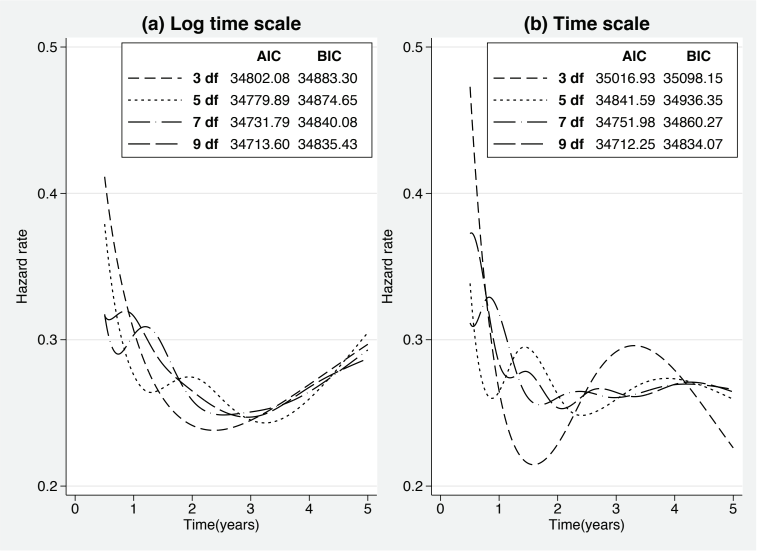

Figure 4. Predictions of the hazard rate of deprived patients from a flexible parametric nonproportional hazards model using `strcs` with different baseline degrees of freedom. (a) Displays predictions from models on the log time scale. (b) Displays predictions from models on the time scale. Results are presented from six months.

8.4 Number of nodes and degrees of freedom

Increasing the degrees of freedom or altering the knot positions can make the fit model more complicated. In some complex scenarios, models may not converge with the default number of nodes; if this is the case, the number of nodes can be altered by using the `nodes()` option. Also, to ensure that the number of nodes specified is enough to provide a good approximation in the numerical integration estimation, one should perform sensitivity analyses. The following output illustrates how altering the number of nodes can be implemented. Figure 5 displays the predicted baseline hazard rate from the `strcs` models with differing nodes.

```
. forvalues nodes = 5(5)30 {
  2.    qui strcs deprived old, df(3)  nodes(`nodes´) tvc(deprived)
>                       dftvc(1) nolog
  3.    predict haz_dep_nodes`nodes´, haz at(dep 0 old 0)
  4. }
```

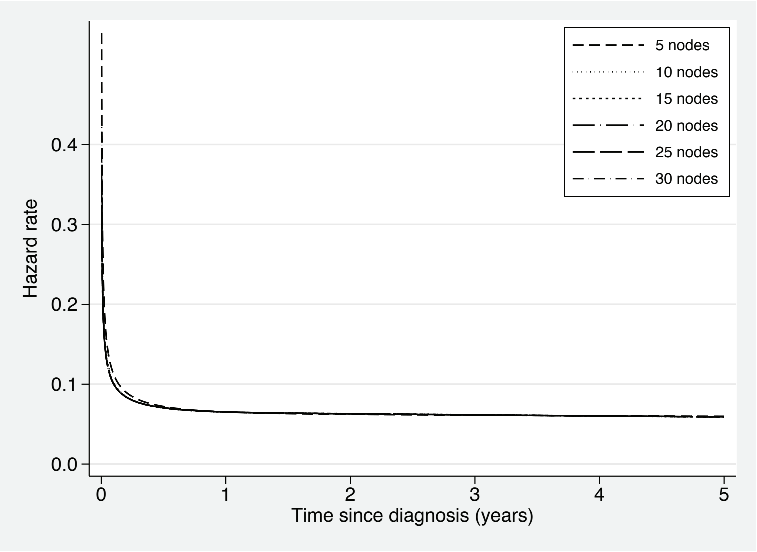

Figure 5. Baseline hazard rate (old, affluent patients) predicted from flexible parametric survival models on the log-hazard scale using `strcs` with varying nodes

The predicted hazard rates are very similar in figure 5, except for when 5 nodes are specified. This suggests that in this example, provided the number of nodes is at least 15, the approximation involved in the Gauss–Legendre numerical integration is accurate.

Table 1 shows the estimated parameters and standard errors from a proportional hazards model fit using the two-step approach implemented in `strcs` and the fully numeric approach implemented in `stgenreg`. Comparing the two integration methods indicates that the two-step approach obtains more consistent estimates with a lower number of nodes than the fully numeric approach. For example, the estimates of `_cons` are the same to 5 decimal points after 40 nodes is specified when implementing the two-step approach in `strcs`, whereas the estimates of `_cons` are more different when implementing the fully numeric approach in `stgenreg`.

Table 1. Estimated hazard ratios (standard errors) from a proportional hazards flexible parametric survival model on the log-hazard scale with 3 degrees of freedom. The estimates from the two-step approach are estimated using `strcs`, and those from the fully numeric approach are estimated using `stgenreg`.

		Number of nodes						
	Parameter	5	10	20	30	40	50	100
Two-step approach (`strcs`)	`deprived`	1.18672	1.18804	1.18856	1.18867	1.18869	1.18870	1.18871
		(0.02568)	(0.02568)	(0.02568)	(0.02568)	(0.02568)	(0.02568)	(0.02568)
	`old`	4.19165	4.22481	4.23820	4.24098	4.24178	4.24203	4.24216
		(0.02554)	(0.02551)	(0.02550)	(0.02550)	(0.02550)	(0.02550)	(0.02550)
	`__s1`	0.75738	0.77458	0.78323	0.78524	0.78582	0.78600	0.78609
		(0.01399)	(0.01407)	(0.01409)	(0.01409)	(0.01409)	(0.01409)	(0.01409)
	`__s2`	0.75738	0.90985	0.91782	0.91965	0.92017	0.92034	0.92042
		(0.01222)	(0.01222)	(0.01224)	(0.01224)	(0.01224)	(0.01224)	(0.01224)
	`__s3`	1.00613	1.01603	1.01252	1.01068	1.01004	1.00983	1.00972
		(0.01105)	(0.01114)	(0.01133)	(0.01137)	(0.01137)	(0.01138)	(0.01138)
	`_cons`	0.06420	0.06299	0.06282	0.06283	0.06284	0.06284	0.06284
		(0.02725)	(0.02725)	(0.02727)	(0.02727)	(0.02727)	(0.02727)	(0.02727)
Fully numeric approach (`stgenreg`)	`deprived`	1.17739	1.18440	1.18750	1.18812	1.18834	1.18845	1.18861
		(0.02570)	(0.02569)	(0.02568)	(0.02568)	(0.02568)	(0.02568)	(0.02568)
	`old`	4.01305	4.14870	4.21467	4.22872	4.23377	4.23620	4.23988
		(0.02584)	(0.02564)	(0.02553)	(0.02552)	(0.02551)	(0.02551)	(0.02550)
	`__s1`	0.66201	0.72011	0.75783	0.76949	0.77460	0.77737	0.78221
		(0.01444)	(0.01448)	(0.01438)	(0.01429)	(0.01424)	(0.01421)	(0.01415)
	`__s2`	0.66201	0.85692	0.89306	0.90429	0.90921	0.91189	0.91661
		(0.01250)	(0.01255)	(0.01247)	(0.01240)	(0.01236)	(0.01234)	(0.01229)
	`__s3`	1.10138	1.08763	1.05100	1.03544	1.02806	1.02393	1.01641
		(0.01146)	(0.01170)	(0.01193)	(0.01185)	(0.01177)	(0.01171)	(0.01157)
	`_cons`	0.06411	0.06187	0.06193	0.06221	0.06237	0.06246	0.06265
		(0.02752)	(0.02744)	(0.02742)	(0.02738)	(0.02736)	(0.02734)	(0.02731)

One can specify the degrees of freedom using the df() option as described previously. One can also specify the degrees of freedom, or the position of the knots, by using the knots() and the knotstvc() options, alongside the knscale() option, for the baseline hazard and the time-dependent effects, respectively. It is useful to investigate how different specified degrees of freedom affect the model estimates. This has previously been investigated on the log cumulative-hazard scale; see Lambert and Royston (2009) for further information because the same issues apply here.

8.5 Excess mortality models

Excess mortality models can be easily implemented using the bhazard() option. This option specifies the variable that contains the expected hazard rate. A proportional excess-hazards model can be fit as follows:

```
. generate _year=min(year(datediag+_t),2010)
. generate _age=floor(min(agediag+_t,99))
. merge m:1 _year sex _age using popmort2011, nolabel keepusing(rate)
> assert(2 3) keep(3) noreport nogenerate
. strcs deprived old, df(5) bhazard(rate) nolog
Log likelihood = -16232.411                     Number of obs     =     14,423

             Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]

xb
    deprived   1.276213   .0408588     7.62   0.000     1.198592     1.35886
         old   2.640435   .0877225    29.23   0.000      2.47398    2.818089

rcs
        __s1  -.3002251   .0231134   -12.99   0.000    -.3455265   -.2549237
        __s2  -.0938537   .0184632    -5.08   0.000     -.130041   -.0576664
        __s3   .0113959   .0162236     0.70   0.482    -.0204018    .0431937
        __s4   .0711007   .0147399     4.82   0.000     .0422111    .0999903
        __s5  -.0503612   .0135819    -3.71   0.000    -.0769813   -.0237411
       _cons  -2.856882   .0418619   -68.25   0.000     -2.93893   -2.774834

 Quadrature method: Gauss-Legendre with 30 nodes
```

Here we create variables for attained year and age to merge in the correct expected rates from a population mortality file. We then use these rates to fit the model on the log excess-hazard scale. This model provides estimates of excess hazard ratios; time-dependent excess-hazard ratios can be estimated similarly to the methods for including time-dependent effects described previously. Predictions from this model estimate excess hazard rates and relative survival proportions.

9 Conclusion

strcs is an extension to the general tool stgenreg that fits flexible parametric models on the log-hazard scale in a more efficient, user-friendly way with extended postestimation prediction tools. strcs implements a two-step integration process to increase the accuracy of integration. Modeling using `strcs` avoids problems with multiple time-dependent effects that may be present when fitting flexible parametric survival models on the log cumulative-hazard scale using `stpm2`.

10 References

Andersson, T. M.-L., P. W. Dickman, S. Eloranta, and P. C. Lambert. 2011. Estimating and modelling cure in population-based cancer studies within the framework of flexible parametric survival models. *BMC Medical Research Methodology* 11: 96.

Bower, H., M. J. Crowther, M. J. Rutherford, T. M.-L. Andersson, M. Clements, X.-R. Liu, P. W. Dickman, and P. C. Lambert. 2015. Capturing simple and complex time-dependent effects using flexible parametric survival models: A simulation study. Unpublished manuscript.

Carstensen, B. 2007. Age–period–cohort models for the Lexis diagram. *Statistics in Medicine* 26: 3018–3045.

Coleman, M. P., P. Babb, P. Damiecki, P. Grosclaude, S. Honjo, J. Jones, G. Knerer, A. Pitard, M. Quinn, A. Sloggett, and B. De Stavola. 1999. *Cancer Survival Trends in England and Wales, 1971–1995: Deprivation and NHS Region*. London: Stationery Office.

Cox, C. 2008. The generalized *F* distribution: An umbrella for parametric survival analysis. *Statistics in Medicine* 27: 4301–4312.

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

Crowther, M. J., and P. C. Lambert. 2013a. Simulating biologically plausible complex survival data. *Statistics in Medicine* 32: 4118–4134.

———. 2013b. stgenreg: A Stata package for general parametric survival analysis. *Journal of Statistical Software* 53(12): 1–17.

———. 2014. A general framework for parametric survival analysis. *Statistics in Medicine* 33: 5280–5297.

Crowther, M. J., M. P. Look, and R. D. Riley. 2014. Multilevel mixed effects parametric survival models using adaptive Gauss–Hermite quadrature with application to recurrent events and individual participant data meta-analysis. *Statistics in Medicine* 33: 3844–3858.

Durrleman, S., and R. Simon. 1989. Flexible regression models with cubic splines. *Statistics in Medicine* 8: 551–561.

Lambert, P. 2008. rcsgen: Stata module to generate restricted cubic splines and their derivatives. Statistical Software Components S456986, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s456986.html.

Lambert, P. C., and P. Royston. 2009. Further development of flexible parametric models for survival analysis. *Stata Journal* 9: 265–290.

Lambert, P. C., L. K. Smith, D. R. Jones, and J. L. Botha. 2005. Additive and multiplicative covariate regression models for relative survival incorporating fractional polynomials for time-dependent effects. *Statistics in Medicine* 24: 3871–3885.

Nelson, C. P., P. C. Lambert, I. B. Squire, and D. R. Jones. 2007. Flexible parametric models for relative survival, with application in coronary heart disease. *Statistics in Medicine* 26: 5486–5498.

Royston, P. 2000. Choice of scale for cubic smoothing spline models in medical applications. *Statistics in Medicine* 19: 1191–1205.

Royston, P., and P. C. Lambert. 2011. *Flexible Parametric Survival Analysis Using Stata: Beyond the Cox Model*. College Station, TX: Stata Press.

Royston, P., and M. K. B. Parmar. 2002. Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. *Statistics in Medicine* 21: 2175–2197.

Rutherford, M. J., M. J. Crowther, and P. C. Lambert. 2015. The use of restricted cubic splines to approximate complex hazard functions in the analysis of time-to-event data: A simulation study. *Journal of Statistical Computation and Simulation* 85: 777–793.

About the authors

Hannah Bower is a PhD student in the Department of Medical Epidemiology and Biostatistics, Karolinska Institutet, Stockholm, Sweden.

Michael J. Crowther is a postdoc in biostatistics with a joint appointment at the University of Leicester, Leicester, UK, and the Department of Medical Epidemiology and Biostatistics, Karolinska Institutet, Stockholm, Sweden.

Paul C. Lambert is a professor of biostatistics at the University of Leicester in Leicester, UK. He has a long-term secondment arrangement with the Department of Medical Epidemiology and Biostatistics at Karolinska Institutet.