simPH: An R Package for Showing Estimates for Interactive and Nonlinear Effects from Cox Proportional Hazard Models

Christopher Gandrud
Hertie School of Governance

Abstract

The R package **simPH** provides tools for effectively communicating results from Cox Proportional Hazard (PH) models, especially models with interactive and nonlinear effects. The Cox Proportional Hazard model is a popular tool for examining cross-unit cross-time data. However, many mis-specify their models and poorly communicate uncertainty about their estimates. This is unfortunate because causes of model misspecification–e.g., interactive and nonlinear effects–may be substantively meaningful. Uncertainty about these effects can be difficult to assess because quantities of interest are often on asymmetric and nonlinear scales. Part of the problem has been that available computational tools make it difficult to explore and communicate these effects. **simPH** overcomes these problems by making it easy for to simulate and then plot quantities of interest for a variety of effects estimated from Cox PH models. The package can also be used for simple linear effects, making it a useful all-round package for showing results from Cox PH models. **simPH**'s plots employ visual weighting in order to effectively communicate estimation uncertainty. The user also has the option of showing either the standard central interval of the simulation's distribution or the shortest probability interval–which can be especially useful for asymmetrically distributed estimates. Hypothetical and empirical examples are used to illustrate **simPH**'s syntax and capabilities.

Keywords: Cox Proportional Hazard models, hazard ratios, time-interactions, time-varying, nonlinearity, splines, visual-weighting, R.

The Cox (1972) Proportional Hazards (PH) model is used in a wide range of disciplines including epidemiology and political science. However, many mis-specify their models and poorly communicate uncertainty about their estimates. This is unfortunate because causes of model mis-specification–e.g., interactive and nonlinear effects–may be substantively meaningful. Uncertainty about these effects can be difficult to assess because quantities of interest are often on asymmetric and nonlinear scales. Part of the problem has been that available computational tools make it difficult to explore and communicate these effects.

This article aims to improve the use of Cox-type models in two ways. After briefly discussing previous research on how Cox-type models can be misspecified by ignoring time interactions and nonlinear effects for continuous variables, it (a) advocates using shortest probability intervals and visual-weighting to display simulated quantities of interest that describe these estimated effects. Then (b) it makes it easy to use these methods by demonstrating the new R (R Core Team 2014) package **simPH** (Gandrud 2014), that is freely available on the Comprehensive R Archive Network. The package makes it easy to simulate distributions of

quantities of interest estimated from Cox-type models. The central or shortest probability intervals of these distributions can then be plotted using visually-weighted plots. The latter type of interval is often more appropriate for Cox-type model results. Quantities of interest **simPH** can handle include hazard ratios, first differences, relative hazards, marginal effects, and hazard rates from linear, linear multiplicative interactions between covariates, time interactions, and nonlinear coefficients for continuous variables. Hypothetical and empirical examples are used to illustrate **simPH**'s syntax and capabilities.

1. The Cox PH model

Before discussing model misspecification problems, let's quickly look at what Cox PH models are. The Cox PH model is a semi-parametric survival model that allows the examination of how specified factors influence the rate of a particular event happening–e.g., infection, death, the adoption of a public policy–at a particular point in time given that the event has not already occurred. This rate is commonly referred to as the hazard rate ($h_i(t)$). The hazard rate for unit i at time t is estimated with the Cox PH model using:

$$h(t|\mathbf{X}_i) = h_0(t)\mathrm{e}^{(\beta\mathbf{X}_i)}, \tag{1}$$

where $h_0(t)$ is the baseline hazard, i.e., the instantaneous rate of a transition at time t when all of the covariates are zero. β is a vector of coefficients and $\mathbf{X}_i$ is the vector of covariates for unit i.

We are often interested in how a covariate changes the rate of an event happening. In general researchers have tried to address this by looking at Cox PH coefficient estimates β. However, only examining single coefficients can lead to significant model misspecification and erroneous substantive interpretation of Cox PH results.

2. Nonproportional hazards

One of the most important sources of estimation bias in Cox PH models discussed at length by Licht (2011), Box-Steffensmeier and Zorn (2001), and Box-Steffensmeier and Jones (2004) is a violation of the proportional hazards assumption (PHA). The PHA is that the hazards of two units experiencing an event are proportional to one another and that this relationship is constant over time. Formally, for the PHA to hold the hazard for units j and l must be:

$$\frac{h_j(t)}{h_l(t)} = \mathrm{e}^{\beta'(x_j - x_l)}. \tag{2}$$

for all points in time. This is also the equation for the hazard ratio between x_j and x_l. If the proportional hazards assumption is violated and measures are not taken to correct for the violation, then researchers may create biased parameter estimates and statistical tests with lower power (Therneau, Grambsch, and Fleming 1990; Keele 2010). Beyond these statistical problems, not adjusting for violations of the PHA can prevent researchers from finding evidence for phenomena they are interested in studying, including how an effect changes over time and whether or not it changes nonlinearly over the range of a variable's values.

There are a number of widely used tests to examine if the PHA has been violated. See Grambsch and Therneau (1994), Box-Steffensmeier and Zorn (2001), and Box-Steffensmeier

and Jones (2004) for discussions of various PHA testing methods.Many software packages implement versions of these tests. R's **survival** package (Therneau 2014) implements Grambsch and Therneau's (1994) modified Schoenfeld residuals test. This is done with the `cox.zph` function.

2.1. Nonproportional hazards and time-interactive effects

If a covariate is determined to violate the PHA, Box-Steffensmeier and co-authors (see Box-Steffensmeier, Reiter, and Zorn 2003; Box-Steffensmeier and Jones 2004) suggest directly modeling the relationship between the variable and time. This usually entails including an interaction between the variable and some function of time such as the natural logarithm or some exponent. The decision to use a particular functional form should be guided by theory and will likely also be influenced by findings in the data. If $f(t)$ is some function of time then a simple Nonproportional Hazards Cox model estimating the hazard rate for unit i with one time-interaction is given by:

$$h_i(t|\mathbf{x}_i) = h_0(t)\mathrm{e}^{(\beta_1 x_i + \beta_2 f(t) x_i)}. \tag{3}$$

Like any other interaction effect (see Brambor, Clark, and Golder 2006) extra care should be taken when interpreting the β_1 and β_2 parameter estimates and their associated uncertainty. We cannot simply interpret the effect by looking at β_1 or β_2 in isolation. They need to be combined. Licht (2011) argues that post-estimation simulation techniques should be employed to substantively interpret these combined coefficients and the uncertainty surrounding them. Let's briefly look at ways to calculate combined effects. Later in this section we will look at showing our uncertainty about them using simulation techniques.

Licht (2011) describes two methods for calculating the combined effect of a time interaction on the hazard of an event happening in ways that are relatively easy to interpret: (a) first differences and (b) relative hazards. A first difference is the percentage change in the hazard rate at time t between two values of x:

$$\%\triangle h_i(t) = (e^{(x_j - x_l)(\beta_1 + \beta_2 f(t))} - 1) \cdot 100. \tag{4}$$

Relative hazards are given by:

$$\frac{h_j(t)}{h_l(t)} = \mathrm{e}^{x_j(\beta_1 + \beta_2 f(t))}. \tag{5}$$

In this situation the covariate x_l is 0. Relative hazards represent the change in the hazard when x is 'switched on'. As such, relative hazards are a special case of the hazard ratio (Licht 2011, 231). They are the expected change in the hazard when x is fitted at a value different from zero compared to when x is zero.

2.2. Nonproportional hazards and nonlinear effects

Time-interactive effects are not the only cause of PHA violations. Building on Grambsch and Therneau (1994) and Therneau and Grambsch (2000), Keele (2010) points out that common diagnostic tests will also indicate PHA violations if the model is misspecified for other reasons. Omitting an important covariate, using a proportional hazards model even if another survival model is more appropriate, or including a continuous covariate as linear when its effect is

actually nonlinear can lead to significant PHA tests. Addressing the omission of important covariates is beyond the scope of the **simPH** package.

Because of this Keele suggests that *before* testing the PHA we should try to make sure that we are not omitting important variables and find the covariates' appropriate functional forms, typically using either polynomials or splines. He demonstrates this in replication studies by adding penalized splines then using a Wald test to examine if the spline estimates have a better fit than their linear counterparts. Many studies using Cox PH models do not test for nonlinearity, but instead jump straight to testing the PHA, including time-interactions when it is violated. As Keele (2010) demonstrates, ascribing a time-interactive effect to a covariate when in fact the effect varies not over time, but nonlinearly over values of the covariate can have major implications for substantive interpretations of results.

3. Show estimation uncertainty

How can we effectively examine and communicate both the point estimates of and our uncertainty about quantities of interest from time-interactive and nonlinear effects? In this section I advocate showing these results using simulations, shortest probability intervals, and visually-weighted plots.

3.1. Post estimation simulations

Following King, Tomz, and Wittenberg (2000), Licht (2011) proposes post-estimation simulation techniques to make it easier to estimate the uncertainty surrounding quantities of interest for time interactions like first differences and relative hazards. See King *et al.* (2000, 352-353) for a discussion of alternative approaches including fully Bayesian Markov-Chain Monte Carlo techniques and bootstraping. The main difference between these three approaches is how the parameters are drawn. Using the post-estimation simulation technique, we first find the parameter point estimates for $\hat{\beta}_1$ and $\hat{\beta}_2$ from a Nonproportional Hazards Cox model that make up the time-interactive effect as well as the parameter covariance estimates. Second, we draw n values of β_1 and β_2 from the multivariate normal distribution with a mean of $\hat{\beta}$ and variance specified by the parameters' estimated covariance. Third, we use these simulated values to calculate a quantity of interest such as the first difference or relative hazard for a range of times as well as specified values of x_j and x_l (as appropriate). Finally, we plot the results. Using this simulation technique allows us to estimate full time-interactive effects, how they change over time, substantively evaluate the effects, and show the uncertainty surrounding the estimates.

We can easily extend this simulation technique to quantities of interest for other estimated effect types, including nonlinear effects. For example if a nonlinear effect is modeled with a second order polynomial, i.e. $\beta_1 x_i + \beta_2 x_i^2$, we can once again draw n simulations from the multivariate normal distribution for both β_1 and β_2. Then we simply calculate quantities of interest for a range of values and plot the results as before. For example, we find the first difference for a second order polynomial with:

$$\%\triangle h_i(t) = (\mathrm{e}^{\beta_1 x_{j-l} + \beta_2 x_{j-l}^2} - 1) \cdot 100, \tag{6}$$

where $x_{j-l} = x_j - x_l$. Note we will not be showing the estimated effect over time. For this we need to estimate the hazard rate for a range of comparisons between x_j and x_l.

We can use a similar procedure for splines. Penalized splines (Eilers and Marx 1996) are a commonly used way of showing more complex nonlinear effects than polynomials (see Keele 2008). They involve "linear combinations of B-spline basis functions" (Strasak, Lang, Kneib, Brant, Klenk, Hilbe, Oberaigner, Ruttmann, Kaltenbach, Concin, Diem, Pfeiffer, and Ulmer 2009, 5) joined at points in the range of observed values of x called "knots" (Keele 2008, 50). A Cox PH model with one penalized spline is given by:

$$h(t|\mathbf{X}_i) = h_0(t)\mathrm{e}^{g(x)}, \tag{7}$$

where $g(x)$ is the penalized spline function. For our post-estimation purposes $g(x)$ is basically a series of linearly combined coefficients such that:

$$g(x) = \beta_{k_1}(x)_{1+} + \beta_{k_2}(x)_{2+} + \beta_{k_3}(x)_{3+} + \ldots + \beta_{k_n}(x)_{n+}, \tag{8}$$

where k are the equally spaced spline knots with values inside of the range of observed x and n is the number of knots. x_{c+} indicates that:

$$(x)_{c+} = \begin{cases} x & \text{if } k_{c-1} < x \leq k_c \\ x & \text{if } x \leq k_1 \text{ and } k_c = k_1 \\ x & \text{if } x \geq k_n \text{ and } k_c = k_n \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Note, x should be within the observed data.

We can again draw values of each $\beta_{k_1}, \ldots \beta_{k_n}$ from the multivariate normal distribution described above. We then use these simulated coefficients to estimate quantities of interest for a range of covariate values. For example, the first difference between two values x_j and x_l is:

$$\%\triangle h_i(t) = (\mathrm{e}^{g(x_j)-g(x_l)} - 1) * 100. \tag{10}$$

Relative hazards and hazard ratios can be calculated by extension. Once we find the simulated quantities of interest, plotting the results is straightforward.

We can use this post-estimation simulation technique for virtually any quantity of interest estimated from Cox-type models including marginal effects for multiplicative interactions (see Brambor *et al.* 2006) and plain linear effects.

3.2. Which interval to show?

If researchers go beyond the usual 'train timetable' coefficient tables and graphically show their parameter estimates, as **simPH** makes easier, they usually do so by plotting lines of some measure of central tendency and confidence bands calculated from standard errors over a range of fitted values. Previous work with post-estimation simulations, e.g., Licht (2011), has mirrored this approach in graphs with a line for the median or mean of the simulation distribution and lines representing the boundaries of a central interval of the distribution. For example, the central 95 percent interval could be represented by lines at the 2.5 and 97.5 percentiles of the distribution.

Many quantities of interest from Cox-type models have asymmetric probability distributions on a nonlinear scale and can therefore be poorly summarized by central intervals. Recall that most quantities of interest are on an exponential scale with a lower bound of 0 or, in the

case of first differences, -100. They can have very long and sparse upper regions relative to a tighter concentration of the distribution near the lower boundary. In these cases it can be more useful to look at highest density regions (see Box and Tiao 1973; Hyndman 1996). The underlying idea is that we should be more interested in the set of quantities of interest values with the most probability (Hyndman 1996, 120), rather than an arbitrary central interval. When the simulation has a normal distribution, the highest density region will be equivalent to the central interval with the same percentage of the simulations, e.g., 95 percent. However, when the highest density is at the boundary, for example when many of the simulated relative hazard values are close to 0, then Liu, Gelman, and Zheng (2013) argue that the highest density region is preferable to the central interval. In these cases "central intervals can be much longer and have the awkward property [of] cutting off a narrow high-posterior slice that happens to be near the boundary, thus ruling out a part of the distribution that is actually strongly supported by the inference" (Liu *et al.* 2013, 2). If this happens Liu *et al.* recommend finding the shortest probability interval (SPIn). This is the shortest interval of a particular probability coverage based on the simulations. They find this to be a very efficient way of finding the shortest highest density region for unimodal distributions.

3.3. Visual weighting

Whether graphing a central or shortest probability interval, only using lines to represent the center and edges can draw the reader's attention away from what the graph is trying to communicate. This approach overemphasizes the edges of the interval, the areas of lowest probability. Some graphs uniformly shade the interval between the upper and lower bounds. Uniform shading suggest to the reader a uniform distribution between the edges. Both of these characteristics give misleading information about the quantities of interests' probability distributions, especially when they are on an exponential scale.

Visual weighting presents a solution to these problems. Hsiang calls visual weight "the amount of a viewer's attention that a graphical object or display region attracts, based on visual properties" (2012, 3). More visual weight can be created by using more "graphical ink" (Tufte 2001). Visual weight is decreased by removing graphical ink. The simplest way to increase or decrease graphical ink with our simulations is to simply plot each simulation value as a point or series of values with a line that is semi-transparent. Areas of the distribution with many simulations will be darker. Areas with fewer simulations, often near the edges of the distribution, will be lighter. Plotting semi-transparent points or lines allows for very clear communication of a quantity of interest's probability distribution. When each point or line is partially transparent, areas of the chart where the points or lines are darker indicate areas of the distribution with higher probability, because more points or lines are stacked on top of one another. This approach gives more visual weight to areas of higher probability and avoids drawing the reader's attention to the edges of the distribution, the areas of lower probability, in the way that traditional confidence interval lines do.

As a practical issue if a plot shows very many simulations as individual points, and to a lesser extent lines, it may create a very large file size. This is especially true if higher quality vector graphics are desired. An alternative it to summarize simulated distributions with multiple ribbons of increasing transparency the further from the central tendency a portion of the distribution is. These ribbons could stretch across given segments of the distribution such as the central 50 and 95 percentage intervals. Using ribbons rather than points conveys somewhat

Table 1: **simPH** quantity of interest simulation functions

Simulation function	Effect type
`coxsimLinear`	linear
`coxsimInteract`	linear multiplicative interactions
`coxsimPoly`	polynomials
`coxsimtvc`	time-interactive
`coxsimSpline`	penalized splines

less information about a distribution, but can be convenient if very many simulations are plotted.

4. simPH: Tools for simulating and graphing effects

One reason that researchers have inconsistently incorporated suggestions to test for and examine time interactions and nonlinearities in their Cox-type models and show their estimation uncertainty is that there has been a lack of computational capabilities to easily do so. In R the **survival** (Therneau 2014) package has functions for testing the proportional hazards assumption. The **survival** and **Zelig** (Owen, Imai, King, and Lau 2013) packages can estimate models with splines and interactions. However, their capabilities for graphically showing these estimates and associated uncertainty are very limited. **Zelig** can plot basic estimates from Cox PH models using simulation techniques, but not time-interactive or nonlinear spline effects. Current capabilities for showing results from splines present results in difficult to interpret quantities. In general the capabilities for showing results from time interactions is very limited. Usually, showing these types of results requires considerable effort to extract estimates from model objects and then devise ways to show them graphically. See for example Licht's () Stata (StataCorp 2009) code for replicating the time-interaction plots in her paper. No method allows you to easily plot shortest probability intervals and virtually none effectively uses visual weighting.

To solve these problems I am introducing the **simPH** package for R. There are three basic steps to use the package:

1. Estimate a Cox-type model using **survival**'s `coxph` function.
2. Simulate parameter estimates and calculate quantities of interest, e.g., relative hazards, first differences, hazard ratios, marginal effects for linear interactions, or hazard rates, using the function from the **simPH** package corresponding to the effect type. See Table 1 for a summary of the simulation functions.
3. Plot the simulations using the `simGG` method.

simPH's simulation functions follow King *et al.* (2000) to simulate parameters[1] and calculate a variety of quantities of interest. The user can specify the number of simulations to run per

[1] **simPH** uses the **MASS** (Ripley 2014) R package to draw the simulations.

fitted value with the `nsim` argument. The default is 1000. The more simulations conducted, the better picture we get of the probability distributions our parameters are from (King *et al.* 2000, 349). Warning: in some cases, especially with penalized splines, it is easy to ask the program to create more simulations than average desktop computers can easily handle. Therefore the user may need to balance a desire for a clear view of the probability distribution a quantity of interest comes from with what is computationally feasible.

''**simPH**'s simulation functions allow the user to keep either the traditional central interval of the simulations' distributions (the middle 95 percent by default) or use the shortest probability interval (also 95 percent by default). In either case the range is set with the `ci` argument, i.e., to find the central 90 percent interval use `ci = 0.9`. To tell **simPH** to find the shortest probability interval with any of **simPH**'s simulation functions simply set the argument `spin = TRUE`.[2]

The `simGG` plotting method can then be used to plot these simulated values as semi-transparent points, lines, or ribbons. It is important to note that while points and ribbons plot values in the selected interval for each value of the x-axis lines plots simulations in the selected interval for all values plotted on the x-axis. This is to avoid creating jagged line plots, though it creates an interval with a slightly different interpretation. The transparency level can be set with the argument `alpha`. If points or lines are used, `alpha` sets the transparency level for simulation values at the center of the distribution. Simulation values further from the center become more transparent the further out they are. A smoothing line of a type that can be specified by the user with the `method` argument is also plotted to summarize the distribution's central tendency. In general, any smoothing method accepted by **ggplot2** (Wickham and Chang 2013) can be used. By plotting semi-transparent points for each simulated quantity of interest value or lines for the series of values from each simulation, `simGG` visually weights simulation distributions.

If the user selects ribbon plots, three ribbons will be show. The most transparent shows the furthest extent of the central or shortest probability interval. The less transparent ribbon shows the central 50 percent of this interval. And the middle line shows the interval's median.

To choose between points, lines, or ribbons use the `type` argument and set it to `points`, `lines`, or `ribbons`, respectively. The default is `type = 'points'`.

The `simGG` method draws on **ggplot2** to plot the simulations. In most cases `simGG` returns a **ggplot2** gg object. As such you can add any aesthetic attributes to the plots that **ggplot2** will allow.

5. Demonstrations

To illustrate **simPH**'s syntax and capabilities we will first go through a simple example without interactive or nonlinear effects. Then we will replicate key figures and findings from Licht (2011) and Keele (2010). The quantities of interest from these studies are from time-interactive and penalized spline effects. For examples with other quantities of interest and variable types please see **simPH**'s documentation.

[2]This capability was developed from Liu *et al.* (2013) and the accompanying code in Liu's (2013) **SPIn** R package. It is currently unavailable for hazard rates.

5.1. Simple non-interactive and linear example

simPH can be used to show results from effects that are not explicitly modeled as interactions or nonlinearly. To do this use the `coxsimLinear` function. Let's look at an example using a hypothetical data set called `hmohiv` from Hosmer Jr, Lemeshow, and May (2008). The data is hosted by and the basic estimation model is from UCLA Academic Technology Services (2014). The data set contains 100 members of a Health Maintenance Organization (HMO). All of the members are HIV positive. The HMO wants to examine their survival times. Note: the model we will estimate below does not violate the proportional hazard assumption.

Let's estimate a simple Cox PH model that includes information on the members' ages and intravenous drug use. We can load the necessary packages for this example and download the data with the following code:

```
R> library("survival")
R> library("simPH")
R> hmohiv <- read.table(
+              "http://www.ats.ucla.edu/stat/r/examples/asa/hmohiv.csv",
+              sep = ",", header = TRUE)
```

In the data set intravenous drug use is recorded as a binary variable called `drug`, where one is a history of drug use and zero otherwise. Each member's age in years is recorded in a variable called `age`. The ages range from 20 to 54 with a median of 35.

We are going to present results for how age is associated with survival times. Remember that quantities of interest such as relative hazards and hazard ratios are comparisons. Relative hazards are simply hazard ratios comparing a unit with a value of zero on some variable to another unit with another value of this variable. For the `age` variable this would mean a comparison between someone with an age of zero and someone with some other age. This may not be a particularly interesting comparison. In our case, it is also an out of sample comparison as the youngest observed HMO member is 20 years old. We can easily create a much more substantively interesting comparison by making the reference value the median age (35):

```
R> hmohiv$AgeMed <- hmohiv$age - 35
```

The new variable ranges from -15 to 19. Now all of the simulated relative hazards will be based on a comparison with a 35 year old. Let's estimate the model using the `coxph` function from the **survival** package:

```
R> M1 <- coxph(Surv(time, censor) ~ AgeMed + drug,
+              method = "breslow", data = hmohiv)
```

Age is clearly a time-varying quantity. Not only does it simply vary over time in the way a person's income (probably) does, for example, but in effect it has a linear interaction with time. However, in our model we do not explicitly interact age with time as we will interact other variables with time in later examples. Therefore we can treat the variable in the same way as personal income for the purposes of creating simulations with **simPH** and can therefore use the `coxsimLinear` function:

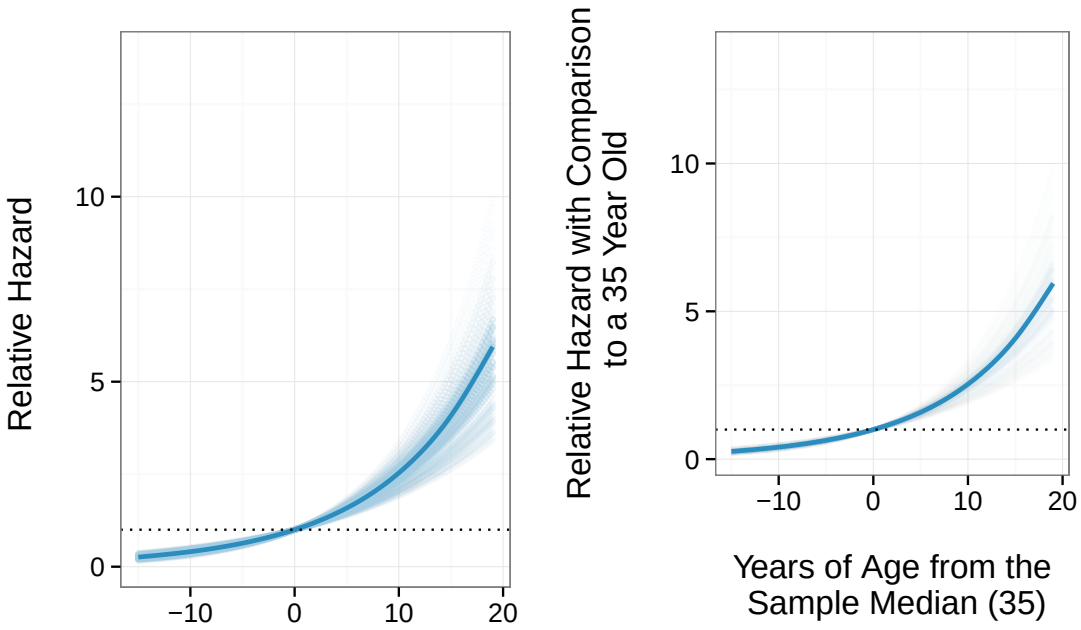

Figure 1: Simulated relative hazards of age on survival time for HIV positive HMO members.

```
R> Sim1 <- coxsimLinear(M1, b = "AgeMed", Xj = seq(-15, 19, by = 0.2), nsim = 100)
```

We told `coxsimLinear` that we wanted to simulate relative hazards (the default quantity of interest) based on our model object `M1`. We specify which variable to find hazard ratios for using the `b` argument. The `Xj` argument lets us set the fitted values of x_j; in this case a sequence of values between -15 and 19 at intervals of 0.2. The smaller the interval, the smoother the resulting graph will look. To graphically present the results now in the `Sim1` object we can use the `simGG` method. The following code creates the left panel of Figure 1:

```
R> simGG(Sim1)
```

We can set the x and y axis labels make other aesthetic changes to create the right panel of Figure 1 using the following code:

```
R> simGG(Sim1, xlab = "\nYears of Age from the Sample Median (35)",
R>       ylab = "Relative Hazard with Comparison\n to a 35 Year Old\n",
R>       alpha = 0.05, type = 'lines')
```

In Figure 1 we can see that the relative hazard at `AgeMed` zero (age 35) is one. A relative hazard for a unit at zero is always one, as it is a ratio of the hazards with itself. The simulated relative hazards for ages below the median are less than one. This means that they are less likely to die at a given point in time than someone aged 35. Ages above the median have a relative hazard greater than one and are therefore more likely to die than 35 year olds. We can see in this figure that we have estimated a 54 year old (those 19 years older than the median) to be about six times more likely than 35 year olds to die, all else equal.

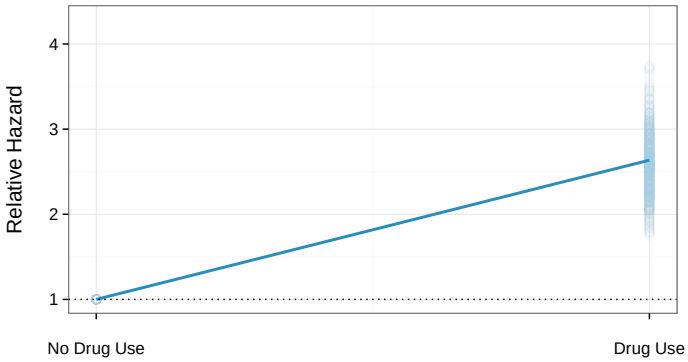

Figure 2: Simulated relative hazards of drug use history for HIV positive HMO members.

Three quick notes on syntax. The `xlab` and `ylab` arguments in `simGG` allowed us to set the x-axis and y-axis labels respectively. The backslash followed by the letter `n` creates a new line in the output.

We could also use **simPH** to find and plot quantites of interest for the binary `drug` use variable. For example, using the model object `M1` as before we first simulate relative hazards for the variables' two levels zero and one:

```
R> Sim2 <- coxsimLinear(M1, b = "drug", Xj = 0:1)
```

We can then use `simGG` (with some stylistic modifications)[3] to create the plot in Figure 2. Remember from our earlier discussion in Section 2 that a relative hazard is the change in the hazard when the covariate is 'switched on' at some level that is not zero. As the `drug` variable is binary, it only has two realistic levels: zero and one. As such there is only really one substantively meaningful relative hazard in Figure 2.

5.2. Time-interactive effects example

Let's now look at how to use **simPH** to explore time-interactive effects that are explicitly created by interacting a given variable in the estimation model with some time transformation. To do this we will recreate plots from Licht (2011). She re-examines Golub and Steunenberg's (2007) analysis of European Union legislative deliberation time. They wanted to find out what factors affected the amount of time it took the European Union to pass a new piece of legislation. Key variables they examined included the voting procedure that was used for a given piece of legislation, including the so-called qualified majority vote (QMV) procedure[4], and the amount of other legislation pending, i.e. legislative backlog. Both of these variables violated the proportional hazard assumption and were more accurately modeled with log-time interactions.

[3]It is important to note that the `method` argument should be set to `method = "lm"` (i.e. linear) when using binary variables. The nonlinear smoothers will not work with fewer than 10 x-axis values.

[4]The procedure changed over time, but essentially QMV requires some voting majority greater than 50% + 1. See the European Union's website for more details: http://europa.eu/legislation_summaries/glossary/qualified_majority_en.htm (accessed January 2014).

The left panel of Figure 3 recreates Licht's (2011) figure showing the first difference of a log-time interaction for how the effect of the QMV procedure on legislative deliberation time changes as the number of days of deliberation increases (see Licht 2011, 236).[5] To create this figure we first load the data `GolubEUPData`. This data set is included in **simPH**.

```
R> data("GolubEUPData")
```

Then we create the log-time interactions. Before jumping into this, let's look at the data's format:

```
R> head(GolubEUPData[, 2:5])
##   caseno begin  end event
## 1      1     0  595     1
## 2      2     0 1246     1
## 3      4     0  341     1
## 4      5     0  335     1
## 5      6     0  522     1
## 6      7     0 1664     0
```

We can see that each piece of prospective legislation is identified with a case number in the `caseno` variable. The `event` variable records if the event of interest occurred (i.e. the legislation was passed) or not. Observation intervals `begin` at 0 and extend to the `end` when either the event occurs, or one of the covariate values change. If we simply create a time interaction by using the `end` variable as the time value, we will over-measure the interaction for failures before this time. For example, imagine we create a simple interaction between a binary variable and time. Imagine one unit has the value 1 for the binary variable and that this unit experiences an event of interest at time 1,000 in a data set formatted as above. The time interaction would then be treated as 1,000 for all failures at time $t \leq 1,000$. This would be inaccurate. At time 500, for example the interaction should be 500, not 1,000.

To solve this problem, **simPH** includes a function called `SurvExpand` to expand a data set out into equally spaced intervals. It keeps all intervals that end at an observed event time or when a covariate changes. These are the only time points interesting to the Cox PH model and removing unneeded periods helps save memory. The resulting data frame allows us to create accurate time interactions. Here is how we use the function with the `GolubEUPData` data:

```
R> GolubEUPData <- SurvExpand(GolubEUPData, GroupVar = 'caseno',
+                         Time = 'begin', Time2 = 'end', event = 'event')
R>
R> head(GolubEUPData[, 1:4])
##   caseno begin end event
```

[5] Her original figure was not in terms of a percentage difference to make it more comparable to a figure in Golub and Steunenberg's original. The results are presented in terms of percentage difference, as the first difference is commonly reported. The pre and post Single European Act time periods are not separated out for simplicity. Finally, I also examined whether or not nonlinearity functional forms would be better fits than time interactions as per our discussion above. However, no evidence of this.

```
## 1      1     0   1     0
## 2      1     1   2     0
## 3      1     4   5     0
## 4      1     6   7     0
## 5      1     8   9     0
## 6      1    10  11     0
```

Now we can create the log-time interactions.

The two time-interaction variables we will focus on are for qualified majority voting (`qmv`) and legislative backlog (`backlog`). Other time-interacted variables are from the original model. Notice below the use of the `tvc` function to create the log-time interactions. The function is included with the **simPH** package. The `data` argument specifies the data frame where our covariate and time variables are. The covariate we wish to interact is specified with `b` and the time variable with `tvar`. Finally, we specify the function of time with the `tfun` argument. The additional creation of the `Golubtvc` function is not necessary, it just makes the code tidier.

```
R> Golubtvc <- function(x){
+    tvc(data = GolubEUPData, b = x, tvar = "end", tfun = "log")
+  }
R> GolubEUPData$Lqmv <- Golubtvc("qmv")
R> GolubEUPData$Lbacklog <- Golubtvc("backlog")
R> GolubEUPData$Lcoop <- Golubtvc("coop")
R> GolubEUPData$Lcodec <- Golubtvc("codec")
R> GolubEUPData$Lqmvpostsea <- Golubtvc("qmvpostsea")
R> GolubEUPData$Lthatcher <- Golubtvc("thatcher")
```

Now estimate the model with `coxph`. Note that each log-time interaction is entered into the model along with the corresponding non-time interacted term.

```
R> M2 <- coxph(Surv(begin, end, event) ~ qmv + qmvpostsea + qmvpostteu +
+                coop + codec + eu9 + eu10 + eu12 + eu15 + thatcher +
+                agenda + backlog + Lqmv + Lqmvpostsea + Lcoop + Lcodec +
+                Lthatcher + Lbacklog,
+              data = GolubEUPData, ties = "efron")
```

In the following code chunk we take the `M2` model object created by `coxph` and use it in **simPH**'s `coxsimtvc` function to simulate the first difference of the time-interactive effect of qualified majority voting on directive deliberation time from 80 through 2000 days after deliberation begins. We specify the quantity of interest that we want to simulate with the `qi` argument. The non-time interacted term is entered with the `b` argument as before. The time interacted term is entered with `btvc`. The form of the time interaction is declared with the `tfun` argument, i.e `tfun = "log"`.[6] `Xj = 1` specifies that the difference is between values of `qmv` 1 and 0. This is a comparison between using QMV or not. We could change the x_l

[6] Other options include `"linear"` for linear time-interactions and `"pow"` for polynomials. If `tfun = "pow"` then also set the argument `pow` to specify the power the time interaction was raised to.

value with the `Xl` argument, but this is not relevant for binary variables. The `from` and `to` arguments allow us to specify the time period over which to simulate the quantity of interest. The `by` argument allows us to specify the increment of the time sequence to simulate values at.

```
R> Sim3 <- coxsimtvc(obj = M2, b = "qmv", btvc = "Lqmv",
R>                   qi = "First Difference", Xj = 1, tfun = "log",
R>                   from = 80, to = 2000, by = 5, nsim = 100)
```

Once we have created the `Sim3` object containing the simulations, we can simply use the `simGG` method to plot the results. In this example, we adjusted the median line size with the `lsize = 0.5` argument. This makes this particular plot clearer. Finally, the `legend = FALSE` argument hides the legend describing the fitted value comparisons that are being made between x_j and x_l. A legend would be fairly uninformative in this case as we are only comparing the values 1 and 0.[7] Finally, in we added a title to the whole plot using the `title` argument.

```
R> simGG(Sim3, xlab = "\nTime in Days", title = "Central Interval\n",
R>       type = "ribbons", lsize = 0.5, legend = FALSE, alpha = 0.3)
```

We can clearly see in the left panel of Figure 3[8] that QMV increases the probability of passing a directive early in the deliberation process (almost by 300 percent at about 80 days). But as the deliberation time increases, the effect decreases and then becomes negative at about 1000 days. Finally, the rate of decrease levels off after 1000 days relative to before.

The right panel of Figure 3 shows the shortest-probability interval for QMV simulated from `M2`. To create this we used all of the same code as above, except `spin = TRUE` was added to the `coxsimtvc` call. Notice that in this case, the central and shortest probability intervals are largely equivalent, because the underlying simulation distribution is fairly normally distributed. Also because `simGG` returns **ggplot2** `gg` model objects, we can use `grid.arrange` from the **gridExtra** package (Auguie 2012) to combine the two plots into one figure.[9]

Let's look at another example, this time with multiple comparisons plotted in one figure. Figure 4 [10] shows the effect of different levels of legislative backlog (`backlog`) on directive deliberation time from 1200 to 7000 days after the directive was proposed. The effect shown is also modeled as a log-time interaction. The process for creating the plot is similar to what we have already seen. The only differences to note are that we entered a sequence of values for backlogged legislation for `Xj` in `coxsimtvc` ranging from 40 to 200 at increments of 40. This creates five separate sets of relative hazard simulations, one for each value of `Xj` compared

[7] The default is to show the legend. The `legend` argument follows the **ggplot2** syntax for creating plot guides, so to show the legend use `legend = "legend"`.

[8] The figures' ribbons extend across the middle or shortest probability 95 percent interval of the simulations. As in Licht's (2011) original the time period plotted is truncated from 80 to 2000 to make the estimates more easily interpretable.

[9] If we placed the two plots created by `simGG` into their own objects called `Plot1` and `Plot2` then we could combine them into one figure using `gridExtra::grid.arrange(Plot1, Plot2, ncol = 2)`.

[10] It replicates the right panel of Licht's (2011) Figure 3 (2011, 237). One difference is that she estimates uncertainty from 10 draws of 1000 simulations, whereas Figure 4 is based on one draw of 1000 simulations. The figure's ribbons extend across the middle 95 percent of the simulations.

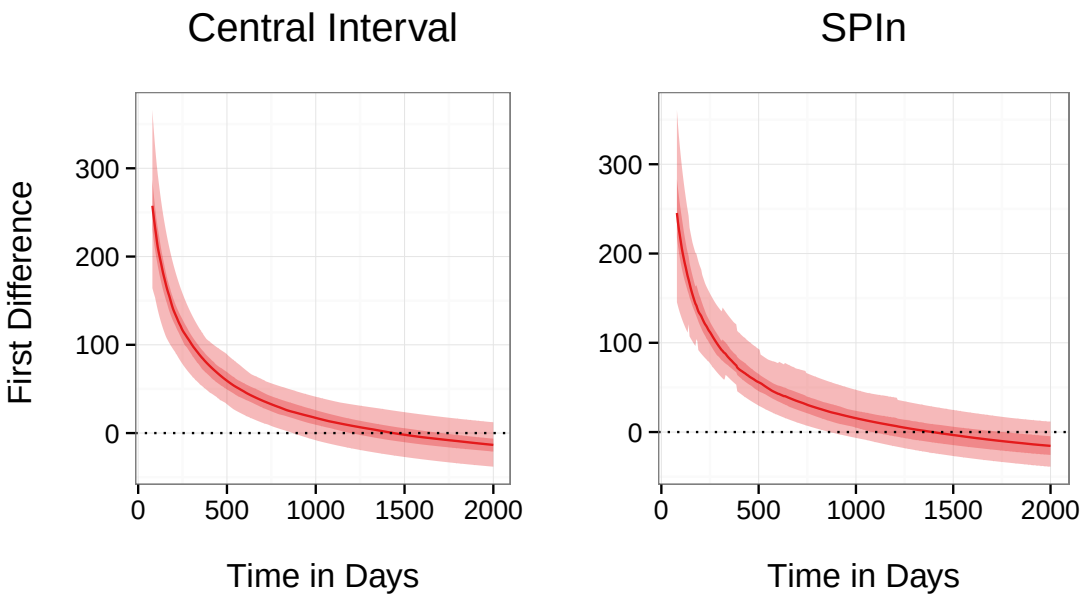

Figure 3: Simulated first differences for the effect of Qualified Majority Voting on the time it takes to pass legislation.

to 0. Finally, we specified the legend name for the plot in the `simGG` call with the `leg.name` argument.

```
R> Sim4 <- coxsimtvc(obj = M2, b = "backlog", btvc = "Lbacklog",
R>                    qi = "Relative Hazard", Xj = seq(40, 200, 40),
R>                    tfun = "log", from = 1200, to = 7000, by = 100,
R>                    nsim = 100)
R>
R> simGG(Sim4, xlab = "\nTime in Days", type = "ribbons",
R>       leg.name = "Backlogged \n Items")
```

The main conclusion we can draw from this presentation of the log-time interaction is that if a piece of legislation is not passed in the first 1200 or so days from when it was proposed, it is less likely that it will be passed if there is a large legislative backlog (for more details see Licht 2011, 236-237).

In the previous examples, the simulation probability distributions are not strongly influenced by boundary effects or long upper tails. Because of this the central and shortest probability intervals are largely equivalent. The distributions are also relatively tight, which is indicated by fairly even visual weight across the distributions.

5.3. Nonlinear effects–penalized spline–example

Let's now turn to look at how **simPH** can be used to simulate and plot quantities of interest estimated from penalized splines. To do this we will build on Keele (2010). To demonstrate why we need to check for and explicitly model nonlinearity he replicated a study by Carpenter

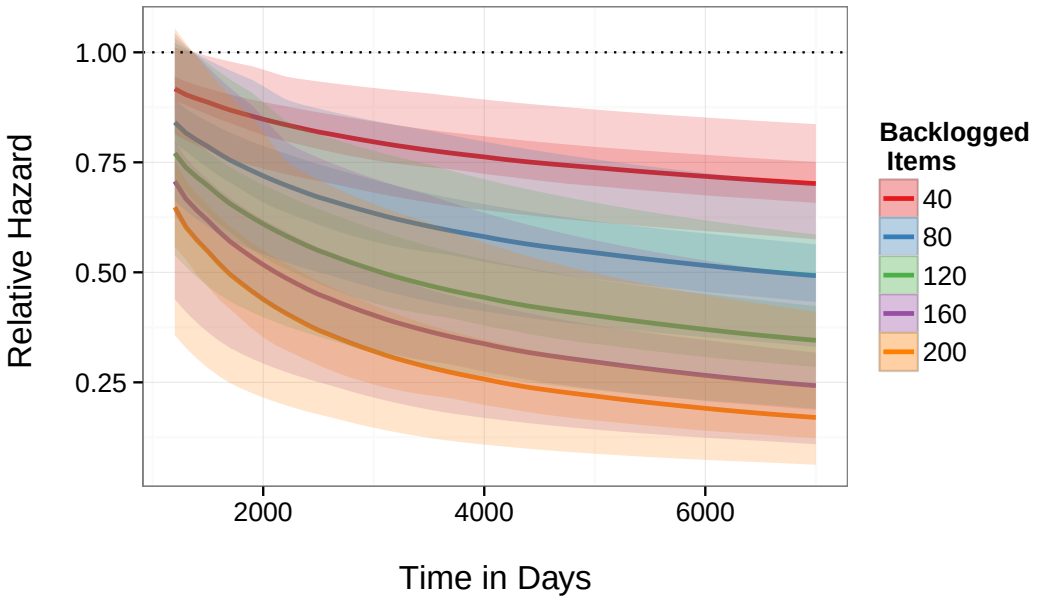

Figure 4: Simulated relative hazards for the effect of different levels of legislative backlog on directive deliberation time.

(2002) on the time it takes the US Food and Drug Administration (FDA) to approve a new drug. Using the steps discussed above, he found that modeling nonlinearity with penalized splines rather than time-interactive effects was the more appropriate strategy for dealing with covariates that violated the proportional hazards assumption. This allowed him to draw conclusions that, for example, the number of FDA review staff increases the likelihood that a drug will be accepted, but that the effect diminishes after a threshold number of staff are assigned.

It has been difficult to examine and communicate the functional form, magnitude, and uncertainty surrounding spline effects. Coefficient tables are very cumbersome, because a spline fitted effect is estimated using multiple coefficients and standard errors for values of a variable in a given range, demarcated by the knots, on the hazard. Depending on the size of the ranges[11] there can quickly be many more coefficients than can be efficiently presented in a table and understood by a reader. In his results tables, Keele does not show spline coefficients and simply denotes their overall significance with standard significance stars.

In R you can plot estimated spline effects over a range of values with the `termplot` function. These plots, however, have a number of drawbacks. First, the plots show the log hazard ratio,[12] which is not a particularly intuitive quantity to understand and is rarely reported in studies using Cox PH models. More importantly, to communicate uncertainty it plots standard errors, instead of the more widely used central confidence intervals. As such, casual readers could easily think the uncertainty around the spline estimates is smaller than it really is. The plots give no other information about the likely distribution of the estimated quantity

[11] With R's `pspline` function the range can effectively be adjusted by changing the spline's degrees of freedom.

[12] Log hazard ratios for a standard Cox PH model are given by: $\log\left\{\frac{h_j(t)}{h_h(t)}\right\} = \beta \mathbf{X}_i$ (modified from Box-Steffensmeier and Jones 2004, 49).

of interest.

simPH allows us to estimate quantities that we are more interested in like relative hazards, hazard rates over time, hazard ratios, and first differences. For example, let's simulate the hazard ratios for the effect of an additional drug review staff on the time it takes for a drug to be approved. Figure 5[13] shows the simulated hazard ratios over the full range of FDA staff per drug trial (`stafcder`) observed in Carpenter's data. Not only is this more informative than simply showing the coefficients and their standard errors that comprise the spline parameter estimates, it is also more informative than the current plotting method in R.

Let's look at the syntax used to create the left panel of Figure 5. Load Carpenter's data included with **simPH** and estimate the model with splines using `coxph` and `pspline`. The model was originally used to create the results in Keele's (2010) Table 7.

```
R> data("CarpenterFdaData")
R>
R> M3 <- coxph(Surv(acttime, censor) ~  prevgenx + lethal + deathrt1 +
+                acutediz + hosp01 + pspline(hospdisc, df = 4) +
+                pspline(hhosleng, df = 4) + mandiz01 +
+                femdiz01 + peddiz01 + orphdum + natreg + vandavg3 +
+                wpnoavg3 + pspline(condavg3, df = 4) +
+                pspline(orderent, df = 4) + pspline(stafcder, df = 4),
+              data = CarpenterFdaData)
```

Now simulate the hazard ratios for a sequence of values with `coxsimSpline` and graph the results with `simGG`.

```
R> XjFit <- seq(1100, 1700, by = 10)
R> XlFit <- setXl(Xj = XjFit, diff = 1)
R>
R> Sim4 <- coxsimSpline(M3, bspline = "pspline(stafcder, df = 4)",
R>                      bdata = CarpenterFdaData$stafcder,
R>                      qi = "Hazard Ratio",
R>                      Xj = XjFit, Xl = XlFit)
R>
R> simGG(Sim4, xlab = "\n Number of FDA Drug Review Staff",
R>         title = "Central Interval\n", alpha = 0.1,
R>         type = "lines")
```

There are a few syntax points to note. First, we created our vector of x_l values using the `setXl` function. This simply takes a vector of x_j values and creates a vector of corresponding x_l values that are different from x_j by a value we set with the `diff` argument. Second we tell `coxsimSpline` the term to estimate for using the `bspline` argument and the same syntax we used to enter the term as a penalized spline in the `coxph` call, i.e., `pspline(stafcder, df = 4)`. Third, we specified the vector containing the observed `stafcder` data with the `bdata` argument. This is important for `coxsimSpline` to be able to accurately find the spline knots.

[13]The figure's points show the middle 95 percent of the simulations at each value of FDA staff. The line summarizing the central tendency of the distribution was created with a generalized additive model for integrated smoothness estimation.

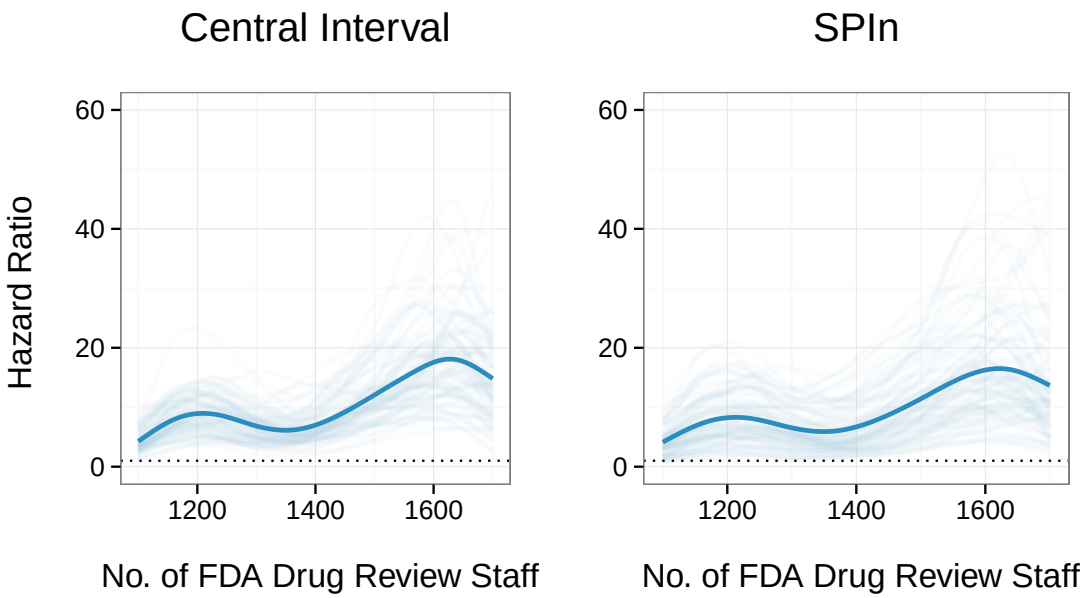

Figure 5: Simulated hazard ratios for the effect of FDA staff on drug approval time.

We can see in left panel of Figure 5 that the simulated values are concentrated near the bottom of the distribution.[14] Following Liu *et al.* we may find it useful to show not the central 95 percent interval, but the shortest 95 percent probability interval. The right panel of Figure 5 shows this interval.[15] To create it we used all of the same syntax as before, while simply adding `spin = TRUE` to `coxsimSpline`. Using the shortest 95 percent probability interval indicates that there is actually a higher probability that the hazard ratio is 1, i.e., no effect, than the central 95 percent interval for all but the highest quartile or so of FDA drug review staff. It also deemphasizes the higher hazard ratio values, where there is a low concentration of the probability. In both panels of Figure 5 the visual weighting draws readers' attention to the lower part of the distribution where the model estimates there is a higher probability that the hazard ratio will be.

Finally, to ease comparison, the second plot is on the same y-axis scale as the first. To illustrate how to do this, imagine that we placed the output of `coxsimSpline` with the argument `spin = TRUE` into an object called `SimPlot1`. We can then use **ggplot2** to change the y-axis range so that it is comparable to Figure 5's:

```
R> library("ggplot2")
R>
R> SimPlot1 + scale_y_continuous(breaks = c(0, 20, 40, 60), limits = c(0, 60))
```

[14]Note that the values have been smoothed using post-simulation cubic smoothing splines. This gives a more continuous appearance to the simulations. Smoothing is accomplished with `smooth.spline`, which is part of basic R. You can tell `simGG` to not smooth the simulations by setting the argument `SmoothSpline = FALSE`.

[15]The line summarizing the central tendency of the distribution was created with a generalized additive model for integrated smoothness estimation.

6. Conclusion

This paper reviewed a number of insights about avoiding Cox Proportional Hazard model misspecification. Looking out for and properly estimating the interactions and nonlinearities that can cause biased estimates is crucial when using Cox Proportional Hazard models. Beyond creating biased estimates, they can actually be factors that we are interested in studying. Until now the tools for fully exploring them, including their associated uncertainty, have been lacking. Coefficient estimate tables are very difficult to interpret for these types of effects. The **Zelig** package for R comes closest to allowing us to estimate and communicate uncertainty from Cox Proportional Hazards models. However, it has limited or no ability to simulate quantities of interest for interactive and nonlinear estimated effects. So, this paper has demonstrated a new R package, **simPH**, that makes it considerably easier to effectively explore and present quantities of interest for interactive and nonlinear effects. It can also be used for simple linear effects, making it a useful all-round package for showing results from Cox Proportional Hazard models. The paper has also argued for and demonstrated the usefulness of visual weighting and shortest probability intervals for understanding results from these models. **simPH** fully supports these methods.

Acknowledgments

Thank you to Andreas Beger, Jeffrey Chwieroth, Kelly Kadera, Luke Keele, Mintao Nie, Alison Post, Meredith Wilf, participants of the International Studies Association Annual Convention (2013), two anonymous reviewers, and the JSS editors.

References

Auguie B (2012). *gridExtra: functions in Grid graphics*. R package version 0.9.1, URL http://CRAN.R-project.org/package=gridExtra.

Box GEP, Tiao GC (1973). *Bayesian Inference in Statistical Analysis*. Wiley Classics, New York.

Box-Steffensmeier JM, Jones BS (2004). *Event History Modeling: A Guide for Social Scientists*. Cambridge University Press, Cambridge.

Box-Steffensmeier JM, Reiter D, Zorn CJ (2003). "Nonproportional Hazards and Event History Analysis in International Relations." *Journal of Conflict Resolution*, **47**(1), 33–53.

Box-Steffensmeier JM, Zorn CJ (2001). "Duration Models and Proportional Hazards in Political Science." *American Journal of Political Science*, **45**(4), 972–988.

Brambor T, Clark WR, Golder M (2006). "Understanding Interaction Models: Improving Empirical Analyses." *Political Analysis*, **14**(1), 63–82.

Carpenter DP (2002). "Groups, the Media, Agency Waiting Costs, and FDA Drug Approval." *American Journal of Political Science*, **46**(3), 490–505.

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

Eilers PHC, Marx BD (1996). "Flexible Smoothing with B-splines and Penalties." *Statistical Science*, **11**(2), 89–102.

Gandrud C (2014). *simPH: Tools for simulating and plotting quantities of interest estimated from Cox Proportional Hazards models*. R package version 1.2, URL http://christophergandrud.github.io/simPH/.

Golub J, Steunenberg B (2007). "How Time Affects EU Decision-Making." *European Union Politics*, **8**(4), 555–566.

Grambsch PM, Therneau TM (1994). "Proportional Hazards Tests and Diagnostics Based on Weighted Residuals." *Biometrika*, **81**(3), 515–526.

Hosmer Jr DW, Lemeshow S, May S (2008). *Applied Survival Analysis: Regression Modeling of Time to Event Data*, volume 618. 2nd edition. Wiley-Interscience.

Hsiang SM (2012). "Visually-Weighted Regression." Available at: https://dl.dropboxusercontent.com/u/3011470/WorkingPapers/HSIANG_VISUALLY_WEIGHTED_REGRESSION_v1.pdf.

Hyndman RJ (1996). "Computing and Graphing Highest Density Regions." *The American Statistician*, **50**(2), 120–126.

Keele L (2008). *Semiparametric Regression for the Social Sciences*. John Wiley & Sons, Chichester.

Keele L (2010). "Proportionally Difficult: Testing for Nonproportional Hazards in Cox Models." *Political Analysis*, **18**(2), 189–205.

King G, Tomz M, Wittenberg J (2000). "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." *American Journal of Political Science*, **44**(2), 347–361.

Licht AA (????). "Replication data for: Change Comes with Time." URL http://hdl.handle.net/1902.1/15633.

Licht AA (2011). "Change Comes with Time: Substantive Interpretation of Nonproportional Hazards in Event History Analysis." *Political Analysis*, **19**, 227–243.

Liu Y (2013). *SPIn: Simulation-efficient Shortest Probability Intervals*. R package version 1.1, URL http://CRAN.R-project.org/package=SPIn.

Liu Y, Gelman A, Zheng T (2013). "Simulation-efficient Shortest Probability Intervals." *Arvix*, pp. 1–22. http://arxiv.org/pdf/1302.2142v1.pdf.

Owen M, Imai K, King G, Lau O (2013). *Zelig: Everyone's Statistical Software*. R package version 4.2-1, URL http://CRAN.R-project.org/package=Zelig.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. Version 3.0.3, URL http://www.R-project.org/.

Ripley B (2014). *MASS: Support Functions and Datasets for Venables and Ripley's MASS*. R package version 7.3-31, URL http://CRAN.R-project.org/package=MASS.

StataCorp (2009). *Stata Statistical Software: Release 11*. College Station, TX.

Strasak AM, Lang S, Kneib T, Brant LJ, Klenk J, Hilbe W, Oberaigner W, Ruttmann E, Kaltenbach L, Concin H, Diem G, Pfeiffer KP, Ulmer H (2009). "Use of Penalized Splines in Extended Cox-Type Additive Hazard Regression to Flexibly Estimate the Effect of Time-varying Serum Uric Acid on Risk of Cancer Incidence: A Prospective, Population-Based Study in 78,850 Men." *Annals of Epidemiology*, **19**(1), 15–24.

Therneau T (2014). *survival: Survival Analysis*. R package version 2.37-7, URL http://CRAN.R-project.org/package=survival.

Therneau T, Grambsch P (2000). *Modeling Survival Data: Extending the Cox Model*. Statistics for Biology and Health. Springer-Verlag. ISBN 9780387987842.

Therneau TM, Grambsch PM, Fleming TR (1990). "Martingale-based Residuals for Survival Models." *Biometrika*, **77**(1), 147–160.

Tufte ER (2001). *The Visual Display of Quantitative Information*. 2nd edition. Graphics Press, Cheshire, CT.

UCLA Academic Technology Services (2014). *R Textbook Examples: Applied Survival Analysis, by Hosmer and Lemeshow*. URL http://www.ats.ucla.edu/stat/r/examples/asa/asa_ch4_r.htm.

Wickham H, Chang W (2013). *ggplot2: An implementation of the Grammar of Graphics*. R package version 0.9.3.1, URL http://CRAN.R-project.org/package=ggplot2.

Affiliation:

Christopher Gandrud
Hertie School of Governance
Friedrichstrasse 180
Berlin, 10117
Germany
E-mail: gandrud@hertie-school.org