The Stata Journal (2001)
1, Number 1, pp. 1–28

Flexible parametric alternatives to the Cox model, and more

Patrick Royston
UK Medical Research Council
patrick.royston@ctu.mrc.ac.uk

Abstract. Since its introduction to a wondering public in 1972, the Cox proportional hazards regression model has become an overwhelmingly popular tool in the analysis of censored survival data. However, some features of the Cox model may cause problems for the analyst or an interpreter of the data. They include the restrictive assumption of proportional hazards for covariate effects, and "loss" (non-estimation) of the baseline hazard function induced by conditioning on event times. In medicine, the hazard function is often of fundamental interest since it represents an important aspect of the time course of the disease in question. In the present article, the Stata implementation of a class of flexible parametric survival models recently proposed by Royston and Parmar (2001) will be described. The models start by assuming either proportional hazards or proportional odds (user-selected option). The baseline distribution function is modeled by restricted cubic regression spline in log time, and parameter estimation is by maximum likelihood. Model selection and choice of knots for the spline function are discussed. Interval-censored data and models in which one or more covariates have non-proportional effects are also supported by the software. Examples based on a study of prognostic factors in breast cancer are given.

Keywords: st0001, parametric survival analysis, hazard function, proportional hazards, proportional odds

1 Introduction

Cox proportional hazards regression has essentially become the automatic choice of analysis tool for modeling survival data in medical studies. However, the Cox model has intrinsic features that may cause problems for the analyst or the interpreter of the data:

- It treats the baseline distribution of the observations as a high-dimensional nuisance parameter. For example, a typical estimate of the baseline hazard function following Cox is a "noisy" step function.
- It assumes that covariate effects act proportionally on the baseline hazard function, independent of time. This strong assumption is often not checked.
- Extending it to allow for nonproportional hazards is by no means a trivial modeling exercise.

- It does not give a complete probability specification for the data. Validation of the model and simulation of datasets realistically similar to a given one are impeded.

Hjort (1992) aptly noted that

> "A parametric version [of the Cox model], ... if found to be adequate, would lead to more precise estimation of survival probabilities and ... concurrently contribute to a better understanding of the phenomenon under study."

In the present article, I will present parametric versions of the Cox model and more. The idea of spline-smoothing the distribution function was suggested by Efron (1988) and was taken up by other authors, as indicated below. The models are implemented in an ado-file called `stpm`. Further details and additional examples are given by Royston and Parmar (2001).

Visualisation of the survival function for censored survival data is easily done by using the Kaplan–Meier plot. In the Cox model, however, the baseline hazard function is regarded as a high-dimensional nuisance parameter and is highly erratic. The behavior of the hazard function is certainly of potential interest because it directly reflects the time course of the process under study. To estimate it informatively (i.e., smoothly), some type of parametric model may be appropriate. For example, Gelfand et al. (2000) proposed a parametric method based on a mixture of Weibull distributions. Other significant recent contributions to hazard regression include Kooperberg et al. (1995) and Rosenberg (1995).

A second important issue is how to deal with nonproportional hazards. Although the Cox model may be extended to allow for nonproportional hazards, such as by incorporating time-varying regression coefficients. See, for example, Hastie and Tibshirani (1993) or Hess (1994). There is no natural, widely-accepted approach, and obtaining a satisfactory model can be complicated. There are further concerns about the complexity involved in the practical interpretation of the coefficients and in the robustness of such models.

The proportional hazards model is well known, but the proportional odds model for survival data also has a fairly long history. It was first described in a semiparametric framework by Bennett (1983), was further developed by several authors including Yang and Prentice (1999), and was adapted by Rossini and Tsiatis (1996) for modeling current status (i.e., interval censored) data.

Here, I will present the Stata ado-file `stpm`, which implements the flexible parametric models described by Royston and Parmar (2001). Generically, such models are based on transformation of the survival function by a link function $g(.)$,

$$g\{S(t;\mathbf{z})\} = g\{S_0(t)\} + \beta'\mathbf{z}$$

where $S_0(t) = S(t; \mathbf{0})$ is the baseline survival function and β is a vector of parameters to be estimated for covariates $\mathbf{z}$. Within this framework, Younes and Lachin (1997) used the parameterized link function of Aranda-Ordaz (1981),

$$g(x;\theta) = \log \frac{x^{-\theta} - 1}{\theta}$$

where $\theta = 1$ corresponds to the proportional odds model and $\theta \to 0$ to the proportional hazards model. Younes and Lachin (1997) took a related approach, estimating the baseline hazard function by B-splines and determining $S_0(t)$ by integration. Shen (1998) used sieve maximum likelihood and monotone splines with variable orders and knots to estimate very flexible proportional odds models.

The Royston and Parmar (2001) approach is to use natural cubic splines to model $g[S_0(t)]$ within the Aranda-Ordaz family of link functions. They chose to work only with the odds ($\theta = 1$) and hazards ($\theta \to 0$) scaling, rather than with more general values of θ for which the interpretation of covariate effects is obscure. Models with a probit link function, which extend the lognormal distribution, are also supported by `stpm`, but are not further described here. When smoothing of $g[S_0(t)]$ is implemented on the log-time scale, as here, the fitted function is typically gently curved or nearly linear, and is usually very smooth. The smoothness tends to reduce the chance of artifacts in the estimated hazard function. The estimate of $g[S_0(t)]$ must theoretically be monotone in t, whereas natural cubic splines, which are constrained to be linear beyond certain extreme observations, are not globally monotone. However, the linearity constraint imposes monotonicity in the tail regions where the observed data are sparse, whereas in regions where data are dense (and provided the sample size is not too small), monotonicity is effectively imposed by the data themselves.

When the relationship between the baseline log cumulative hazard or log cumulative odds of failure and log time is modeled as linear rather than by using splines, the approach reduces to fitting Weibull or log-logistic distributions. The Weibull is, of course, familiar as a model for lifetimes. The (generalized) log-logistic distribution has been used for survival modeling by Mackenzie (1996) and for modeling recurrent event data by Mackenzie (1997).

Extensions of the basic models to include models with nonproportional scaling for some subset of the covariates are also briefly mentioned. The covariates may be of any type (binary, categoric, or continuous). Nonproportionality is induced by multiplicative interactions between the covariates and the spline basis functions. Such models are more complex than the basic ones and require extra care in construction, evaluation of appropriateness, and interpretation. Even under statistically significant but quantitatively mild departures from proportionality, the proportionately scaled models may give a description of the data which is adequate for practical purposes.

2 Syntax

`stpm` is a regression-like command with the following syntax:

`stpm` [*varlist*] [`if` *exp*] [`in` *range*] , *model_complexity*
`scale(hazard | normal | odds)` [`left(`*leftvar*`)` `stratify(`*strat_varlist*`)`
`noconstant nolog offset(`*offsetvar*`)` `spline(`*splinevar derivativevar*`)`
`theta(est | #)`]

where *model_complexity* is `df(`*#*`)` or `knots(`[`l`|`%`]*knotlist*`)` or `knots(u`*#1 #2 #3*`)`.

`stpm` is for use with `st` data. You must `stset` your data first.

3 Options

Note that the complexity of the spline part of the model is defined by either `df()` or `knots()`, so one (but not both) of these options must be specified.

`df(`*#*`)` specifies the degrees of freedom for the natural spline function, and must be between 1 and 6. The `knots()` option is not applicable, and the knots are placed at the following centiles of the distribution of the uncensored log times:

df	Centile positions
1	(no knots)
2	50
3	33 67
4	25 50 75
5	20 40 60 80
6	17 33 50 67 83
>6	(not allowed)

`knots(`[`l`|`%`]*knotlist*`)` defines the internal knot positions for the spline. If you specify `knots(`*knotlist*`)`, then *knotlist* should consist of values of log time. If you specify `knots(l`*knotlist*`)`, then the values in *knotlist* are taken to be times and are automatically log transformed by `stpm`. (This is a convenience feature; it is easier to enter times than it is to enter log times.) If you specify `knots(%`*knotlist*`)`, then the values in *knotlist* are taken to be centile positions in the distribution of the uncensored log times.

`knots(u`*#1 #2 #3*`)` also defines the internal knots for the spline. #1 knots are assigned at random uniformly distributed positions between #2 and #3, where #2 is the lowest centile position of the uncensored log times you wish to entertain and #3 is the highest. A suggested choice is #2 = 10, #3 = 90; knots are to be placed at random positions between the 10th and 90th centiles of the uncensored log times.

`scale(hazard | normal | odds)` is not optional and specifies the scale of the model. The `hazard`, `normal` and `odds` options fit models on the scale of the log cumulative hazard, normal equivalent deviate, or log-cumulative odds of failure respectively.

`left(`*leftvar*`)` specifies that some or all of the survival-time observations are interval-censored. The rules for specifying values of *leftvar* and their meanings in terms of interval censoring are as follows:

leftvar	`_d`	Meaning
. or `_t`	0	Right censored at `_t`
. or `_t`	1	Event at `_t`
0	0	Right censored at `_t`
0	1	Interval censored, event in (0,`_t`]
<`_t`	0	Late entry at *leftvar*, right censored at `_t`
<`_t`	1	Interval censored, event in [*leftvar*,`_t`]

Note that `stpm` does not support datasets with late entry (specified via the `enter()` option of `stset` and generating positive values of the variable `_t0`) and interval censoring together, except when the late entry is specified by way of *leftvar* and `_d` as in the above table.

`stratify(`*strat_varlist*`)` stratifies the spline functions according to the variables in *strat_varlist*. It will rarely make sense for the variables in *strat_varlist* not to be among those in *varlist*, but this is not checked.

`noconstant` suppresses the constant term in the xb equation.

`nolog` suppresses the iteration log while the model is fit.

`offset(`*offsetvar*`)` defines the offset for the xb equation. *offsetvar* is added to the linear predictor.

`spline(`*splinevar derivativevar*`)` allows you to specify the baseline spline function and its derivative with respect to log(`_t`). For a given model where the spline function has been estimated, *splinevar* can be created by using, for example,

```
. predict < splinevar>, zero < scale_option>
```

where <*scale_option*> is `cumodds`, `cumhazard`, or `normal`. *derivativevar* can be created by using, for example,

```
. predict < derivativevar>, dzdy
```

`theta(est | #)` only applies with `scale(odds)` and estimates the transformation parameter θ or performs estimation with θ set to #. The transformation of the (baseline) survival function $S_0(t)$ is then

$$g_\theta\{S_0(t)\} = \ln\left\{\frac{S_0(t)^{-\theta}-1}{\theta}\right\}$$

Thus, `theta` $= 0$ corresponds to the cumulative hazards model. With `theta(est)`, θ is estimated and presented on a log scale, i.e., $\ln(\theta)$. With `theta(#)`, # must be positive.

4 Example: Breast cancer data

I will use a subset of the data for 686 node-positive breast cancer patients analyzed in great detail by Sauerbrei and Royston (1999). The data are provided in `bc.dta`. The full dataset is available on web site *http://www.blackwellpublishers.co.uk/rss/*. The outcome of interest is the recurrence-free survival time; that is, the duration in years from entry into the study (typically, the time of diagnosis of primary breast cancer) until either death or disease recurrence, whichever occurred first. There were 299 events for this outcome, and the median follow-up time was about 5 years. The dataset in `bc.dta` has already been `stset`. In addition, three prognostic groups of about equal size have been created for use in this exercise. The grouping is based on prognostic model III of Sauerbrei and Royston (1999), and takes into account the patient's age, number of positive lymph nodes (a number that is positively and strongly associated with a poor prognosis), the tumor grade, the tumor progesterone receptor status, and whether the patient had received hormonal treatment. The three groups are labeled Good, Medium, and Poor. Figure 1 shows the Kaplan–Meier survival curves for these groups.

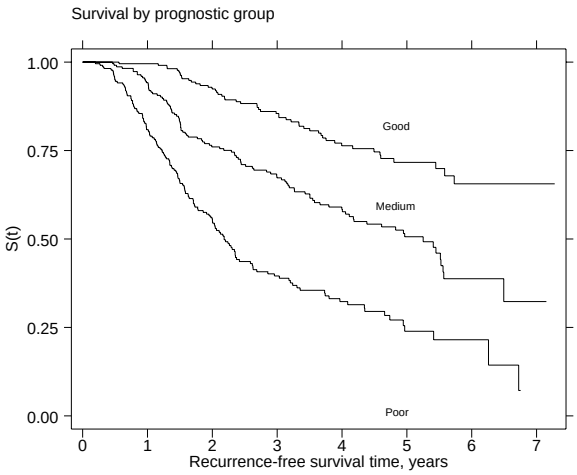

Figure 1: Kaplan–Meier survival curves for prognostic groups in the breast cancer data.

`bc.dta` also contains dummy variables `group2` and `group3` which are indicators of membership in the Medium and Poor groups, respectively. (The Good group is therefore indicated by `group2==0` and `group3==0`.) The output from running `stcox` is as follows:

```
. stcox group2 group3
         failure _d:  censrec
   analysis time _t:  rectime/365.25
Iteration 0:   log likelihood = -1788.1731
Iteration 1:   log likelihood = -1734.3879
Iteration 2:   log likelihood = -1731.1803
Iteration 3:   log likelihood = -1731.1664
Refining estimates:
Iteration 0:   log likelihood = -1731.1664
```

```
Cox regression -- Breslow method for ties
No. of subjects =          686                     Number of obs   =       686
No. of failures =          299
Time at risk    =  2111.978097
                                                   LR chi2(2)      =    114.01
Log likelihood  =   -1731.1664                     Prob > chi2     =    0.0000
```

_t _d	Haz. Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
group2	2.316138	.3969696	4.90	0.000	1.655292	3.240815
group3	5.041443	.8295395	9.83	0.000	3.651697	6.960093

Clearly the prognostic grouping is highly significant, with hazard ratios (HRs) of 2.3 and 5.0 for the Medium and Poor groups compared with the Good group.

What of the proportional hazards assumption? We re-estimate the Cox model and save the Schoenfeld and scaled Schoenfeld residuals and then carry out Stata's test of the PH assumption, `stphtest`:

```
. stcox group2 group3, schoenfeld(sch*) scaledsch(sca*)
  (output omitted)
. stphtest, rank detail
      Test of proportional hazards assumption
      Time:  Rank(t)
```

	rho	chi2	df	Prob>chi2
group2	-0.10829	3.49	1	0.0616
group3	-0.18132	9.41	1	0.0022
global test		9.56	2	0.0084

The results show that there is definite evidence of nonproportional hazards overall ($p = 0.008$) and nonproportional hazards for at least the second dummy variable ($p = 0.002$).

Let us now reanalyze the data using `stpm`. We will fit models with two different metrics for the covariates: proportional hazards (PH) and proportional odds (PO). The former metric is familiar. The latter metric is also familiar; it is the one usually used in logistic regression of a binary outcome variable on covariates, where the regression coefficient expresses the effect of a covariate on the odds of an event. For survival analysis, see Bennett (1983). The assumption of proportional odds means that the effect of the covariate on the *cumulative* odds of an event is independent of time. If $S(t)$ is the survival function, then the cumulative odds function is defined as $\{1 - S(t)\}/S(t)$. Compare this with the proportional hazards assumption, which may also be expressed in terms of the cumulative hazard function.

4.1 Parametric proportional hazards modeling

The command to fit a parametric PH model to the breast cancer data is, for example,

```
. stpm group2 group3, df(2) scale(hazard)
initial:       log likelihood = -619.87786
rescale:       log likelihood = -619.87786
rescale eq:    log likelihood = -619.87786
Iteration 0:   log likelihood = -619.87786
Iteration 1:   log likelihood = -619.37792
Iteration 2:   log likelihood = -619.37621
Iteration 3:   log likelihood = -619.37621
                                                Number of obs   =        686
                                                Wald chi2(2)    =     104.33
Log likelihood = -619.37621                     Prob > chi2     =     0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
s0						
_cons	2.774104	.2680826	10.35	0.000	2.248671	3.299536
s1						
_cons	.2184732	.0377998	5.78	0.000	.144387	.2925595
xb						
group2	.8344981	.1712758	4.87	0.000	.4988036	1.170193
group3	1.61209	.1641745	9.82	0.000	1.290314	1.933866
_cons	-3.088731	.1713793	-18.02	0.000	-3.424628	-2.752834

```
Deviance =  1238.752 (686 observations.)
```

The syntax resembles that of `stcox`, but with the addition of two options: `df(2)` and `scale(hazard)`. The latter is self-explanatory. The `df()` option determines the complexity of the model for the baseline distribution. For now, we will take `df(2)` and return to that issue later.

The output from `stpm` is divided into several parts, with each part representing the different "equations" that have been estimated. Equations `s0`, `s1`, ... relate to modeling of the baseline distribution; we will return to them later. The final equation, `xb`, contains the usual regression coefficients expressed here as log-hazard ratios. Compare these coefficients with those from `stcox`:

```
. stcox group2 group3, nohr
  (output omitted)
```

_t _d	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
group2	.8399011	.1713929	4.90	0.000	.5039772	1.175825
group3	1.617692	.164544	9.83	0.000	1.295192	1.940193

The log hazard ratios and their standard errors are almost identical to those from `stpm`. Note that `stpm` also gives us an intercept term, `[xb]_cons` $= -3.088731$, which

in the Cox model is absorbed into the baseline hazard function and is not estimated explicitly.

`predict` may be used following `stpm` to provide (among several possibilities) an estimate of the hazard function, evaluated at each observed time and conditional on the covariates:

```
. predict haz, hazard
```

Figure 2 shows the behavior of this estimated hazard function according to time since diagnosis.

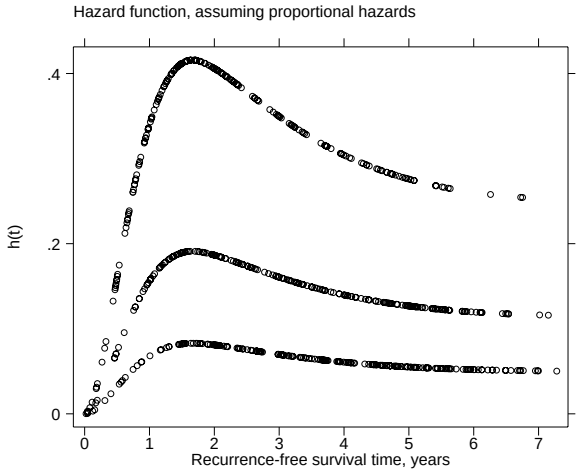

Figure 2: Hazard function for prognostic groups in the breast cancer data

The hazard function rises to a maximum of about 1.5 years after diagnosis, then steadily falls. To give some meaning to the numbers on the y-axis, note that the all-cause hazard of death for a female aged 85 years in the UK is approximately 0.1, rising to approximately 0.4 at age 100 years. Thus, the risk of recurrence or death for a breast cancer patient in the worst of our 3 prognostic groups is comparable with the force of mortality experienced by a woman of over 85 years old.

4.2 Comparison with Weibull model

The prototype PH parametric survival model is the Weibull (or the exponential, which is a Weibull with a constant hazard function, or equivalently a Weibull with shape parameter $p = 1$). With the Weibull model, the hazard function is proportional to t^{p-1}, where t is time. The hazard is therefore monotone in t. We will compare the Weibull with the PH spline model just presented. First we fit a Weibull model to the breast cancer data using Stata's `streg` command:

```
. streg group2 group3, dist(weibull)
  (output omitted)
Weibull regression -- log relative-hazard form
No. of subjects =          686                     Number of obs   =       686
No. of failures =          299
Time at risk    =  2111.978097
                                                   LR chi2(2)      =    122.53
Log likelihood  =   -638.45432                     Prob > chi2     =    0.0000
```

_t	Haz. Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
group2	2.331564	.3993457	4.94	0.000	1.666692	3.261666
group3	5.325107	.8746166	10.18	0.000	3.859434	7.347389
/ln_p	.3218311	.048409	6.65	0.000	.2269512	.4167111
p	1.379652	.0667876			1.254769	1.516964
1/p	.7248206	.0350879			.6592114	.7969597

The deviance is $-2 \times (-638.45) = 1276.9$, compared with 1238.8 for the parametric model, showing that the Weibull fits less well. Next, we will compare the log cumulative hazards estimated from the two models with the empirical log cumulative hazards from the Nelson–Aalen estimator:

```
. for num 1 2 3: predict sX if group==X, surv \ gen lnHweibX=ln(-ln(sX))
. * Fit stpm model and predict log cumulative hazard functions
. stpm group2 group3, df(2) scale(hazard)
  (output omitted)
. for num 1 2 3: predict lnHX if group==X, cumhaz
. for num 1 2 3: sts gen naX=na if group==X \ gen lnaX=ln(naX)
```

Figure 3 shows the resulting functions plotted against log time.

(*Graph on next page*)

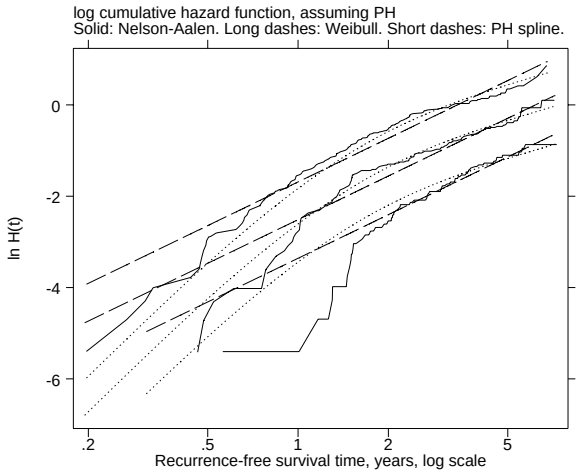

Figure 3: Log cumulative hazard functions for prognostic groups in the breast cancer data.

This graph shows several features. First, the proportional hazards assumption seems to break down in that the empirical log cumulative hazard functions do not appear parallel between the prognostic groups; the functions become closer together at large t. Second, there is curvature, which is not captured by the Weibull model. The curvature is somewhat better accommodated by the spline model, though far from perfectly, since the PH assumption is still imposed on the data. Thirdly, the empirical functions are highly variable for low values of t. This would be more evident if we were to show 95% pointwise confidence limits for them. Clearly, there is scope for finding models that fit the data better.

We have not shown the estimated hazard functions from the Weibull model. Since $p > 1$, they are monotone increasing and look extremely different from those shown in Figure 2.

4.3 Parametric proportional odds modeling

The command to fit a parametric proportional odds model to the breast cancer data is, for example,

```
. stpm group2 group3, df(2) scale(odds)
initial:       log likelihood =  -616.3553
rescale:       log likelihood =  -616.3553
rescale eq:    log likelihood =  -616.3553
Iteration 0:   log likelihood =  -616.3553
Iteration 1:   log likelihood = -615.49536
Iteration 2:   log likelihood = -615.49431
Iteration 3:   log likelihood = -615.49431
```

```
                                                Number of obs   =       686
                                                Wald chi2(2)    =    110.24
Log likelihood = -615.49431                     Prob > chi2     =    0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
s0						
_cons	2.914762	.2978938	9.78	0.000	2.330901	3.498623
s1						
_cons	.1910632	.0442298	4.32	0.000	.1043744	.2777519
xb						
group2	1.052285	.206176	5.10	0.000	.6481878	1.456383
group3	2.170996	.2093211	10.37	0.000	1.760734	2.581258
_cons	-3.451212	.2030715	-17.00	0.000	-3.849225	-3.053199

```
Deviance =  1230.989 (686 observations.)
```

The regression coefficients for `group2` and `group3` are now log-odds ratios. The deviance for the model is 1230.99, which is about 8 lower than for the PH model. These deviances cannot be converted to a significance test since the models are nonnested. Figure 4 compares the hazard function from the proportional odds model, computed using `predict`, with that from the PH model.

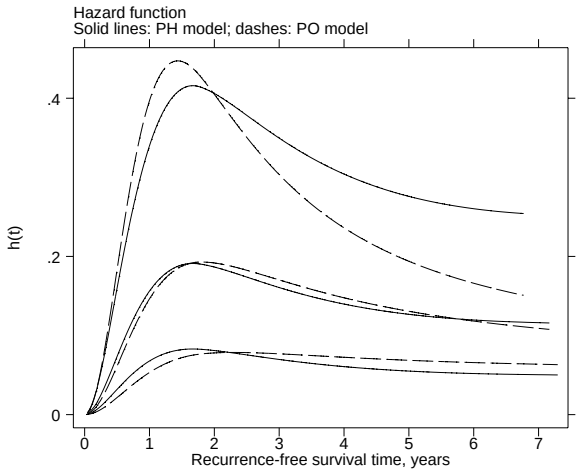

Figure 4: Hazard functions for PH and PO models in the breast cancer data.

The main difference between the hazard functions for the models is that the hazard for the Poor group eventually approaches that for the other two groups with the PO model, whereas with the PH model, a constant hazard ratio is maintained indefinitely.

4.4 Comparison with log-logistic model

The prototype PO parametric survival model is the log-logistic model. The hazard function is usually unimodal in t, though monotone functions are also possible. Here we will somewhat extend the earlier example. We will compare the estimated log cumulative hazard functions from the Nelson–Aalen estimator, the log-logistic model, and the PO spline model just presented. The required Stata instructions are as follows:

```
. streg group2 group3, dist(llogistic)
  (output omitted)
Log-logistic regression -- accelerated failure-time form

No. of subjects =          686                     Number of obs   =       686
No. of failures =          299
Time at risk    =  2111.978097
                                                   LR chi2(2)      =    129.65
Log likelihood  =    -625.5647                     Prob > chi2     =    0.0000

------------------------------------------------------------------------------
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      group2 |  -.6229638   .1201301    -5.19   0.000    -.8584144   -.3875132
      group3 |   -1.28827   .1182966   -10.89   0.000    -1.520127   -1.056413
       _cons |    2.13636   .0986964    21.65   0.000     1.942918    2.329801
-------------+----------------------------------------------------------------
     /ln_gam |  -.5622217   .0483756   -11.62   0.000    -.6570362   -.4674073
-------------+----------------------------------------------------------------
       gamma |   .5699414   .0275713                      .5183855    .6266248
------------------------------------------------------------------------------
```

The deviance is $-2 \times (-625.56) = 1251.1$, compared with 1231.0 for the PO spline model, showing that the log-logistic fits less well. Next, the log cumulative hazard functions:

```
. for num 1 2 3: predict sX if group==X, surv \ gen lnHllogX=ln(-ln(sX))
. * Fit stpm model and predict log cumulative hazard functions
. stpm group2 group3, df(2) scale(odds)
  (output omitted)
. for num 1 2 3: predict lnHX if group==X, cumhaz
. for num 1 2 3: sts gen naX=na if group==X \ gen lnaX=ln(naX)
```

Figure 5 shows the resulting functions plotted against log time.

(*Graph on next page*)

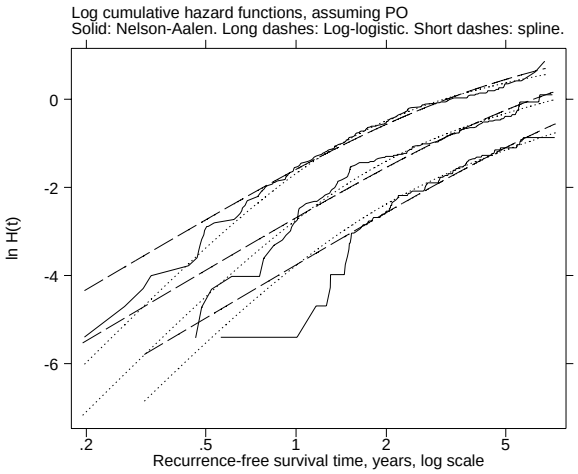

Figure 5: Log cumulative hazard functions for prognostic groups in the breast cancer data.

The spline model seems to fit better than the log-logistic at low t, but as already noted, the fit is hard to appraise in that region due to the high variance of the Nelson–Aalen estimate. Figure 6 illustrates this point.

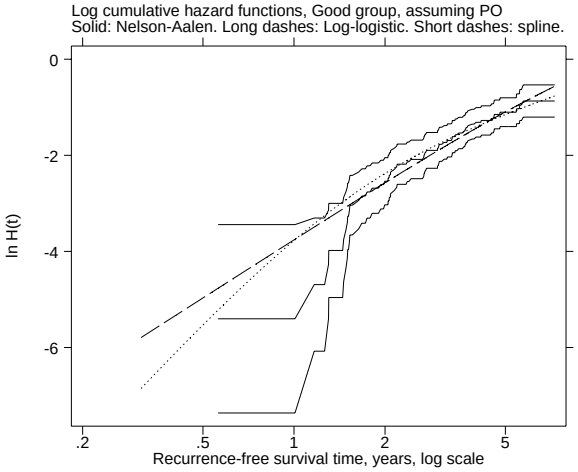

Figure 6: Log cumulative hazard functions and 95% CI for Nelson–Aalen, Good prognostic group.

The improvement in fit from the log-logistic to the spline model is subtle and is somewhat masked by the cumulative nature of the plot. Small differences in the cumulative hazard function may result in much larger differences in the hazard function (its first derivative), as illustrated in Figure 7.

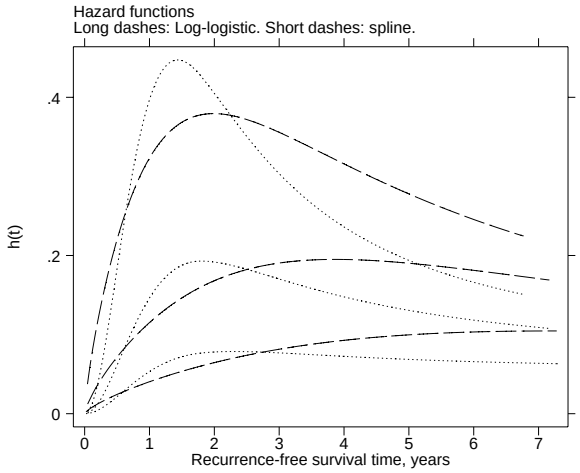

Figure 7: Hazard functions for prognostic groups, log-logistic and PO spline models.

The hazards estimated by the spline model are more peaked and diminish more rapidly than those from the log-logistic model. Confirmation of the pattern would require a substantial independent dataset.

5 Some methodological details

5.1 Proportional odds and proportional hazards models

The general proportional hazards model for survival data with a covariate vector $\mathbf{z}$ is defined through the hazard function $h(t;\mathbf{z})$ as

$$h(t;\mathbf{z}) = h_0(t)\exp(\beta'\mathbf{z})$$

where $h_0(t) = h(t;\mathbf{0})$ is the baseline hazard function. The model may be written in integrated form as

$$H(t;\mathbf{z}) = \left(\int_0^t h_0(u)\,du\right)\exp(\beta'\mathbf{z}) = H_0(t)\exp(\beta'\mathbf{z})$$

where $H(t;\mathbf{z})$ is the cumulative hazard function. By analogy, the general proportional (cumulative) odds model with covariate vector $\mathbf{z}$, see Bennett (1983), may be defined as

$$O(t;\mathbf{z}) = \frac{1-S(t;\mathbf{z})}{S(t;\mathbf{z})} = O_0(t)\exp(\beta'\mathbf{z})$$

where $O_0(t) = O(t; \mathbf{0})$ and $O(t; \mathbf{z})$ is the odds of an event occurring in $(0, t)$ for an individual with covariate vector $\mathbf{z}$. Covariates in the model act multiplicatively on the odds of an event, as with the more familiar logistic regression model for a binary outcome Y, where, for example, $O_0(t)$ is replaced with the baseline odds

$$\Pr(Y = 1 \,|\mathbf{z} = 0) / \Pr(Y = 0 \,|\mathbf{z} = 0)$$

5.2 Survival, density, and hazard functions

The approach used by Royston and Parmar (2001) to estimate the hazard, density, and survival functions is to smooth either the baseline cumulative odds function or the baseline cumulative hazard function. With the notation of the previous section, but for the time being suppressing $\mathbf{z}$, suppose that T is a survival-time random variable having a log-logistic distribution with location parameter μ and scale parameter σ. Let $x = \ln t$. We have

$$S(t) = \left\{1 + \exp\left(\frac{x - \ln \mu}{\sigma}\right)\right\}^{-1}$$

so that

$$\ln O(t) = \ln \frac{1 - S(t)}{S(t)} = \frac{x - \ln \mu}{\sigma}$$

is linearly related to x. If T has a distribution similar to a log-logistic, the log cumulative odds function will be curvilinearly related to x by a function $s = s(x)$. The survival, density, and hazard functions are then

$$\begin{aligned} S(t) &= (1 + \exp s)^{-1} \\ f(t) &= \frac{ds}{dt} \exp(s) (1 + \exp s)^{-2} \\ h(t) &= \frac{ds}{dt} \exp(s) (1 + \exp s)^{-1} \end{aligned}$$

Suppose now that T has a Weibull distribution with characteristic life μ and shape parameter p (or scale parameter $\sigma = p^{-1}$). Let the cumulative hazard function be $H(t) = -\ln S(t)$. Then

$$\ln H(t) = \ln\left\{\left(\frac{t}{\mu}\right)^p\right\} = px - p \ln \mu = \frac{x - \ln \mu}{\sigma}$$

which is linear in x. If T has a distribution similar to Weibull, then $\ln H(t)$ will again be curvilinearly related to x by a function s. The survival, density, and hazard functions are

$$\begin{aligned} S(t) &= \exp(-\exp s) \\ f(t) &= \frac{ds}{dt} \exp(s - \exp s) \\ h(t) &= \frac{ds}{dt} \exp(s) \end{aligned}$$

5.3 Spline-based parametric survival models

Since the distribution of survival times may be neither log-logistic nor Weibull, more flexible models are needed. The approach taken by Royston and Parmar (2001) is to model the logarithm of the baseline cumulative odds or hazard function as a natural cubic spline function of log time, so the general function $s(x)$ of the previous section is approximated by a spline. The PH spline model with fixed covariate vector $\mathbf{z}$ may be written

$$\ln\{-\ln S(t;\mathbf{z})\} = \ln H(t;\mathbf{z}) = \ln H_0(t) + \beta'\mathbf{z} = s(x) + \beta'\mathbf{z}$$

whereas for the PO spline model,

$$\ln\left\{S(t;\mathbf{z})^{-1} - 1\right\} = \ln O(t;\mathbf{z}) = \ln O_0(t) + \beta'\mathbf{z} = s(x) + \beta'\mathbf{z}$$

Therefore,

$$\left.\begin{array}{ll}\text{PH spline model:} & \ln H(t;\mathbf{z}) \\ \text{PO spline model:} & \ln O(t;\mathbf{z})\end{array}\right\} = s(x) + \beta'\mathbf{z}$$

Since $\ln\{-\ln S(t;\mathbf{z})\} = \ln H(t;\mathbf{z})$, a PH model may also be regarded as one in which the covariates act linearly on the complementary log-log probability of an event in $(0,t)$. Also, due to the nonlinear transformation of the log time scale, the metric for covariate effects with these models is the log (cumulative) hazard, complementary log-log probability, or log odds scale. An accelerated failure time interpretation is not available.

Natural cubic splines are defined as cubic splines constrained to be linear beyond boundary knots $k_{\min}, k_{\max}$. Such knots are usually, but not necessarily, placed at the extreme observed x-values. In addition, m internal knots $k_1 < \cdots < k_m$ with $k_1 > k_{\min}$ and $k_m < k_{\max}$ are specified. One can show that the natural cubic spline may be written as

$$s(x) = \gamma_0 + \gamma_1 x + \gamma_2 v_1(x) + \cdots + \gamma_{m+1} v_m(x)$$

where the jth basis function is defined for $j = 1, \ldots, m$ as

$$v_j(x) = (x-k_j)_+^3 - \lambda_j (x-k_{\min})_+^3 - (1-\lambda_j)(x-k_{\max})_+^3$$

and

$$\begin{aligned}\lambda_j &= \frac{k_{\max} - k_j}{k_{\max} - k_{\min}} \\ (x-a)_+^3 &= \max\left\{0, (x-a)^3\right\}\end{aligned}$$

The curve complexity is governed by the number of degrees of freedom (df), which ignoring γ_0 equals $m+1$. By convention, $m = 0$ is taken to mean that no internal and no boundary knots are specified. The straight line model $s(x) = \gamma_0 + \gamma_1 x$ with $\mathrm{df} = 1$ is then obtained.

5.4 Model extension

The model may be extended by allowing any of the coefficients $\gamma_1, \ldots, \gamma_m$ of the spline basis functions to depend on covariates (typically, subsets of $\mathbf{z}$). For example, consider PH models with no spline terms, i.e., Weibull models. The time-related component is $\gamma_1 \ln t$, where γ_1 is the Weibull shape parameter. By including covariates in γ_1, one can model variations in the shape parameter. If the covariates are categoric, one is effectively stratifying the model by them. To ensure that appropriate spline functions are obtained, it is important to include any such covariates in all of $\gamma_1, \ldots, \gamma_m$ and $\mathbf{z}$.

6 Implementation in stpm

As already mentioned, the `df()` option determines the complexity of the model for the baseline distribution. This is done via the number of knots chosen for the spline function $s(x)$ discussed above. With `df(1)`, the parent distribution is obtained (Weibull for `scale(hazard)`, log-logistic for `scale(odds)`). With `df(2)`, a single internal knot is placed at the median of the uncensored log survival times. Alternatively, the `knots()` option may be used to choose knot positions manually, either directly as log times, for example `knots(-0.5 0.2 0.8)`, or indirectly as centile positions, e.g., `knots(%50 70 80)`.

Each of the coefficients $\gamma_1, \ldots, \gamma_{m+1}$ is associated with an equation in the output from `stpm`. The constant term γ_0 is represented by coefficient `[xb]_b[_cons]` in equation `[xb]`, and covariates $\mathbf{z}$ act at this level of the model. γ_1 is represented by `[s0]_b[_cons]`, γ_2 by `[s1]_b[_cons]`, and so on.

Extensions to the model are handled using the `stratify()` option. For example, to fit a separate 2 df spline model to the log cumulative odds function at each level of the variable `group` in the breast cancer dataset, one could enter

```
. stpm group2 group3, df(2) scale(odds) stratify(group2 group3)
initial:       log likelihood =  -616.3553
rescale:       log likelihood =  -616.3553
rescale eq:    log likelihood =  -616.3553
Iteration 0:   log likelihood =  -616.3553
Iteration 1:   log likelihood = -612.80513
Iteration 2:   log likelihood = -612.62289
Iteration 3:   log likelihood = -612.62274
Iteration 4:   log likelihood = -612.62274
                                                Number of obs   =        686
                                                Wald chi2(2)    =       3.18
Log likelihood = -612.62274                     Prob > chi2     =     0.2035
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
s0						
group2	-2.443586	1.70277	-1.44	0.151	-5.780955	.8937817
group3	-2.859845	1.646575	-1.74	0.082	-6.087074	.3673827
_cons	5.583426	1.605347	3.48	0.001	2.437003	8.729849
s1						
group2	-.2769887	.221613	-1.25	0.211	-.7113421	.1573647
group3	-.3284494	.2154241	-1.52	0.127	-.7506728	.093774
_cons	.5036676	.2068428	2.44	0.015	.0982631	.9090721
xb						
group2	1.683409	.436123	3.86	0.000	.8286242	2.538195
group3	2.8153	.4185716	6.73	0.000	1.994915	3.635686
_cons	-4.045346	.3831531	-10.56	0.000	-4.796313	-3.29438

```
Deviance =  1225.245 (686 observations.)
```

The coefficients γ_1 and γ_2, represented by equations `[s0]` and `[s1]`, respectively, now depend on the prognostic group. For example, the log cumulative odds function in the Medium group (`group2==1`) is estimated as

$$(-4.045 + 1.683) + (5.583 - 2.444)\, x + (0.504 - 0.277)\, v_1\,(x)$$

The term stratify is perhaps slightly misleading, since the concept is more general than that applied in the context of the Cox model. The covariates in `stratify()` are not required to be categoric. They could be continuous, e.g., age. If you wish to indicate a categoric variable with more than 2 levels, you could use the `xi:` prefix. For instance,

```
. xi: stpm i.group, df(2) scale(odds) stratify(i.group)
```

7 Model selection

Royston and Parmar (2001) suggest selecting the df for the spline part of the model by minimizing the Akaike Information Criterion (AIC). The AIC is defined as the deviance (i.e., -2 times the log likelihood) plus twice the number of model parameters, and is stored by `stpm` in the post-estimation scalar `e(aic)`. The AIC may also be used to select the scale for the model. For example, Table 1 shows the AIC for the breast cancer data for PH and PO models with between 1 and 6 df (0 and 5 knots), using default knot positions provided by `stpm`. The values in the table were obtained by using the following commands:

```
. for num 1/6, nohead: quietly stpm group2 group3, scale(hazard) df(X)\
 display X, e(aic)
1 1284.9086
2 1248.7524
3 1248.0013
4 1247.0252
5 1249.0723
6 1250.6245
```

```
. for num 1/6, nohead: quietly stpm group2 group3, scale(odds) df(X)\
 display X, e(aic)
1 1259.1294
2 1240.9886
3 1241.521
4 1240.5969
5 1242.6683
6 1244.0632
```

Table 1: AIC values for several spline survival models for the breast cancer data.

df	No. of knots	AIC (PH model)	AIC (PO model)
1	0	1284.9	1259.1
2	1	1248.8	1241.0
3	2	1248.0	1241.5
4	3	1247.0	1240.6
5	4	1249.1	1242.7
6	5	1250.6	1244.1

The model minimizing the AIC is the PO model with 3 knots, for which AIC = 1240.6. However, the more parsimonious model with 1 knot has almost as low an AIC (1241.0), and is to be preferred since the evidence favoring greater complexity is weak.

A large topic is the problem of model selection when there are many candidate predictors, some of which may be continuous. There is a convenient pragmatic approximation available that eases the mechanics of searching many such models within the `stpm` framework. It turns out that the regression coefficients for the covariates `z` and the functional forms of continuous covariates are largely robust to misspecification of the baseline distribution function. Therefore, the process of choosing `z` for PH models may be explored within the Cox framework using `stcox`, which in Stata runs fast. Similarly, it turns out that PO scale models may be investigated within the log-logistic model (which in `stpm` is obtained by specifying `scale(odds) df(1)`). This is no great advantage since Stata's `streg` command, which also implements the log-logistic model, is not particularly fast. However, the log-logistic distribution is not dissimilar to the lognormal distribution, and Stata's `cnreg` command for estimating a censored (log)normal regression model *is* fast. Therefore, covariate exploration may be performed using `cnreg`. The final model will be fitted using `stpm` with `scale(odds)` and as many knots as are indicated by the AIC.

8 Knot selection

The placement of the internal knots is an issue. Royston and Parmar (2001) chose to place boundary knots at the extreme uncensored log survival times, and internal knots at positions given in the help file for `stpm`. Following Durrleman and Simon (1989), the knots are placed at predefined percentiles of the uncensored log survival times.

Experience so far suggests that models with df > 4 are required infrequently. Models with df > 6 are not entertained since the resulting curves are likely to be unreliable.

While the default knot positions may be fine in many cases, the choices are predefined and can always be improved on to some extent. The aim of the present section is to suggest a simple strategy to check whether any vastly better knot position(s) are available with the given data. One might quite reasonably be tempted to regard knot positions as parameters and attempt to estimate them by maximum likelihood, but except for the case of one knot, this is not in general practicable since the likelihood surface may be multimodal. An alternative is to assign knots at random positions and evaluate the likelihood of each resulting model. Random knot positioning may be achieved by using the `knots(u ...)` option of `stpm` (`u` signifying a random uniform distribution). The procedure may be repeated say 50 or 100 times, with the model with the lowest deviance being regarded as the "best" among those entertained. The resulting AIC can be adjusted approximately by adding two for each knot position thus "estimated".

Table 2 shows an example of this procedure applied 100 times to the breast cancer data.

Table 2: AIC values for several proportional odds models for the breast cancer data. The default knot placement is compared with partially optimized knot positions.

No. of knots	Centiles (default)	AIC (default)	Centiles ('Best')	AIC ('Best')
1	50	1241.0	21	1242.5
2	33, 67	1241.5	72, 75	1242.3
3	25, 50, 75	1240.6	36, 41, 58	1245.3
4	20, 40, 60, 80	1242.7	11, 34, 46, 48	1246.2

Clearly, there is nothing to be gained in this example from attempting to optimize the knot positions, since increasing the model dimension always increases the AIC. The default model with one knot at the 50th centile remains the most satisfactory choice.

Figure 8 compares the baseline hazard functions (i.e., the hazards in the Good group) from the preferred model with one knot and the model with four "optimized" knots at the 11, 34, 46 and 48 centiles.

(*Graph on next page*)

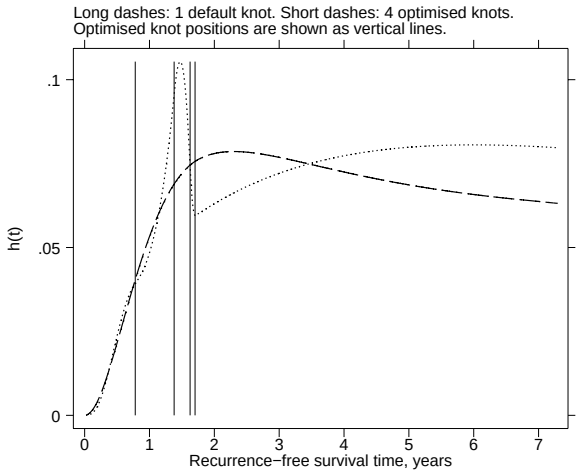

Figure 8: Hazard functions with one default and four "optimized" knots

The optimized hazard function looks implausible and quite unstable, with sharp corners at about 1.4 and 1.8 years.

9 Prediction

9.1 Basics

Prediction (additional estimation) following `stpm` is performed by using `predict`. The syntax is

`predict` [*type*] *newvarname* [`if` *exp*] [`in` *range*] [`,` *statistic*
`at(`*varname* | #|*vn* [*varname* | #|*vn* ...]`) noconstant nooffset stdp`
`time(`#|*vn*`) zero`]

where

statistic	Result
`xb`	index (linear predictor)
`cumodds`	log cumulative odds function
`cumhazard`	log cumulative hazard function
`normal`	normal deviate function
`spline`	fitted spline function
`dzdy`	derivative of fitted spline function with respect to ln(`_t`)
`density`	density function
`hazard`	hazard function
`survival`	survival function
`centile(`#`)`	#th centile of survival time distribution
`tvc(`*varname*`)`	time-varying coefficient for *varname*

`tvc(`*varname*`)` stands for "time-varying coefficient" and computes the estimated coefficient for *varname*, a covariate in `stpm`'s *varlist*. If *varname* is time fixed, then *newvarname* will be a constant. If *varname* is included in *strat_varlist*, then *newvarname* will depend on `_t`, and may be interpreted as the time-varying effect of *varname* on the chosen scale of the model (proportional hazards, proportional odds or probit). For example, in a hazards-scale model (`scale(hazard)`), *newvarname* multiplied by *varname* will give an estimate of the time-varying log cumulative hazard ratio for *varname* (compared with *varname* = 0) at every observed value of *varname*. *newvarname* alone will give the log cumulative hazard ratio for a one-unit change in *varname*. Note that the time-varying log cumulative hazard ratio for *varname* will not be identical to the time-varying log hazard ratio for *varname*.

Prediction is conditional on the observed values of the covariates `z`, unless particular values of the covariates are specified by using the `at()` or `zero` options. Similarly, predictions are evaluated at values of the `st` time variable `_t`, unless particular time values are specified by using the `time()` option. All statistics are available both in and out of sample; type `predict ... if e(sample)` if prediction is wanted only for the estimation sample. The default is linear prediction of the covariate part of the model, i.e., for statistic `xb`. You can predict any of the above statistics for any `scale()` model.

9.2 Options for predict

`at(`*varname* #|*vn* ...`)` computes the various statistics at value(s) (# or *vn*)... of model covariates *varname*..., where *vn* means "variable name". The `at()` option is a convenient way of specifying out-of-sample prediction for some or all of the covariates in the model. Covariates in `stpm`'s *varlist* that are not listed in `at()` are used in computing predicted values, unless the `zero` option is specified, in which case adjustment is to value 0 of such predictors.

`noconstant` is relevant only if you specified `predict, xb`. It removes the constant (if any) in equation *xb*.

`nooffset` is relevant only if you specified `offset()` for `stpm`. It modifies the calculations made by `predict, xb` so that they ignore the offset variable.

`stdp` computes the standard error of statistics `xb`, `cumhazard`, `cumodds`, or `normal`, or of the log survival time for `centile()`. `stdp` is not implemented for other statistics, but note that confidence intervals for the survival function may be found by back-transformation of confidence intervals for the cumulative hazard or odds or normal function.

`time(`#|*vn*`)` predicts at time # or at the time values in variable *vn*. If `time()` is not specified, prediction is at time `_t`.

`zero` predicts at zero values of covariates in *varlist* and similarly for *strat_varlist*, if `stratify()` is specified. See also option `at()`.

10 More on hazards

Suppose, for example, you wished to predict the hazard function for group 3 (i.e., for `group3==1 & group 2==0`) from a PO model for the breast cancer dataset. Having already executed `stpm`, you would enter

```
. predict haz3, at(group2 0 group3 1)
```

Note that prediction with the `at()` option is across the whole sample. Thus, in the above example, `haz3` will be the hazard function for group 3 evaluated at every observation time, irrespective of the values of `group2` and `group3` for those observations. This feature is useful for computing the hazard ratio between covariate levels. For example,

```
. predict haz1, at(group2 0 group3 0)
. predict haz2, at(group2 1 group3 0)
. predict haz3, at(group2 0 group3 1)
. gen hr2 = haz2/haz1
. gen hr3 = haz3/haz1
```

By using the `at()` option, the hazard functions `haz1`, `haz2` and `haz3` are defined for all observations, enabling the computation of hazard ratios `hr2` and `hr3`. Use of `if` to restrict calculation to each group separately will not work, since the resulting hazards will be defined for nonoverlapping subsets of the data.

It is interesting to compare estimates of the baseline hazard function from `stcox` and `stpm`. Unfortunately, a "proper" estimate of the baseline hazard function is not provided directly by `stcox`, but it may be computed by numerical differentiation of the cumulative hazard function (generated by option `basechazard()`). The simplest way to compute the baseline hazard function following `stpm` is via the `zero` option of `predict`:

```
. predict h0, hazard zero
```

Figure 9 compares estimates of the baseline hazard function calculated from `stpm` and by numerical differentiation following `stcox`.

(*Graph on next page*)

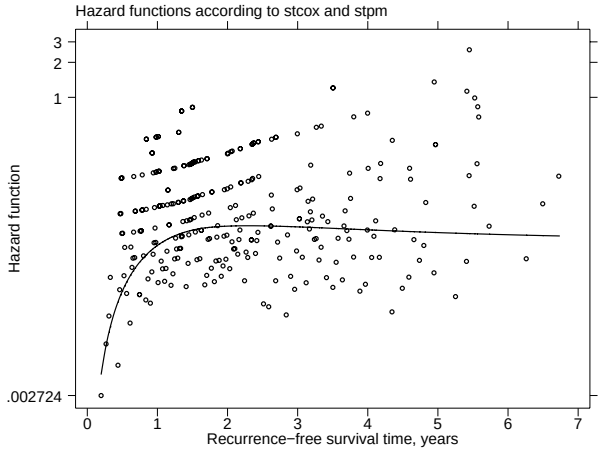

Figure 9: Estimates of the baseline hazard function by `stcox` and `stpm`.

Notice how extremely noisy the `stcox`-estimated function is. Effectively, an estimate of the hazard function is made for every non-censored observation time. Plotting on a logarithmic y-axis makes it easier to see the shape of the function.

10.1 Centiles of the survival-time distribution

Another useful feature of `predict` is to compute centiles and standard errors of the survival-time distribution. For example, one might want the median (50th centile) together with 95% confidence intervals according to values of covariates. We could obtain these for the breast cancer data as follows:

```
. predict median, centile(50)
. predict sem, centile(50) stdp
. gen lci = exp(ln(median)-1.96*sem)
. gen uci = exp(ln(median)+1.96*sem)
```

Note that `predict` estimated the 50th centile on the time scale but that the standard error is on the log time scale. The results are shown in Figure 10.

(*Graph on next page*)

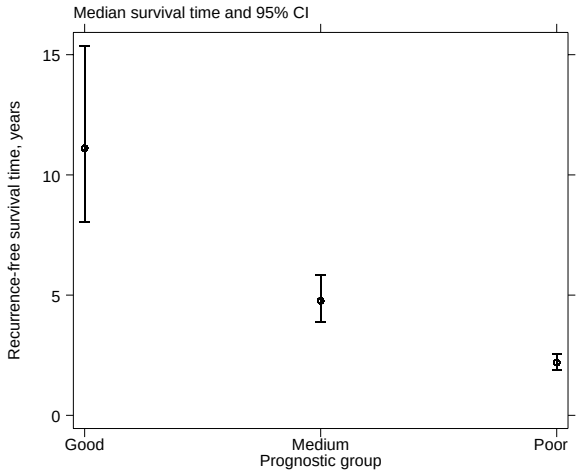

Figure 10: Median recurrence-free survival time and 95% confidence intervals by prognostic group.

Although the Kaplan–Meier survival curve for the Good group does not reach 0.5 during the observation period (see Figure 1), the model has extrapolated a median survival time of 11 years, with a very wide confidence interval of $(8, 15)$ years. Depending on your point of view, such extrapolation may be seen as a weakness or a strength of a parametric approach to modeling!

11 Other aspects

Interval-censored data is not infrequently encountered in practice. An event is known to have happened within a certain time interval, but exactly when it occurred is not known. An example in cancer studies is when patients are followed up for an event such as disease recurrence or progression. The patient may be found to have progressed at a scheduled follow-up visit, but the recurrence may have occurred at some unknown point after the previous visit. `stpm` handles such data by way of the `left()` option, which specifies the left-hand endpoint of the intervals.

The `stratify()` option may be used to assess the PH or PO assumption for a dataset. Details of the principle are given by Royston and Parmar (2001). Rather than describe this aspect in detail here, the topic will be deferred to a later article in which a separate command, `stsctest`, to test for appropriateness of scale will be introduced.

Estimation of confidence intervals for the hazard function or hazard ratio is another topic that will be deferred to a later article. An ado-file called `stpmhaz` is under development that will use the bootstrap to provide confidence intervals for hazard and density functions and for hazard ratios.

Models based on the lognormal distribution and its generalization by using spline

functions are also implemented in `stpm`. These models may be seen as the survival analysis version of logistic regression models with a probit link (the `probit` command in Stata). Sometimes such models are useful, but they will not be described further here.

12 Conclusion

The Cox model, with its associated machinery and extensions, remains a vital tool for the analysis of censored survival data. I have tried to show that other approaches are available which can throw light on additional important aspects of the data, aspects the Cox model is not designed to examine. In particular, they can enable the analyst to model the hazard function flexibly and they can provide smooth curves to approximate more or less any desired baseline distribution function. I believe that such models are well worth further study and use by practitioners, and I have therefore made them available in Stata.

13 References

Aranda-Ordaz, F. J. 1981. On two families of transformations to additivity for binary response data. *Biometrika* 68: 357–363.

Bennett, S. 1983. Analysis of survival data by the proportional odds model. *Statistics in Medicine* 2: 273–277.

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

Efron, B. 1988. Logistic regression, survival analysis and the Kaplan–Meier curve. *Journal of the American Statistiscal Association* 83: 414–425.

Gelfand, A. E., S. K. Ghosh, C. Christiansen, S. B. Soumerai, and T. J. McLaughlin. 2000. Proportional hazards models: a latent competing risk approach. *Applied Statistics* 49: 385–397.

Hastie, T. J. and R. J. Tibshirani. 1993. Varying-coefficient models (with discussion). *Journal of the Royal Statistical Society (Series B)* 55: 757–796.

Hess, K. 1994. Assessing time-by-covariate interactions in proportional hazards regression models using cubic spline models. *Statistics in Medicine* 13: 1045–1062.

Hjort, N. L. 1992. On inference in parametric survival data models. *International Statistical Review* 60: 355–387.

Kooperberg, C., C. J. Stone, and Y. K. Truong. 1995. Hazard regression. *Journal of the American Statistical Association* 90: 78–94.

Mackenzie, G. 1996. Regression models for survival data: the generalized time-dependent logistic family. *The Statistician* 45: 21–34.

—. 1997. On a non-proportional hazards regression model for repeated medical random counts. *Statistics in Medicine* 16: 1831–1843.

Rosenberg, P. S. 1995. Hazard function estimation using B-splines. *Biometrics* 51: 874–887.

Rossini, A. J. and A. A. Tsiatis. 1996. A semiparametric proportional odds model for the analysis of current status data. *Journal of the American Statistical Association* 91: 713–721.

Royston, P. and M. K. B. Parmar. 2001. Flexible parametric models for censored survival data, with application to prognostic modelling and estimation of treatment effects. *Statistics in Medicine*. In press.

Sauerbrei, W. and P. Royston. 1999. Building multivariable prognostic and diagnostic models: transformation of the predictors using fractional polynomials. *Journal of the Royal Statistical Society, Series A* 162: 71–94.

Shen, X. T. 1998. Proportional odds regression and sieve maximum likelihood estimation. *Biometrika* 85: 165–177.

Yang, S. and R. L. Prentice. 1999. Semiparametric inference in the proportional odds regression model. *Journal of the American Statistical Association* 94: 125–136.

Younes, N. and J. Lachin. 1997. Link-based models for survival data with interval and continuous time censoring. *Biometrics* 53: 1199–1211.

About the Author

Patrick Royston is a medical statistician of 25 years of experience, with a strong interest in biostatistical methodology and in statistical computing and algorithms. At present he works in clinical trials and related research issues in cancer. Currently he is focusing on problems of model building and validation with survival data, including prognostic factors studies, on parametric modeling of survival data, and on novel trial designs.