coxed: An R Package for Computing Duration-Based Quantities from the Cox Proportional Hazards Model

by Jonathan Kropko and Jeffrey J. Harden

Abstract The Cox proportional hazards model is one of the most frequently used estimators in duration (survival) analysis. Because it is estimated using only the observed durations' rank ordering, typical quantities of interest used to communicate results of the Cox model come from the hazard function (e.g., hazard ratios or percentage changes in the hazard rate). These quantities are substantively vague and difficult for many audiences of research to understand. We introduce a suite of methods in the R package **coxed** to address these problems. The package allows researchers to calculate duration-based quantities from Cox model results, such as the expected duration (or survival time) given covariate values and marginal changes in duration for a specified change in a covariate. These duration-based quantities often match better with researchers' substantive interests and are easily understood by most readers. We describe the methods and illustrate use of the package.

Introduction

The Cox proportional hazards model (Cox, 1972) is frequently used for duration (survival) analysis in a myriad of disciplines including the health sciences, social sciences, operations research, and engineering. For many researchers who employ the Cox model, the chief concept of substantive interest is the duration of an event, such as the survival time of a patient or the duration of a civil war. However, the standard methods of reporting results from the Cox model—which are based in the hazard function—communicate no *specific* information about duration. As a result, standard interpretations of Cox model results are often substantively vague and difficult for many audiences of research to understand.

Here we introduce an R package implementation of *Cox proportional hazards model with expected durations*, or COX ED (Kropko and Harden, 2020). The COX ED suite of methods available in the **coxed** package provides a more intuitive approach to communicating results from the Cox model. Specifically, it computes duration-based quantities of interest, such as the expected time until event occurrence according to the estimated model. These quantities have long been available with parametric duration models, but in some instances researchers may not wish to make the distributional assumptions required of those estimators. The COX ED methods allow researchers to stay within the Cox model framework, but communicate results in the language of time. This affords more conceptual precision when conversing with other researchers and makes the results of the analysis more intuitive and accessible for general audiences.

The methodology

The goal of COX ED is to generate expected durations for individual observations and/or marginal changes in expected duration given a change in a covariate from the Cox model. Specifically, the methods can compute (1) the expected duration for each observation used to fit the Cox model, given the covariates, (2) the expected duration for a "new" observation with a covariate profile set by the analyst, or (3) the first difference, or change, in expected duration given two new observations.

There are two different methods of generating duration-based quantities in the package. The first method employs a generalized additive model (GAM) to map the model's exponentiated linear predictor values to duration times. The second method calculates expected durations by using nonparametric estimates of the baseline hazard and survivor functions. We present overviews of these methods here. See Kropko and Harden (2020) for additional details, including simulation results comparing the two methods. Importantly, both approaches use coefficient estimates from the Cox model, so researchers must first estimate the model just as they always have. COX ED is a postestimation procedure, not a new estimator. All of the choices required of applied researchers in estimating the Cox model must be made first, at the estimation stage, before proceeding to implement COX ED.[1]

[1] Additionally, because it is used after estimation, more extensive modeling features—such as non-linear effects or time-varying covariates—can be incorporated into the use of COX ED.

Method 1: GAM

The GAM approach to COX ED proceeds according to five steps. As is noted above, the first step is model estimation. Then the method computes expected values of risk for each observation by matrix-multiplying the covariates, $\boldsymbol{X}$, by the estimated coefficients from the model, $\hat{\boldsymbol{\beta}}$, then exponentiating the result. This creates $\exp(\boldsymbol{X}\hat{\boldsymbol{\beta}})$, or the exponentiated linear predictor (ELP). Then the observations are ranked from smallest to largest according to their values of the ELP. This ranking is interpreted as the expected order of failure; the larger the value of the ELP, the sooner the model expects that observation to fail, relative to the other observations.

The next step is to connect the model's expected risk for each observation (ELP) to duration time (the observed durations). A GAM fits a model to data by using a series of locally-estimated polynomial splines set by the user (Hastie and Tibshirani, 1990). It is a flexible means of allowing for the possibility of nonlinear relationships between variables. COX ED uses a GAM to model the observed durations as a function of the linear predictor ranks generated in the previous step. More specifically, the method utilizes a cubic regression spline to draw a smoothed line summarizing the bivariate relationship between the observed durations and the ranks (for more details, see Wood, 2006, 2011).[2]

The GAM fit can be used directly to compute expected durations, given the covariates, for each observation in the data. However, for most researchers it is more useful to assess how a change to a particular covariate of interest corresponds to changes in expected duration. In order to examine such marginal changes, it is necessary to create two or more "new" observations corresponding to theoretically-interesting, hypothetical covariate profiles. For example, the analyst might set an indicator variable to 0 and 1 or a continuous variable to a "low" and a "high" value. COX ED allows the covariates in the model to vary naturally over the entire data, then averages over them in the computations.[3] For instance, to estimate the effect of an increase in a covariate X_1 from 0 to 1 on the expected duration, we use the following steps:

(a) Set X_1 to 1 for the entire data (all N observations) and calculate the ELP for every observation, then take an average value of those computations (the median is the default).

(b) Repeat step (a) while setting X_1 equal to 0.

(c) Take the values obtained in steps (a) and (b) and append them to the list of ELP values from the original Cox model in which X_1 is left as exogenous data. Then compute new rankings of the linear predictor values from this list, which is now length $N + 2$.

(d) Pass the list of rankings from step (c) to the GAM as new data to generate expected values. Note that a new GAM is not estimated at this step. Rather, expected durations are generated for each observation—including the two new ones created in steps (a) and (b)—using the previously estimated GAM. This produces point estimates of the expected durations for those two new observations.

(e) Compute the difference between the two estimates obtained in step (d): the expected duration for the data in which X_1 is set to 1 and the expected duration for the data in which X_1 is set to 0. This quantity is a point estimate for the marginal effect, or first difference, corresponding to the change in X_1 from 0 to 1.

Finally, to produce estimates of uncertainty, the GAM approach repeats this process via bootstrapping. The method generates bootstrap samples of the data and re-estimates the Cox model coefficients on each bootstrap sample.[4] At each iteration, this produces a new vector of actual durations and a new ranking of ELP values, which are then used to fit a new GAM. This process results in a distribution of expected durations for each independent variable profile (e.g., step d) and a distribution of the marginal effect (step e). These distributions can be used to produce standard errors and confidence intervals for the estimates.[5] Importantly, by bootstrapping the entire process, this step incorporates the uncertainty from the Cox model estimation *and* the uncertainty from the GAM.

[2] The GAM is fit with the uncensored observations only. If the sample contains a large proportion of censored observations, the NPSF method (see below) may be preferable to the GAM method.

[3] This default option can be changed at the discretion of the analyst.

[4] Standard bootstrapping at the observation level or bootstrapping at the group level (Cameron et al., 2008) are both available.

[5] By default, the method computes the standard errors of each quantity as the standard deviation of its bootstrap distribution. The halfwidth of the confidence interval is then computed by multiplying a tunable critical value based on the standard normal distribution by the standard error. The default critical value is 1.96 (i.e., a 95% confidence interval). Fully non-parametric confidence intervals based on quantiles or bias-corrected quantiles of the bootstrap distribution are also available (see below).

Method 2: Nonparametric step-function

One drawback to the GAM approach is that it uses two statistical models (Cox model and GAM), which yields two sources of estimation uncertainty. An alternative approach comes from the method proposed by Cox and Oakes (1984, 107–109) for estimating the cumulative baseline hazard function. This method is nonparametric and results in a step-function representation of the cumulative baseline hazard; we refer to it as the nonparametric step-function (NPSF) approach.

Cox and Oakes (1984, 108) show that the cumulative baseline hazard function can be estimated after fitting a Cox model by

$$\hat{H}_0(t) = \sum_{\tau_j < t} \frac{d_j}{\sum\limits_{l \in \Re(\tau_j)} \hat{\psi}(l)}, \tag{1}$$

where τ_j represents time points earlier than t, d_j is a count of the total number of failures at τ_j, $\Re(\tau_j)$ is the remaining risk set at τ_j, and $\hat{\psi}(l)$ represents the ELP from the Cox model for observations still in the risk set at τ_j. The NPSF method uses equation (1) to calculate the cumulative baseline hazard at all time points in the range of observed durations with the following steps.

(a) Tied durations are handled by collapsing the dataset by unique duration. The method calculates d_j, the numerator in equation (1), for all time points τ_j by summing the indicator for a non-censored failure within each unique duration ($d_j = 0$ only if all observed durations at τ_j are right-censored). Additionally, it sums the ELPs for all observations with the same duration, because these observations leave the risk set at the same time.

(b) The NPSF approach calculates a running sum, in reverse, for the collapsed ELPs. That is, at the first time point this sum includes the ELP for observations at every time point. At the second time point, this sum includes the ELP for every observation except for those with the earliest observed duration. At the last time point, this sum is equal to the sum of only the ELPs of observations with the latest observed duration. These sums represent the denominator of equation (1).

(c) For each time point, the method divides the number of failures d_j by the sum of ELPs for observations still in the risk set.

(d) Finally, the method calculates the running sum of the ratios we derived in the previous step. This running sum is the non-parametric estimate of the cumulative hazard function.

This procedure yields a stepwise function. Time points with no failures do not contribute to the cumulative hazard, so the function is flat until the next time point with observed failures.

The NPSF approach next obtains expected durations and marginal changes in expected duration by first calculating the baseline survivor function from the cumulative hazard function, using

$$\hat{S}_0(t) = \exp[-\hat{H}_0(t)]. \tag{2}$$

Each observation's survivor function is related to the baseline survivor function by

$$\hat{S}_i(t) = \hat{S}_0(t)^{\hat{\psi}(i)}, \tag{3}$$

where $\hat{\psi}(i)$ is the ELP for observation i. These survivor functions can be used directly to calculate expected durations for each observation. The expected value of a non-negative random variable can be calculated by

$$E(X) = \int_0^\infty \Big(1 - F(t)\Big) dt, \tag{4}$$

where $F(.)$ is the cumulative distribution function for X. In the case of a duration variable t_i, the expected duration is

$$E(t_i) = \int_0^T S_i(t)\, dt, \tag{5}$$

where T is the largest possible duration and $S(t)$ is the individual's survivor function. The NPSF method approximates this integral with a right Riemann-sum by calculating the survivor functions at every discrete time point from the minimum to the maximum observed durations, and multiplying these values by the length of the interval between time points with observed failures:

$$E(t_i) \approx \sum_{t_j \in [0,T]} (t_j - t_{j-1}) S_i(t_j). \tag{6}$$

To calculate a marginal effect, the NPSF approach to COX ED follows the same strategy employed in the GAM approach. It creates two new covariate profiles, setting a variable of interest to two theoretically interesting values. It calculates expected values from each profile, then computes the difference in the two estimates. Finally, the method bootstraps to obtain a standard error and/or confidence intervals for this point estimate.

Implementation in R and empirical example

The methods described above are mostly automated in the package; analysts generally need only a `coxph` model object from the **survival** package (Therneau, 2015) or a `cph` model object from the **rms** package (Harrell, 2018), and, if covariate effects are desired, the name of the variable of interest and the two values of that variable they wish to input.[6] However, the functions also allow for several changes to default settings, such as the formulation of the GAM in the first approach or the computation of confidence intervals.

We illustrate the main features of the package with an empirical example. Martin and Vanberg (2003) examine the determinants of negotiation time among political parties forming coalition governments in Western Europe. The outcome variable in this analysis is the number of days between the beginning and end of the bargaining period. The covariates include the range of government—the ideological distance between the extreme members of the coalition—the number of parties in the coalition, as well as several others. Their main hypotheses predict negative coefficients on the range of government and number of parties variables. They expect that increases in the ideological distance between the parties and the size of the coalition correspond with decreases in the risk of government formation, or longer negotiation times.

The authors demonstrate support for their hypotheses with a sample of data on bargaining in Western European democracies between 1950 and 1995. They estimate a Cox model, then interpret the covariate effects with quantities based in the hazard rate. As an alternative, we employ COX ED with these data. We use the `coxed()` function to predict bargaining duration for every case in the data. Then test the first of their hypotheses by computing estimates of bargaining duration at different values of ideological range of government.

The first step with COX ED is to estimate the model. We estimate the Cox model from Martin and Vanberg (2003) using the `Surv()` and `coxph()` functions from the **survival** package:

```
library(coxed)
data(martinvanberg)

mv.surv <- Surv(martinvanberg$formdur, event = rep(1, nrow(martinvanberg)))
mv.cox <- coxph(mv.surv ~ postel + prevdef + cont + ident + rgovm + pgovno +
                          tpgovno + minority, data = martinvanberg)
```

We report these results in Table 1.

Next we use the GAM version of `coxed()` to examine expected durations and marginal changes in duration.[7] We can calculate standard errors and confidence intervals for any of these quantities with the `bootstrap = TRUE` option. By default the bootstrapping procedure uses 200 iterations (to set this value to a different number, use the `B` argument).[8]

```
ed <- coxed(mv.cox, method = "gam", bootstrap = TRUE, B = 30)
```

[6] Future versions of the software may accept Cox models estimated from other packages, such as **timereg** (Scheike and Zhang, 2011).

[7] For an example of this analysis using the NPSF method, see the vignette for the **coxed** package.

[8] Here we use 30 iterations simply to ease the computational burden of compiling this example. For more reliable results, set `B` to a higher value. There are different methods for calculating a bootstrapped confidence interval. The default method used by `coxed()` (setting the argument `confidence = "studentized"`) adds and subtracts `qnorm(level - (1 - level)/2)` times the bootstrapped standard error to the point estimate, where `level` is the analyst's chosen threshold for evaluating statistical significance. The alternative approach is to take the `(1 - level)/2` and `level + (1 - level)/2` quantiles of the bootstrapped draws, which can be done by specifying `confidence = "empirical"`. We recommend a higher number of bootstrap iterations for empirical confidence intervals. Additionally, the nonparametric bias corrected and accelerated ($\mathrm{BC_a}$) method can be computed with `confidence = "bca"`, which implements the bias correction and acceleration procedure in DiCiccio and Efron (1996) using code modified from the **mediation** package (Tingley et al., 2014).

Table 1: Cox model results from Martin and Vanberg (2003). Entries report coefficients with standard errors in parentheses. These results represent a common approach to presenting Cox model output, but the coefficients themselves are not immediately intuitive.

Range of government	-0.213^{*}
	(0.120)
Number of government parties	1.191^{***}
	(0.124)
Number of government parties	-0.432^{***}
$\times \ln(t)$	(0.035)
Do negotiations commence	-0.577^{***}
immediately after an election?	(0.169)
Did the government take a	-0.100
parliamentary defeat?	(0.230)
Continuation	1.100^{***}
	(0.240)
Identifiability	0.146
	(0.119)
Minority government	-0.428^{**}
	(0.208)
Observations	203
R^2	0.745
Max. Possible R^2	1.000
Log Likelihood	-745.478
Wald Test	218.130^{***} (df = 8)
LR Test	277.239^{***} (df = 8)
Score (Logrank) Test	279.277^{***} (df = 8)

Note: Cell entries report Cox model coefficient estimates with standard errors in parentheses. $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$.

Now every predicted duration has a standard error and a 95% confidence interval. The first several cases' predicted durations are estimated as follows:

```
> head(ed$exp.dur)
    exp.dur bootstrap.se         lb        ub
1 48.978295    5.6915889 37.8229859 60.133605
2 42.036276    4.8132767 32.6024267 51.470125
3 55.440293    6.8188818 42.0755303 68.805056
4 15.734577    1.7119205 12.3792749 19.089880
5  1.530695    0.3512462  0.8422652  2.219125
6 64.449942    7.7421823 49.2755433 79.624340
```

The summary() function, when applied to coxed() output, reports either the mean or median estimated duration along with the bootstrapped standard error and confidence interval for the statistic:

```
> summary(ed, stat = "mean")
   mean bootstrap.se     lb    ub
 28.034        1.998 24.119 31.95
> summary(ed, stat = "median")
 median bootstrap.se     lb     ub
 21.208        2.263 16.773 25.643
```

coxed() can be used to provide duration predictions for observations outside of the estimation sample.

Suppose that we observe five new cases and place them inside a data frame:

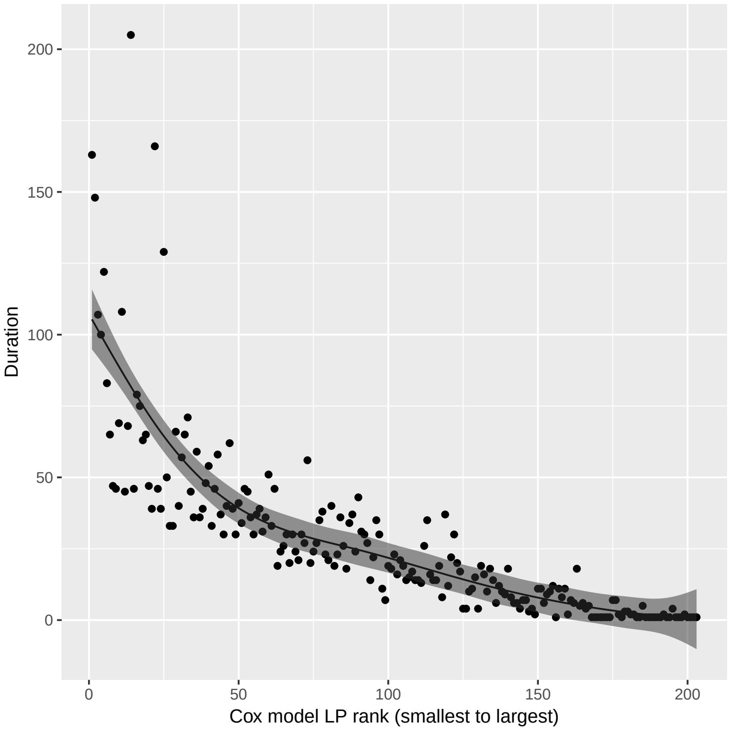
```
new.coalitions <- data.frame(postel = c(1, 1, 1, 0, 1),
                             prevdef = c(0, 0, 1, 1, 0),
                             cont = c(1, 0, 1, 0, 1),
                             ident = c(1, 2, 2, 3, 3),
                             rgovm = c(.3, .8, 1.1, .2, .35),
                             pgovno = c(2, 3, 3, 2, 4),
                             tpgovno = c(3.2, 0, 5, 0, 2.6),
                             minority = c(0, 0, 1, 0, 0))
```

To forecast durations for these cases along with standard errors and confidence intervals, we use the `coxed()` function and place `new.coalitions` into the `newdata` argument:

```
forecast <- coxed(mv.cox, newdata = new.coalitions, method = "gam",
                                                   bootstrap = TRUE, B = 30)
> forecast$exp.dur
    exp.dur bootstrap.se         lb       ub
1 4.5845636    2.7517846 -0.8088352 9.977962
2 0.9542265    0.5656203 -0.1543688 2.062822
3 5.2816962    1.1323172  3.0623953 7.500997
4 1.2358600    0.4684499  0.3177151 2.154005
5 0.5924056    0.7286877 -0.8357961 2.020607
```

The data used by `coxed()` to map rankings to durations are stored in the `gam.data` attribute, and can be used to visualize the fit of the GAM, as in Figure 1.

Figure 1: Mapping duration rankings to observed durations using a GAM. The x-axis plots the ranks of the linear predictor from smallest to largest and the y-axis plots the observed durations. The downward trend shows a non-linear relationship between the model's expectation and the observed data.



We use `coxed()` to provide an answer to the key question, "how much longer will negotiations take for an ideologically polarized coalition as compared to an ideologically homogeneous one?" Specifically, we call `coxed()` and specify two new datasets, one in which `rgovm = 0` indicating that all political parties in the governing coalition have the same ideological position (i.e., a coalition of one party), and one in which `rgovm = 1.24`, indicating that the parties have very different ideological positions.[9] We use mutate() from the **dplyr** package (Wickham et al., 2018) to quickly create new data frames in which rgovm equals 0 or 1.24 for all cases, and set these two data frames as newdata and newdata2 inside coxed().

```
me <- coxed(mv.cox, method = "gam", bootstrap = TRUE, B = 30,
            newdata = mutate(martinvanberg, rgovm = 0),
            newdata2 = mutate(martinvanberg, rgovm = 1.24))
```

coxed() calculates expected durations for all cases under each new data frame and subtracts the durations for each case. To obtain point estimates we can request the mean or median difference.

```
> summary(me, stat = "mean")
             mean bootstrap.se     lb     ub
newdata2   28.927        3.285 22.489 35.365
newdata    25.321        2.632 20.163 30.480
difference  3.605        2.417 -1.133  8.343
> summary(me, stat = "median")
           median bootstrap.se     lb     ub
newdata2   22.392        3.234 16.053 28.730
newdata    19.692        3.449 12.932 26.451
difference  2.928        1.931 -0.857  6.714
```

These results demonstrate that a coalition in which the parties have average ideological differences will take 3.6 more days on average (with a median of 2.9 days) to conclude negotiations than a coalition in which all parties have the same position (i.e., a single-party government).

The NPSF method can be used to compute estimates of these same quantities simply by specifying method = "npsf" in the coxed() function. Additionally, the package includes a function called sim.survdata() designed for simple simulations of duration data that do not assume a distributional form for the baseline hazard. This method, which is fully described in Harden and Kropko (2019), can be useful in several applied and computational settings that involve the Cox model.

Conclusions

The Cox model is popular among applied researchers in a wide range of disciplines due to its inherent flexibility. However, this flexibility makes conveying the substantive meaning of results challenging. By using only the rank ordering of the observed duration times, the Cox model limits researchers to interpreting results in the language of hazard and changes in risk. This yields two key problems. First, it is substantively vague because hazard does not have a meaningful scale. This hinders researchers' capacity to determine whether an estimated effect is substantively "large" or "small." Furthermore, hazard-based interpretations require specialized knowledge to understand. This makes the research less accessible to general audiences, who may be able to learn from the work but cannot due to the means by which results are communicated.

The COX ED methods provide a solution to these problems by allowing researchers to compute duration-based quantities from the Cox model. Communicating results in the language of time allows for more substantive precision and is intuitive to a broad audience of readers. We demonstrate above that COX ED is straightforward to implement in R. The **coxed** package contains functions that allow researchers to use the methods even with minimal knowledge of R. Additionally, the functions are flexible; users can make several changes to many of their features to suit the problem at hand. Finally, the output from the functions provide point estimates, standard errors, and confidence intervals, so researchers can report their results with appropriate measures of uncertainty.

In sum, the **coxed** package provides a useful alternative for researchers to communicate results from the Cox model. It gives them the benefits of the intuitive quantities available in parametric models while retaining the desirable estimation properties of the Cox model. Thus, the analysis can be guided by appropriate modeling choices, but reported in an intuitive, accessible manner.

Bibliography

A. C. Cameron, J. B. Gelbach, and D. L. Miller. Bootstrap based improvements for inference with clustered errors. *Review of Economics and Statistics*, 90(3):414–427, 2008. URL https://doi.org/10.1162/rest.90.3.414. [p2]

[9] Martin and Vanberg (2003) select these values in making hazard rate comparisons. The value rgovm = 1.24 reflects the average ideological range of coalition governments in the sample.

D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972. URL https://doi.org/10.1111/j.2517-6161.1972.tb00899.x. [p1]

D. R. Cox and D. Oakes. *Analysis of Survival Data*. Monographs on Statistics & Applied Probability. Chapman & Hall/CRC, Boca Raton, FL, 1984. [p3]

T. J. DiCiccio and B. Efron. Bootstrap confidence intervals. *Statistical Science*, 11(3):189–228, 1996. URL https://doi.org/10.1214/ss/1032280214. [p4]

J. J. Harden and J. Kropko. Simulating duration data for the Cox model. *Political Science Research and Methods*, 7(4):921–928, 2019. URL https://doi.org/10.1017/psrm.2018.19. [p7]

F. E. Harrell. *rms: Harrell Miscellaneous*, 2018. R package version 5.1–2. http://biostat.mc.vanderbilt.edu/wiki/Main/Rrms. [p4]

T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman & Hall/CRC, Boca Raton, FL, 1990. [p2]

J. Kropko and J. J. Harden. Beyond the hazard ratio: Generating expected durations from the Cox proportional hazards model. *British Journal of Political Science*, 50(1):303–320, 2020. URL https://doi.org/10.1017/S000712341700045X. [p1]

L. W. Martin and G. Vanberg. Wasting time? The impact of ideology and size on delay in coalition formation. *British Journal of Political Science*, 33(2):323–344, 2003. URL https://doi.org/10.1017/S0007123403000140. [p4, 5, 7]

T. H. Scheike and M.-J. Zhang. Analyzing competing risk data using the R timereg package. *Journal of Statistical Software*, 38(2):1–15, 2011. URL http://dx.doi.org/10.18637/jss.v038.i02. [p4]

T. Therneau. *survival: A Package for Survival Analysis in S*, 2015. R package version 2.38. [p4]

D. Tingley, T. Yamamoto, K. Hirose, L. Keele, and K. Imai. mediation: R package for causal mediation analysis. *Journal of Statistical Software*, 59(5):1–38, 2014. URL http://dx.doi.org/10.18637/jss.v059.i05. [p4]

H. Wickham, R. François, L. Henry, and K. Müller. *dplyr: A Grammar of Data Manipulation*, 2018. URL https://CRAN.R-project.org/package=dplyr. R package version 0.7.6. [p7]

S. N. Wood. *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC, Boca Raton, FL, 2006. [p2]

S. N. Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 73(1):3–36, 2011. URL https://doi.org/10.1111/j.1467-9868.2010.00749.x. [p2]

Jonathan Kropko
University of Virginia
School of Data Science
Dell 1 Building
Charlottesville, VA 22904
jkropko@virginia.edu

Jeffrey J. Harden
University of Notre Dame
Department of Political Science
2055 Jenkins Nanovic Halls
Notre Dame, IN 46556
jeff.harden@nd.edu