Journal of Statistical Software

October 2014, Volume 61, Code Snippet 1. *http://www.jstatsoft.org/*

Tutorial: Survival Estimation for Cox Regression Models with Time-Varying Coefficients Using SAS and R

Laine Thomas
Duke University

Eric M. Reyes
Rose-Hulman Institute of Technology

Abstract

Survival estimates are an essential compliment to multivariable regression models for time-to-event data, both for prediction and illustration of covariate effects. They are easily obtained under the Cox proportional-hazards model. In populations defined by an initial, acute event, like myocardial infarction, or in studies with long-term follow-up, the proportional-hazards assumption of constant hazard ratios is frequently violated. One alternative is to fit an interaction between covariates and a prespecified function of time, implemented as a time-dependent covariate. This effectively creates a time-varying coefficient that is easily estimated in software such as SAS and R. However, the usual programming statements for survival estimation are not directly applicable. Unique data manipulation and syntax is required, but is not well documented for either software. This paper offers a tutorial in survival estimation for the time-varying coefficient model, implemented in SAS and R. We provide a macro `coxtvc` to facilitate estimation in SAS where the current functionality is more limited. The macro is validated in simulated data and illustrated in an application.

Keywords: time-dependent covariates, time-varying coefficients, Cox proportional-hazards model, survival estimation, SAS, R.

1. Introduction

Clinical studies with long-term follow-up regularly measure time-to-event outcomes, such as survival time, for which multivariable models are used to identify covariate associations and make predictions. The most common regression modeling framework is the Cox proportional-hazards model. The name implies the restrictive assumption of constant hazard ratios over time, though Cox proposed a simple extension in which covariates are allowed to vary ac-

cording to a pre-defined function of time (Cox 1972). This time-varying coefficient model is implemented in SAS (SAS Institute Inc. 2008) and R (R Core Team 2014) via the inclusion of an appropriately constructed time-dependent covariate. Due to its simplicity and available software, this approach is widely used in medical literature to account for violations of proportional-hazards (Buchholz and Sauerbrei 2011; Natarajan *et al.* 2009; Guanghui and Schaubel 2008; Gao, Grunwald, Rumsfeld, Schooley, MacKenzie, and Shroyer 2006). The forgoing articles do not provide survival estimates despite the fact that these are a standard complement to proportional-hazards regression. This is likely due to uncertainty about the capacity of statistical software for the time-varying coefficient model.

Allison (2010) provides an excellent review of alternative ways to implement time-dependent covariates and time-varying coefficients in SAS. As usual, survival estimation can be requested by the `baseline` statement in `proc phreg` (SAS Institute Inc. 2010), but the log contains a warning:

```
NOTE: Since the counting process style of response was specified
in the MODEL statement, the SURVIVAL = statistics in the baseline
statement should be used with caution.
```

Allison (2010) comments that "For most applications, however, this should not be a concern." We aim to provide more explicit guidance. Specifically, SAS Institute Inc. (2010) provides two alternative survival estimators in `proc phreg`: the product-limit and the empirical cumulative hazard. The documented equations are written with time-constant coefficients, though the latter approach can easily substitute time-dependent covariates or time-varying coefficients. Compared to the documentation, a less flexible simplification of the empirical cumulative hazard estimator is actually implemented by the `baseline` statement and, given a time-varying coefficient model, is only applicable to the reference individual with all covariates equal to zero. Subsequently, we describe additional programming statements that can be used to obtain estimates for any set of covariates.

The **survival** (Therneau 2014) package in R has functions, `coxph` and `survfit`, that will produce survival estimates in the presence of time-varying coefficients. However, the function inputs and data need to be carefully structured. Documentation with examples for this topic is sparse. Fox and Weisberg (2011) provides a related example for time-dependent covariates using the `fold` function for data manipulation. This function is a bit cumbersome and we prefer `survSplit` as demonstrated below. Other approaches to flexible survival modeling are available in R. Martinussen and Scheike (2006) provide an especially comprehensive review of "Dynamic Regression Models for Survival Data," with a corresponding **timereg** (Scheike and Zhang 2011) package. This includes a function, `timecox` for fitting an extended version of the Cox model with unspecified, smooth, time-varying coefficients. A resampling algorithm for estimating survival is provided with examples, but not incorporated into the function (Chapter 6). This provides a flexible alternative, but is not required for the present goal of allowing coefficients to vary according to a pre-defined function of time.

We aim to exemplify the utility of the SAS and R softwares for survival estimation in the time-varying coefficient model and provide SAS macros to facilitate this process. In Section 2, we review survival estimation in various generalizations of the Cox model that fall under the classification of multiplicative hazard models. In Section 3, we exemplify the syntax required to obtain appropriate estimates in SAS and R, including data manipulation which is facilitated by a simple SAS macro `cpdata`. The SAS macro `coxtvc` is introduced in Section 4

and illustrated in Section 5. In Section 6, we discuss the utility of the two softwares and highlight potential applications that may be overlooked because of the current uncertainty about existing software.

2. Survival estimation in multiplicative hazard models

2.1. Cox proportional-hazards model

Cox proportional-hazards regression (Cox 1972) is thoroughly described elsewhere (Therneau and Grambsch 2000; Kalbfleisch and Prentice 2002; Klein and Moeschberger 2003; Harrell 2006; Allison 2010). Here we provide a short background that may facilitate the discussion of survival estimation. In his 1972 paper, Cox introduced two key ideas: a simple model for the relationship between covariates and the hazard of experiencing an event, and a partial-likelihood approach to estimate the model parameters. For subjects $i = 1, \ldots, n$, let T_i denote the failure time, C_i denote the censoring time, and $N_i(t)$ represent a counting process such that $N_i(t) = I(T_i >= t)$, where $I(u)$ is the indicator function taking value 1 if event u occurs and 0 otherwise. A subject is at risk until they experience an event or are censored. $Y_i(t)$ indicates whether the ith subject is at risk at time t, i.e., $Y_i(t) = I\{\min(T_i, C_i) < t\}$. Let X_i denote a predictor of interest; and $\boldsymbol{Z}_i$ a $(p \times 1)$ vector of additional covariates, where T_i and C_i are independent given X_i and $\boldsymbol{Z}_i$. The failure time T_i is not available for all subjects, but instead $\min(T_i, C_i)$ and $\delta_i = I(T_i \leq C_i)$ are observed. The hazard of failure $\lambda(t|X, \boldsymbol{Z})$ is related to the covariates by

$$\begin{aligned} \lambda(t|X, \boldsymbol{Z}) &= \lim_{h \to 0^+} \left\{ h^{-1} \mathsf{P}(t \leq T \leq t + h | T \geq t, X, \boldsymbol{Z}) \right\} \\ &= \lambda_0(t) \exp(\beta X + \boldsymbol{\beta}_Z^\top \boldsymbol{Z}), \end{aligned} \tag{1}$$

where $\lambda_0(t)$ is an unspecified baseline hazard function for the reference subject with all covariates equal to 0. The effect of a one unit increase in X, given common covariates $\boldsymbol{Z}$, is measured by the hazard ratio $\lambda(t|X = x + 1, \boldsymbol{Z})/\lambda(t|X = x, \boldsymbol{Z}) = \exp(\beta)$. This does not depend on t, reflecting the "proportional-hazards" assumption of a constant hazard ratio over time. The coefficients in Equation 1 can be estimated from a partial-likelihood in which the unknown baseline hazard drops out, leaving a function of coefficients and observed data, free from assumptions on the distribution of event times (Cox 1972). In the following, we drop the covariates $\boldsymbol{Z}$ to simplify notation.

Hazard ratios alone do not provide a complete picture of longitudinal survival. Survival estimates are a standard complement. The survival function, $S(t|X) = \mathsf{P}(T > t|X)$, is related to the hazard by $S(t|X) = \exp\{-\Lambda(t|X)\}$, where $\Lambda(t|X) = \int_0^t \lambda(u|X)du$ denotes the cumulative hazard. This relationship holds regardless of the particular model for the hazard. However, under the Cox proportional-hazards model, the cumulative hazard has a convenient simplification:

$$\begin{aligned} \Lambda(t|X) &= \int_0^t \lambda_0(u) \exp(\beta X) du \\ &= \exp(\beta X) \Lambda_0(t), \end{aligned} \tag{2}$$

where $\Lambda_0(t) = \int_0^t \lambda_0(u)du$. A consistent estimator of $\Lambda_0(t)$ can be used along with $\hat{\beta}$ to estimate survival. A familiar estimator, available in SAS and R, is the empirical cumulative

hazard function,

$$\hat{\Lambda}_0(t) = \sum_{i=1}^{n} \int_0^t \frac{dN_i(u)}{\sum_j Y_j(u) \exp(\hat{\beta} X_j)}.$$

Let x^* denote a particular value of X. Substituting estimators for the unknown quantities we obtain the empirical cumulative hazard estimator given x^*,

$$\begin{aligned} \hat{\Lambda}(t|x^*) &= \exp(\hat{\beta} x^*) \hat{\Lambda}_0(t) \\ &= \sum_{i=1}^{n} \int_0^t \frac{\exp(\hat{\beta} x^*) dN_i(u)}{\sum_j Y_j(u) \exp(\hat{\beta} X_j)}, \end{aligned}$$

and the corresponding estimated survival probability at time t is

$$\hat{S}(t|x^*) = \exp\{-\exp(\hat{\beta} x^*) \hat{\Lambda}_0(t)\}. \tag{3}$$

This equation is computationally convenient. A single integral must be calculated to estimate the baseline hazard $\hat{\Lambda}_0(t)$, and survival probabilities for any value of x^* are then obtained by scalar multiplication and exponentiation.

2.2. Cox model with time-dependent covariates

Suppose that updated values of X are observed over time. This is referred to as a time-dependent covariate, denoted by $X(t)$. Let $x^*(t)$ be a known function, specifying a particular set of values over time. For example, in a model for 5 year mortality, where time is measured in years and $X(t)$ denotes the occurrence of surgery prior to time t, $x^*(t) = I(t > 1)$ describes the covariate function for an individual who has surgery at 1 year. The Cox model is easily generalized to allow time-dependent covariates. Since the hazard is conditional on time t, the relationship to $X(t)$ is straightforward:

$$\lambda\{t|X(t)\} = \lambda_0(t) \exp\{\beta X(t)\}. \tag{4}$$

As before, the probability of survival is $S(t|X) = \exp\left[-\Lambda\{t|X(t)\}\right]$. However,

$$\Lambda\{t|X(t)\} = \int_0^t \lambda_0(t) \exp\{\beta X(u)\} du,$$

and the term $\exp\{\beta X(u)\}$ does not factor out of the integral, as in Equation 2. The empirical cumulative hazard estimator, given a particular covariate trajectory $x^*(t)$, is

$$\hat{\Lambda}\{t|x^*(t)\} = \sum_{i=1}^{n} \int_0^t \frac{\exp\{\hat{\beta} x^*(u)\} dN_i(u)}{\sum_j Y_j(u) \exp\{\hat{\beta} X_j(u)\}}, \tag{5}$$

and the corresponding estimated survivor function is,

$$\hat{S}\{t|x^*(t)\} = \exp\left[-\hat{\Lambda}\{t|x^*(t)\}\right]. \tag{6}$$

Note that the estimate $\hat{S}\{t|x^*(t)\}$ does *not* simplify to $\exp\left[-\exp\{\hat{\beta} x^*(t) \hat{\Lambda}_0(t)\}\right]$, as it did in the case of time-invariant covariates. A unique integration is required to estimate $\hat{\Lambda}\{t|x^*(t)\}$

for every value of $x^*(t)$, rather than scalar multiplication of $\hat{\Lambda}_0(t)$. This is more computationally burdensome. Only when interest is focused on a time invariant value, i.e., $x^*(t) = x^*$, will the survival estimate have the simplified form of Equation 3.

In the presence of time-dependent covariates, it may not make sense to calculate survival probabilities or predictions. This requires knowledge of $X(t)$, which may be unknown until time t, at which point its observation frequently implies survival. In the above example, individuals who have surgery at 1 year are alive at 1 year, by definition. Hence, survival estimation is rarely implemented in this case. There are exceptions, where it is conceptually reasonable to calculate survival, conditioned on a pre-determined trajectory of time-dependent covariates $x^*(t)$, $0 < t < \infty$. Generally, such exceptions fall under the class of *exogenous* covariates, that occur according to a mechanism external to the individual subject; seasonal effects, for example. In our experience, a more common application is the use of a time-dependent covariate to implement a model with a time-varying coefficient. For the remainder of this article we focus on the case of time-varying coefficients.

2.3. Time-varying coefficients

Both models for the hazard, given in Equations 1 and 4, involve the proportional-hazards assumption of constant covariate effects. Standard survival texts include multiple options for testing this assumption and addressing violations (Allison 2010; Harrell 2006; Martinussen and Scheike 2006). One approach is to include an interaction with a deterministic function of time. The hazard of failure $\lambda(t|X)$ is related to the covariates by

$$\lambda(t|X) = \lambda_0(t) \exp\{g(\boldsymbol{\beta}, t)X\},$$

where $\boldsymbol{\beta}$ is a vector of coefficients and $g(\boldsymbol{\beta}, t)$ is a function of time that is specified by the analyst. For example, when the hazard is assumed to change by a factor of log time $\boldsymbol{\beta} = (\beta_1, \beta_2)$ and $g(\boldsymbol{\beta}, t) = \beta_1 + \beta_2 \log(t)$, corresponding to a hazard of $\lambda_0(t) \exp\{\beta_1 X + \beta_2 \log(t)X\}$. Alternatively, the coefficient may be piecewise constant; before and after $t^{'}$. Then, $g(\boldsymbol{\beta}, t) = \beta_1 I(t < t^{'}) + \beta_2 I(t \geq t^{'})$ and the hazard is $\lambda_0(t) \exp\{\beta_1 I(t < t^{'})X + \beta_2 I(t \geq t^{'})X\}$. The benefit of this approach is that changes in the hazard over time are summarized by a simple, interpretable equation.

Generally, $g(\boldsymbol{\beta}, t)$ is a simple function such that $g(\boldsymbol{\beta}, t) = \boldsymbol{\beta}^\top \boldsymbol{G}(t)$, where $\boldsymbol{G}(t) = \{g_1(t), g_2(t), \ldots\}$. In the first example above, $\boldsymbol{G}(t) = \{1, \log(t)\}$. Consequently, the hazard can be factored into

$$\begin{aligned} \lambda(t|X) &= \lambda_0(t) \exp\{\boldsymbol{\beta}^\top \boldsymbol{G}(t)X\} \\ &= \lambda_0(t) \exp\{\boldsymbol{\beta}^\top \boldsymbol{X}(t)\}, \end{aligned}$$

where $\boldsymbol{X}(t) = \{\boldsymbol{G}(t)X\}$. This formulation makes it clear that the time-varying coefficient model can be fitted by constructing a set of time-dependent covariates. For a given covariate x^*, define $\boldsymbol{x}^*(t) = \{\boldsymbol{G}(t)x^*\}$ and cumulative hazard and survival estimates are:

$$\hat{\Lambda}\{t|\boldsymbol{G}(t), x^*\} = \sum_{i=1}^{n} \int_0^t \frac{\exp\{\hat{\boldsymbol{\beta}}^\top \boldsymbol{x}^*(u)\} dN_i(u)}{\sum_j Y_j(u) \exp\{\hat{\boldsymbol{\beta}}^\top \boldsymbol{X}_j(u)\}}, \tag{7}$$

and

$$\hat{S}\{t|\boldsymbol{G}(t), x^*\} = \exp\left[-\hat{\Lambda}\{t|\boldsymbol{G}(t), x^*\}\right]. \tag{8}$$

3. Implications to SAS and R

In this section we assume that the reader is generally familiar with survival analysis in `proc phreg`, `coxph` and `survfit`, but less clear about the case of time-varying coefficients. For an additional review see Allison (2010); Therneau (2014); Therneau and Grambsch (2000). Here, we demonstrate code to implement the above models and identify the corresponding equations that are used to calculate survival probabilities.

For this illustration, we fabricate a toy data set `SURV`, which contains all relevant variables in a single-record-per-patient style for six subjects. The data is as follows:

```
  id time death age female
1  1    1     1  20      0
2  2    4     0  21      1
3  3    7     1  19      0
4  4   10     1  22      1
5  5   12     0  20      0
6  6   13     1  24      1
```

The unique subject identifier is `id`. The variable `death` takes on a value of 1 if the subject dies and 0 if the subject is censored. The time of death or censoring is captured by `time`. The predictors of interest are `age` and gender `female`.

3.1. Data preparation

Both `proc phreg` and `coxph` have relevant functionalities that require a *counting process* style of input. The counting process data structure is nicely described by Allison (2010) and Fox and Weisberg (2011). Essentially, the data are expanded from one record-per-patient to one record-per-interval between each event time, per patient. This structure is motivated by the fact that the partial-likelihood includes a contribution at each event time. Covariate information needs to be updated and available at these times, but not in between. In R, the **survival** package has a function `survSplit` that can be used for this purpose and we provide a SAS macro, `cpdata`, with similar utility.

The inputs to `survSplit` are defined by Therneau (2014). We describe them for the present application.

- `data` = R *data frame*
  that identifies the single-record-per-patient data set that we want to expand into the counting process style.
- `cut` = *numeric vector*
  of unique event times.
- `end` = *character string*
  corresponding to the variable name for time of event or censoring in `data`. This will become the variable representing the end, or stop time, of each time interval in the counting process style.
- `event` = *character string*
  corresponding to the name of the binary variable that indicates events in `data`. It is important that this variable takes a value of 1 for events and 0 otherwise.

- `start` = *character string*
  providing a name for the new time variable that will be created to identify the beginning of each interval.
  ...

To convert `SURV` into the counting process style, we first define a vector of unique event times:

```
R> cut.points <- unique(SURV$time[SURV$death == 1])
```

We then call `survSplit` and save the result to `SURV2`.

```
R> library("survival")
R> SURV2 <- survSplit(data = SURV, cut = cut.points, end = "time",
+    start = "time0", event = "death")
```

To make the appearance match SAS, we sort `SURV2` by subject then rename and reorder the columns.

```
R> SURV2 <- SURV2[order(SURV2$id), ]
R> colnames(SURV2) <- c("id", "time1", "death", "age", "female", "time0")
R> SURV2 <- SURV2[, c("id", "age", "female", "time0", "time1", "death")]
R> SURV2
```

```
   id age female time0 time1 death
1   1  20      0     0     1     1
2   2  21      1     0     1     0
8   2  21      1     1     4     0
3   3  19      0     0     1     0
9   3  19      0     1     7     1
4   4  22      1     0     1     0
10  4  22      1     1     7     0
16  4  22      1     7    10     1
5   5  20      0     0     1     0
11  5  20      0     1     7     0
17  5  20      0     7    10     0
23  5  20      0    10    12     0
6   6  24      1     0     1     0
12  6  24      1     1     7     0
18  6  24      1     7    10     0
24  6  24      1    10    13     1
```

This data set contains four "at-risk" intervals, $(0, 1]$, $(1, 7]$, $(7, 10]$ and $(10, 13]$, representing segments between event times. The follow-up period for each person is broken up into segments, one for each interval they began at-risk (event free and uncensored). The endpoints of the intervals are defined by `time0` and `time1` in the `SURV2` data set. The benefit of this structure is that the value of a time-dependent covariate (time-varying coefficient) can be updated to occupy a row at each event time. `survSplit` creates one extra, final interval for

people who are censored. In the example, these correspond to rows three and twelve, which could be deleted. However, they will be ignored by `coxph` so we leave them. The first subject survived only through the first event time and gets one record. The second person survived through event time 1, but was censored prior to event time 2, and gets two records although the later is extraneous. The third person died at the end of the second at-risk interval, gets two records, and so on.

In order to perform the same data manipulation in SAS, we provide the following SAS macro with similar utility. The original data set is transformed by a call to `cpdata`:

```
%cpdata(data = , time = , event = , outdata = );
```

The input arguments are defined as follows:

- `data` = SAS *data set*
  that identifies the single-record-per-patient data set that we want to expand into the counting process style.
- `time` = *variable*
  in `data` for the time of event or censoring.
- `event` = SAS *code*
  that identifies the event/censoring indicator as used in `PROC PHREG`. Specifically, this is the variable name of the event/censoring indicator, immediately followed by the value(s) that correspond to censoring, enclosed in parenthesis. Examples include: `censor(1)` or `death(0)`.
- `outdata` = SAS *data set*
  name for the data set to be output in the counting process format.

The full macro definition is given in the supplementary material. To convert `SURV` we call:

```
%let FILEPATH = C:\ ;
%include "&FILEPATH.cpdata.sas";
%cpdata(data = SURV, time = time, event = death(0), outdata = SURV2)
proc print data = SURV2; run;
```

Obs	id	death	age	female	time0	time1
1	1	1	20	0	0	1
2	2	0	21	1	0	1
3	3	0	19	0	0	1
4	3	1	19	0	1	7
5	4	0	22	1	0	1
6	4	0	22	1	1	7
7	4	1	22	1	7	10
8	5	0	20	0	0	1
9	5	0	20	0	1	7
10	5	0	20	0	7	10

```
 11     6      0       24       1         0        1
 12     6      0       24       1         1        7
 13     6      0       24       1         7       10
 14     6      1       24       1        10       13
```

This data set resembles SURV2 above, with the exception that extraneous records are eliminated.

3.2. Estimation using R

For completeness, we begin with the simple Cox proportional hazard model for the effect of age and gender on survival, and estimate survival probabilities for a person of average age (21) and male gender. The following code accomplishes the task in R:

```
R> library("survival")
R> model.1 <- coxph(Surv(time, death) ~  female + age, data = SURV)
R> covs <- data.frame(age = 21, female = 0)
R> summary(survfit(model.1, newdata = covs, type = "aalen"))

Call: survfit(formula = model.1, newdata = covs, type = "aalen")

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1      6       1   0.9475   0.108     7.58e-01            1
    7      4       1   0.8672   0.236     5.08e-01            1
   10      3       1   0.7000   0.394     2.32e-01            1
   13      1       1   0.0184   0.117     7.14e-08            1
```

The newdata option is used in survfit to provide a data frame with the desired covariate values. Otherwise, survfit provides estimates for the average of all predictors, regardless of whether they are numeric or categorical. The type = "aalen" option specifies that the empirical cumulative hazard estimator be used. This is the only option available for the time-dependent model. The variable survival in the output is calculated according to Equation 3.

Suppose age does not have a constant effect, but instead the hazard ratio varies by a factor of log(time). Specifically, $\boldsymbol{X}(t) = \{$age, age * log(t)$\}$. In order to accommodate time-varying coefficients coxph requires that a counting process syntax be used. We add a time dependent covariate to SURV2 and fit the model:

```
R> SURV2$lt_age <- SURV2$age * log(SURV2$time1)
R> model.2 <- coxph(Surv(time0, time1, death) ~  female + age + lt_age,
+    data = SURV2)
```

Note that the interaction between age and log(time) is specified using the end of the interval, time1. To estimate survival for a 21 year old male we might be inclined to try the following:

```
R> summary(survfit(model.2, newdata = covs, type = "aalen"))
```

This fails, however, with the error message, "object 'lt_age' not found". The function survfit requires that values be specified for every covariate in the model, including time-dependent ones. We need to enter the changing values of lt_age and communicate to survfit

that these correspond to changing values over time for a single individual. According to the documentation, "When the original model contained time-dependent covariates, then the path of that covariate through time needs to be specified in order to obtain a predicted curve. This requires `newdata` to contain multiple lines for each hypothetical subject which gives the covariate values, time interval, and strata for each line (a subject can change strata), along with an id variable which demarks which rows belong to each subject. The time interval must have the same `(start, stop, status)` variables as the original model: although the `status` variable is not used and thus can be set to a dummy value of 0 or 1, it is necessary for the variables to be recognized as a '`Surv`' object" (Therneau 2014).

In order to achieve this, we take the time interval values corresponding to an individual with the last event time in `SURV2`. This contains all the necessary time intervals. The code is as follows:

```
R> last <- SURV2$id[which.max(SURV2$time1)]
R> intervals <- SURV2[SURV2$id == last, c("time0", "time1", "death")]
```

We then add on the constant values of interest for age and gender and create the special interaction between our fixed age of interest and log(time).

```
R> covs <- data.frame(age = 21, female = 0, intervals)
R> covs$lt_age <- covs$age * log(covs$time1)
```

Next we call `survfit` using the `newdata = covs` and `individual = TRUE` options.

```
R> summary(survfit(model.2, newdata = covs, individual = TRUE))

Call: survfit(formula = model.2, newdata = covs, individual = TRUE)

 time n.risk n.event survival  std.err lower 95% CI upper 95% CI
    1      6       1   0.9625 9.12e-02        0.799            1
    7      4       1   0.8798 9.26e+02        0.000            1
   10      3       1   0.7188 3.36e+03        0.000            1
   13      1       1   0.0815 1.12e+03        0.000            1
```

The variable `survival` in the output is calculated correctly, according to Equation 8.

3.3. Estimation using SAS

As before, we first estimate survival for a standard Cox proportional hazards model. The following SAS code fits a Cox proportional hazards model for the effect of age and gender on survival, and estimates survival for a person of average age and male gender:

```
proc phreg data = SURV;
  class female (ref = "0");
  model time * death(0) = female age;
  baseline out = outset survival = survival / method = emp;
  run;
```

The option `method = emp` specifies that the empirical cumulative hazard estimator is used, corresponding to the `type = "aalen"` option in R. The variable `survival` in the `outset` data set is calculated according to Equation 3.

Consider the case where age does not have a constant effect. Unlike R, we do not immediately have to switch to the counting process style of input to fit the model. Instead, the special time-dependent covariate can be included in `proc phreg` as follows:

```
proc phreg data = SURV;
  class female (ref = "0");
  model time * death(0) = female age lt_age;
  lt_age = age * log(time);
  run;
```

The `baseline` statement will not work with this syntax. Therefore, in order to obtain survival estimates, we must opt for the counting process syntax.

Having created `SURV2`, we can add `lt_age` to the data set and fit the model. As before, `lt_age` is defined using the ending interval time `time1`.

```
data SURV2;
  set SURV2;
  lt_age = age * log(time1);
  run;

proc phreg data = SURV2;
  class female (ref = "0");
  model (time0, time1) * death(0) = female age lt_age;
  baseline out = outset survival = survival / method = emp;
  run;
```

This syntax allows the `baseline` statement to run. However, the resulting estimates are not useful. SAS cannot tell that `lt_age` is time-varying and survival is predicted at the average value of `lt_age`, which is the default for all numeric variables. Correspondingly, the value of `lt_age` in `outset` is constant over all time. The logical remedy is to specify a covariate data set as we did in R. A similar `covs` data set can easily be created, but there is no way to tell `proc phreg` that the multiple records correspond to time epochs for a single individual. SAS will not use the data set correctly and instead repeats the time-invariant Equation 3, over each record as if `lt_age` were set to various constant values. There is not a ready-made syntax in `proc phreg` to correctly calculate Equation 8.

However, there is a simple work-around. First, note that the reference subject has $x^* = 0$ and therefore Equation 7 simplifies to

$$\hat{\Lambda}\{t|\boldsymbol{G}(t), 0\} = \sum_{i=1}^{n} \int_0^t \frac{dN_i(u)}{\sum_j Y_j(u) \exp\{\hat{\boldsymbol{\beta}}^\top \boldsymbol{X}_j(u)\}}.$$

For $x^* = 0$, this estimator is correctly provided by the `baseline` statement in `proc phreg`, using time-varying information in the denominator. We observe that when $x^* \neq 0$ Equation 7

can be re-written as

$$\hat{\Lambda}\{t|\boldsymbol{G}(t), x^*\} = \sum_{i=1}^{n} \int_0^t \frac{dN_i(u)}{\sum_j Y_j(u) \exp\left[\hat{\boldsymbol{\beta}}^\top \{\boldsymbol{X}_j(u) - \boldsymbol{x}^*(u)\}\right]}.$$

We can make a change of variables and run the procedure on a new variable, $\boldsymbol{X}_i^{'}(t) = \boldsymbol{X}_i(t) - \boldsymbol{x}^*(t)$. The baseline cumulative hazard and survival estimators, corresponding to an individual with $\boldsymbol{X}^{'}(t) = 0$, are identical to Equations 7 and 8.

Thus, we change the reference to a desired value using a `data` step, redefine the time-dependent covariate that accounts for changing hazards and use `proc phreg` to estimate the baseline hazard and corresponding survival. For example, we can correctly estimate survival for a male of average age by specifying:

```
data  SURV3;
  set SURV2;
  age = age - 21;
  lt_age = age * log(time1);
  run;

data covs;
  age = 0;
  lt_age = 0;
  female = 0;
  run;

proc phreg data = SURV3;
  class female (ref = "0");
  model (time0, time1) * death(0) = age lt_age female;
  baseline out = outset survival = survival covariates = covs / method = emp;
  run;
```

This gets around the problem of specifying changing values for `lt_age` because it remains constant at 0 for the reference individual. We print `outset` to see that the result closely matches that obtained in R:

```
Obs    age    lt_age    female    time1    survival
 1      0        0         0        0      1.00000
 2      0        0         0        1      0.96246
 3      0        0         0        7      0.87980
 4      0        0         0       10      0.71882
 5      0        0         0       13      0.08155
```

4. Introduction to `coxtvc`

When survival estimation is desired for multiple covariate values, such as prediction for the entire sample, it is relatively straightforward in R and we demonstrate this below. In SAS it

is cumbersome to organize the data and re-fit the model for every individual prediction. A call to `coxtvc` essentially encapsulates these steps:

```
%coxtvc(data = , y = , x = , tvvar = , nontvvar = , covs = , ests = ,
  modopts = , procopts = , addstmts = , out = SurvEsts);
```

The required input arguments are as follows:

- `data` = SAS *data set*
  including outcome and predictor variables in the counting-process format specified in Section 3.1.

- `y` = SAS *code*
  for the response as specified for the counting-process style of input in `proc phreg`. Most often, this takes the form `Y = (time0, time1) * event(0)`.

- `x` = *variable list*
  including all variables that appear in the `model` statement within a call to `proc phreg`. Each variable is separated by a space.

- `tvvar` = *variable list*
  including all variables that have time-varying coefficients. Note that each variable listed in `tvvar` may not necessarily be in `x`, depending on how the model is parameterized.

- `nontvvar` = *variable list*
  including all variables that *do not* have time-varying coefficients. These variables *must* appear in `x` as well.

- `covs` = SAS *data set*
  containing covariate values at which to estimate survival. This data set should contain values for variables listed in both `nontvvar` and `tvvar`. Variables not specified in the `covs` data set will be set to their average values if the variable is numeric and their reference values if categorical. `class` variables are determined as in `proc phreg` when the `baseline` statement is used. However, averages are calculated based on one-record per-patient, which differs from `proc phreg` calculations when using the counting-process style of input.

- `ests` = SAS *data set*
  containing the estimates from the fitted model. If unspecified, the survival model is fit to obtain these estimates. Depending on the complexity of the model, this fitting procedure could be time-costly. See `proc phreg` documentation on the `inest =` option within the `phreg` statement for more details on specifying this data set.

The following arguments are optional. Each governs the options used in fitting the survival model and / or in obtaining survival estimates. It may be easier to fit the model externally and obtain the `ests` data set to avoid possible complications from using these parameters.

- `modopts` = SAS *code*
  specifying options to use in fitting the model that are specified after a / in the `model` statement of `proc phreg` (ignored if `ests` is specified above). This should be enclosed in `%str()` to ensure proper evaluation.

- `procopts` = SAS *code*
  specifying options to use in the `proc phreg` statement when fitting the model (ignored if `ests` is specified above). This should be enclosed in `%str()` to ensure proper evaluation.
- `addstmts` = SAS *code*
  including additional statements that should be used when fitting the model and generating survival estimates. Usually, these will be restricted to the `freq` statement, `weight` statement, and possibly `class` statement. This should be enclosed in `%str()` to ensure proper evaluation.
- `out` = *name of a* SAS *data set*
  that will store the resulting survival estimates (default is `SurvEsts`).

The full macro definition is given in the supplementary material. In addition to the above parameters, a special macro `vardefn` must be defined by the user prior to a call to `coxtvc`. This macro contains the processing statements used to create the variables that account for the time-varying coefficients. All time-dependent variable definitions go inside the `vardefn` macro as if they were encountered in a data step. We do note that the use of the `coxtvc` macro requires SAS 9.1 or higher.

We first exemplify the macro syntax for the toy data set `SURV2` and then address practical applications in the following section. In the example of `SURV2` where the coefficient for age varies by a factor of log(time), the following code is used to estimate survival for a 21-year old male:

```
data covs;
  age = 21;
  female = 0;
  run;

%macro vardefn;
  lt_age = age * log(time1);
  %mend;

%include "&FILEPATH.coxtvc.sas";
%coxtvc(data = SURV2,
  y = (time0, time1) * death(0),
  x = age lt_age female,
  tvvar = age,
  nontvvar = female,
  covs = covs;
```

Note that `vardefn` is just an excerpt of the code that was used to create `lt_age` in the data step when constructing `SURV2`. Observe that the `covs` data set no longer requires us to change the reference value for `age`, nor define a value for `lt_age`; these steps are taken care of within the macro.

Suppose that, in addition to the time-varying coefficient on age, we want to allow the coefficient for female gender to differ before and after 7 days. Corresponding time-dependent variables have not yet been added to the `SURV2` data set. We can let the macro take care of

that by redefining the `vardefn` macro and then appropriately calling the `coxtvc` macro as follows:

```
%macro vardefn;
  if (time1 < 7) then do;
    female_lt7 = female;
    female_ge7 = 0;
    end;

  if (time1 >= 7) then do;
    female_lt7 = 0;
    female_ge7 = female;
    end;

  lt_age = age * log(time1);
  %mend;

%coxtvc(data = SURV2,
  y = (time0, time1) * death(0),
  x = female_lt7 female_ge7 age lt_age,
  tvvar = female age,
  nontvvar = ,
  covs = covs);
```

The `covs` data set is no different than before, since we are still interested in survival for a 21-year old male. Since the `ests` parameter is left empty, `ests` is created internally by fitting the corresponding model. The variables used to account for the time-varying coefficients (`female_lt7 female_ge7 lt_age`) are added to the `SURV2` data set prior to fitting the model. Again, the data set `survests` contains the estimated survival values.

5. Examples

5.1. Simulated example

The `coxtvc` macro is validated in a simulated example. For each of 500 replications, we generate $n = 2500$ event times T with the following survival function,

$$S(t) = \exp\left\{\frac{-t}{8}\exp(-0.8x)\right\} I(t \leq 4) + \exp\left\{\frac{-1}{2}\exp(-0.8x) - \left(\frac{t-4}{8}\right)\exp(0.8x)\right\} I(t > 4),$$

where x is a treatment indicator, taking the value 1 if the subject is randomized to treatment and 0 otherwise. Censoring times were uniformly distributed on the interval $[0, 8]$, and treatment was randomly assigned to each subject with probability 0.5. For each replication, the survival function was estimated using the SAS macro `coxtvc` where we assume the cutpoint ($t = 4$) is known. Figure 1 shows the true survival function with the range of estimates

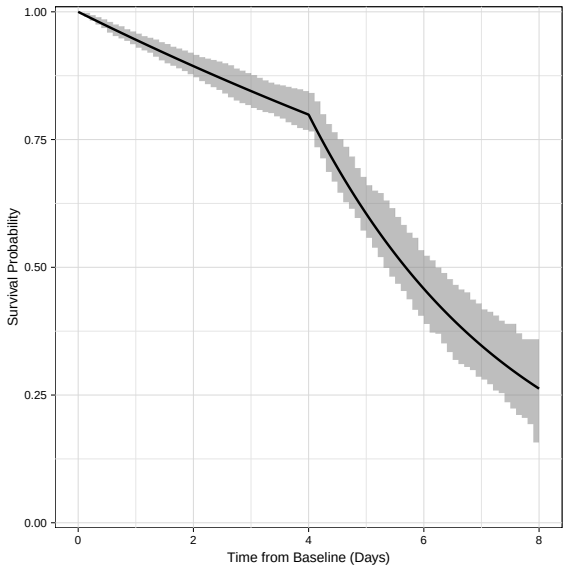

Figure 1: Range of survival estimates from 500 simulated data sets on $n = 2500$ subjects.

obtained overlaid. We also investigated the Monte Carlo confidence band for the mean estimated survival function. It was so narrow as to be indistinguishable from the true survival function on the plot. We conclude that the true survival function is estimated well by this method. The SAS code for running this simulation and R code for creating the figures is given in the supplementary material.

5.2. Application

Allison (2010) describes a study following inmates released from a state prison. The study's aim was to determine factors associated with the first arrest following release. We use the `coxtvc` macro to create survival estimates and assist with the interpretation of the analysis for this recidivism data set. For completeness, we repeat the analysis in R, demonstrating the relative simplicity. The data are publicly available at http://ftp.sas.com/samples/A61339. Define the macro variable `FILEPATH` to be your working directory which contains the `cpdata` and `coxtvc` macros, as well as a text file for the recidivism data.

Consider a model associating the time (in weeks) until first arrest with age of the inmate at the time of release, whether the inmate received financial aid, and the number of prior arrests. Following Allison (2010), we establish that the variables age (`age`) and financial aid (`fin`) violate the assumption of proportional hazards in the survival model.

```
libname proj "&FILEPATH.";

data rossi; set proj.recid;
  keep week arrest fin age race wexp mar paro prio educ;
  run;

proc phreg data = rossi;
  model week * arrest(0) = age fin prio age_week fin_mid / rl;
```

```
  age_week = age * week;
  fin_mid = fin * (20 < week < 30);
  run;
```

An excerpt from the `proc phreg` output shows that the parameters for `age_week` and `fin_mid` are significantly different from 0. That is, there is evidence that the parameter estimate corresponding to each of these predictors varies over time. Specifically, the coefficient for age changes linearly with time, while the coefficient for financial aid differs between 20 and 30 weeks following release.

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq
age	1	0.03186	0.03924	0.6592	0.4169
fin	1	-0.15979	0.20501	0.6075	0.4357
prio	1	0.09770	0.02725	12.8520	0.0003
age_week	1	-0.00380	0.00146	6.7863	0.0092
fin_mid	1	-1.45337	0.66473	4.7803	0.0288

At baseline, the hazard ratio for those who receive financial aid compared to those who do not (holding age and the number of prior arrests constant) is $\exp(-0.160) = 0.85$ while it is $\exp(-0.160 - 1.453) = 0.20$ between 20 and 30 weeks. This violation of proportional hazards seems fairly important since those receiving financial aid status are at much lower risk of being arrested around six months after release. How does this translate to event probabilities on average? Specifically, we are interested in knowing how receiving financial aid reduces the probability of being arrested over time. For that, we want to look at *adjusted* survival curves.

To obtain a direct adjusted survival curve (Zhang, Loberiza, Klein, and Zhang 2007), we want to compute the survival curve for *every* subject in the sample size twice – once with `fin = 0` and once with `fin = 1`. These estimated curves are then averaged across subjects to obtain two adjusted curves. These curves estimate the chance of remaining free from arrest, among two cohorts with equivalent age and prior arrests: one that receives financial aid and one that does not. Computing the adjusted curves would be cumbersome using the centering approach of Section 3.3 as there are several unique values of age in the sample. However, this is easily accomplished using the `coxtvc` macro.

```
data covs;
  set rossi (keep = age prio);
  do fin = 0 to 1;
    output;
    end;
  run;

%cpdata(data = rossi,
    time = week,
    event = arrest(0),
```

```
    outdata = rossi2);

%macro vardefn;
  age_week = age * week1;
  fin_mid = fin * (20 < week1 < 30);
  %mend vardefn;

%coxtvc(data = rossi2,
    y = (week0, week1) * arrest(0),
    x = age fin prio age_week fin_mid,
    tvvar = age fin,
    nontvvar = prio,
    covs = covs,
    out = survest);

proc sort data = survest;
  by fin week1;
  run;

data avgsurv;
  set survest;
  by fin week1;

  retain sumshat total;
  if (first.week1) then do;
    sumshat = 0;
    total = 0;
    end;

  sumshat = sumshat + shat;
  total = total + 1;

  if (last.week1) then do;
    avgshat = sumshat / total;
    output;
    end;
  run;
```

The `survest` data set created by the `coxtvc` macro looks exactly like the standard output data set from the `baseline` statement in `proc phreg`; it contains survival curves for every subject. The call to `proc sort` and the final data step condense the output to create the two survival curves.

The adjusted survival curves contained in `avgsurv` are plotted in Figure 2 along with estimates that assume proportional hazards for financial aid status and age (code for generating plots can be found in the supplementary material).

In the curves that allow time-varying coefficients, we see that whether an inmate receives financial aid has minimal impact on the arrest probability prior to 20 weeks and substantial

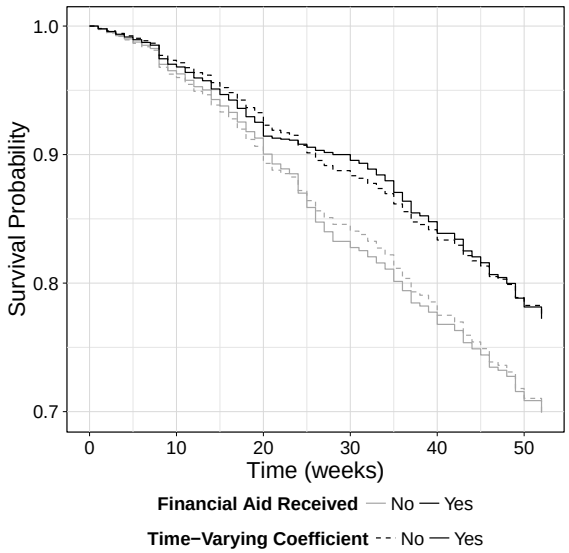

Figure 2: Adjusted effect of financial aid status on survival.

impact beyond. This is completely consistent with the hazard ratios. However, we also see that the proportional hazards model gives a very similar message, that financial aid is beneficial for avoiding arrest. For some applications, this plot may provide reassurance that the proportional hazard model is adequate. For a more refined purpose, nuance in event probabilities could be important. Either way, the implications of alternative modeling strategies have been translated to a scientifically meaningful scale. The survival curves allowing a time-varying coefficient help to reveal the true effect of financial aid on the probability of being arrested.

The analysis in R is sufficiently simple that a macro is not required. We exemplify the steps, first bringing in the data and converting to the counting process style, as in the previous example.

```
R> rossi <- read.csv(paste(FILEPATH, "rossi.csv", sep = ""),
+    header = TRUE)[, 1:10]
R> rossi$id <- 1:nrow(rossi)
R> cut.points <- unique(rossi$week[rossi$arrest == 1])
R> rossi2 <- survSplit(data = rossi, cut = cut.points, end = "week",
+    start = "week0", event = "arrest")
```

Then, we define and run a function that generates the special time-dependent covariates that will allow for time-varying effects of age and financial aid. We name this function `vardefn` exactly as it was named in SAS.

```
R> vardefn <- function(data) {
+    data$age_week <- data$age * data$week
+    data$fin_mid <- data$fin * (20 < data$week & data$week <30)
+    return(data)
+  }
R> rossi2 <- vardefn(rossi2)
```

Now we are able to fit the time-varying coefficient model.

```
R> model <- coxph(Surv(week0, week, arrest) ~  age + fin + prio + age_week +
+    fin_mid, ties = "breslow", data = rossi2)
```

In order to obtain predictions from this time-varying model, we create a data set with the intervals between event times and another for covariate values at which we want predictions.

```
R> last <- rossi2$id[which.max(rossi2$week)]
R> intervals <- rossi2[rossi2$id == last, c("week0", "week", "arrest")]
R> covs <- data.frame(rbind(cbind(rossi[c("age", "prio")], fin = 0),
+    cbind(rossi[c("age", "prio")], fin = 1)))
```

Notice that we do not join the `intervals` and `covs` data sets yet. `covs` is now a matrix with rows corresponding to the covariate vectors of interest. Specifically, we have repeated the entire data set twice, first letting `fin` equal 0 and then letting `fin` equal 1. Next we create a loop where the rows of `covs` are picked off and predictions are obtained exactly as in the toy example above. Covariate values are merged back onto the corresponding predictions `shat`.

```
R> survest <- NA
R> r <- nrow(covs)
R> for (i in 1:r) {
+    newdata <- data.frame(covs[i, ], intervals, row.names = NULL)
+    newdata <- vardefn(newdata)
+    out <- cbind(newdata, shat = summary(survfit(model, newdata = newdata,
+      individual = TRUE))$surv)
+    survest <- rbind(survest, out)
+  }
```

Taking a simple average over the adjustment variable, `age`, within `week` and `fin`, gives us the same results as `avgsurv`, obtained by SAS.

```
R> avgsurv2 <- t(tapply(survest$shat, INDEX = survest[, c("fin", "week")],
+    FUN = mean))
```

These data can be plotted to obtain Figure 2. We do not repeat this exercise because the results are identical.

6. Discussion

In this article, we describe the utility of the R function `coxph`, and SAS procedure `proc phreg`, for survival estimation with time-varying coefficient models and provide SAS macros to facilitate calculations. Statisticians often recommend that our collaborators focus on finding clinically important differences, rather than simply statistical significance. It is hard to define clinical importance in terms of the hazard. When survival curves are plotted for two groups of individuals, differences in net survival are easily perceived. Additionally, in large data sets, a statistically significant violation of proportional hazards may be detected for all

covariates, when in fact, the practical significance of such violations is very minor. As we see in the recidivism data, survival estimates may assist scientists when evaluating whether differences between statistical models translate to practical differences, for a given purpose. In applications like this, hypothesis testing may proceed on the hazard scale and figures are used to show clinical importance of the point estimates. In this case, confidence intervals would greatly distract from the figure. However, estimation of uncertainty around survival estimates can be very important. For prediction on a single vector of covariates, standard errors are easily obtained with the usual `proc phreg` and `coxph` options. The calculation of standard errors for direct adjusted survival curves is complex, even in the simple proportional hazards model (Zhang *et al.* 2007). This is an important avenue of future work.

Once it is understood how each function processes information and implements calculations, extensions to this application are straightforward. For example, one might implement a proportional hazards model for the effect of treatment on outcome, allowing for time-varying effects. Adjustment for imbalance in covariates could be implemented by inverse probability of treatment weighting, in contrast to the regression adjustment shown here. The techniques illustrated here would allow for survival estimation according to treatment with a parametric time-varying hazard. Weights could be included for adjustment. The data processing steps would be the same, and the analytical steps would be easily extended. For that reason, we emphasize strategies to understand and utilize the software, as much as the macros.

Other methods of flexible hazard modeling should be considered. The time-varying coefficient models described by Martinussen and Scheike (2006) have advantages over the current approach, in that coefficients are not required to vary in a predefined fashion. When a predefined function is not known, the approach we have described can be made more general with the use of cubic splines (Hess 1994). However, our scientific collaborators often prefer a pre-specified model for the hazard, where all coefficients change according to a simple, interpretable function. Like any hazards regression, survival estimation is an important complement. This article may help to alleviate confusion about the utility of SAS and R softwares for survival estimation in the context of time-varying coefficients and increase its utilization.

References

Allison PD (2010). *Survival Analysis Using SAS: A Practical Guide*. 2nd edition. SAS Publishing, Cary.

Buchholz A, Sauerbrei W (2011). "Comparison of Procedures to Assess Non-Linear and Time-Varying Effects in Multivariable Models for Survival Data." *Biometrical Journal*, **53**(2), 308–331.

Cox DR (1972). "Regression Models and Life Tables." *Journal of the Royal Statistical Society B*, **34**(2), 187–220.

Fox J, Weisberg S (2011). *Appendix: An R Companion to Applied Regression*. 2nd edition. Sage, Thousand Oaks.

Gao D, Grunwald GK, Rumsfeld JS, Schooley L, MacKenzie T, Shroyer ALW (2006). "Time-Varying Risk Factors for Long-Term Mortality After Coronary Artery Bypass Graft Surgery." *The Annals Thoracic Surgery*, **81**(3), 793–799.

Guanghui W, Schaubel DE (2008). "Estimating Cumulative Treatment Effects in the Presence of Nonproportional Hazards." *Biometrics*, **64**(3), 724–732.

Harrell FE (2006). *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. Springer-Verlag, New York.

Hess KR (1994). "Assessing Time-by-Covariate Interactions in Proportional Hazards Regression Models Using Cubic Spline Functions." *Statistics in Medicine*, **13**(10), 1045–1062.

Kalbfleisch JD, Prentice RL (2002). *The Statistical Analysis of Failure Time Data*. 2nd edition. John Wiley & Sons, New York.

Klein JP, Moeschberger ML (2003). *Survival Analysis: Techniques for Censored and Truncated Data*. 2nd edition. Springer-Verlag, New York.

Martinussen T, Scheike T (2006). *Dynamic Regression Models for Survival Data*. Springer-Verlag, New York.

Natarajan L, Pu M, Parker BA, Thomson CA, Caan BJ, Flatt SW, Madlensky L, Hajek RA, Al-Delaimy WK, Saquib N, Gold EB, Pierce JP (2009). "Time-Varying Effects of Prognostic Factors Associated With Disease-Free Survival in Breast Cancer." *American Journal of Epidemiology*, **169**(12), 1463–1470.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

SAS Institute Inc (2008). *SAS/STAT Software, Version 9.2*. Cary. URL http://support.sas.com/documentation/92/.

SAS Institute Inc (2010). *SAS OnlineDoc, Version 9.2*. SAS Institute Inc., Cary. URL http://www.sas.com/.

Scheike TH, Zhang MJ (2011). "Analyzing Competing Risk Data Using the R **timereg** Package." *Journal of Statistical Software*, **38**(2), 1–15. URL http://www.jstatsoft.org/v38/i02/.

Therneau TM (2014). ***survival**: A Package for Survival Analysis in S*. R package version 2.37-7, URL http://CRAN.R-project.org/package=survival.

Therneau TM, Grambsch PM (2000). *Modeling Survival Data: Extending the Cox Model*. Springer-Verlag, New York.

Zhang X, Loberiza FR, Klein JP, Zhang MJ (2007). "A SAS Macro for Estimation of Direct Adjusted Survival Curves Based on a Stratified Cox Regression Model." *Computer Methods and Programs in Biomedicine*, **88**(2), 95–101.

Affiliation:

Laine Thomas
Department of Biostatistics and Bioinformatics
Duke University
Durham, North Carolina 27705, United States of America
E-mail: laine.thomas@duke.edu
URL: http://biostat.duke.edu/faculty/details/0402620/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 61, Code Snippet 1 *Submitted:* 2012-09-28
October 2014 *Accepted:* 2014-06-06