12-2020

Survival Analysis: An Exact Method for Rare Events

Kristina Reutzel
Utah State University

SURVIVAL ANALYSIS: AN EXACT METHOD FOR RARE EVENTS

by

Kristi Reutzel

A project submitted in partial fulfillment

of the requirements for the degree

of

MASTER OF SCIENCE

in

STATISTICS

Approved:

Christopher Corcoran, PhD Major Professor	Yan Sun, PhD Committee Member
Daniel Coster, PhD Committee Member	

Abstract

Conventional asymptotic methods for survival analysis work well when sample sizes are at least moderately sufficient. When dealing with small sample sizes or rare events, the results from these methods have the potential to be inaccurate or misleading. To handle such data, an exact method is proposed and compared against two other methods: 1) the Cox proportional hazards model and 2) stratified logistic regression for discrete survival analysis data.

Background and Motivation

Survival analysis models are used for data in which the interest is time until a specified event. The event of interest could be heart attack, onset of disease, failure of a mechanical system, cancellation of a subscription service, employee termination, etc. Survival analysis models are used in a variety of applications, including biostatistics, social sciences, economics, and engineering. One important distinguishing characteristic of survival analysis models is the presence of censoring. Censoring arises when the event time is unknown, or if it is not known whether or not the event occurred. Survival analysis has been widely studied and applied across many areas of the biomedical and social sciences. There are many excellent text books containing more details about the application of survival analysis models. For example, Singer and Willett (2003) cover both discrete and continuous time models in *Applied Longitudinal Data analysis*; while Tutz and Schmid (2018) focus on discrete time models in their text *Modeling Discrete Time-to-Event Data.* Kleinbaum and Klein (2012) offer a self-study approach in *Survival Analysis: A Self-Learning Text.*

As with other modeling approaches in statistics, some of the conventional methods that are used for survival analysis are hampered when sample sizes or event rates are small. As a result, there has been significant attention paid to alternatives for small or sparse samples. For example, the asymptotic log-rank test requires the hazards to be proportional, while the generalized Wilcoxon test is a good alternative when the proportional hazards assumption is not met. Both tests require that groups have similar sample sizes, but are still valid in unequal follow-up situations. In situations where the group sizes are markedly different and follow-up is equal, complete permutation tests are available, including the early generalized Wilcoxon test by Gehan (1965), and the modified Savage test (Schemper, 1984). Heinze, Gnant, Schemper (2003) propose two exact tests that are suitable for use in cases where: group sizes are different, there are a low number of events, and groups have unequal follow-up times. Based on simulations, the exact test (conditional on follow-up) has been shown to be the best choice when follow-up is unequal. When follow-up is equal, the exact complete permutation has a slight advantage in terms of performance. These tests are designed to detect differences in survival curves, but do not provide estimates for the hazard rate.

Wang, Lagakos, and Gray propose two types of permutation tests for accelerated failure time (AFT) models when sample sizes are small (2010); as interval estimates tend to perform poorly for small samples. The methods proposed rely on first imputing the survival and censoring times, after which the the permutation methods are applied. Based on a simulation study, the two methods have good Type I error and power properties. Additionally, the confidence intervals constructed under the AFT model have better coverage than the approach of Jin and others (2003) when used with small samples.

Samuelsen (2003) proposes a method where survival data is analyzed using a conditional logistic model. So-called Type I and Type II censoring are considered, where type I censoring refers to an experiment that ends after a pre-specified number of years, and type II censoring refers to an experiment that ends after a pre-specified number of events have occurred. The results of his simulation study show that the type I censored data yields conservative coverage, and lower power than score and likelihood ratio tests; while the type II censored data also yields conservative coverage, although not as conservative as the type I results. Power was demonstrably better for the exact test than the Wald test. Additionally, coverage and power were evaluated with one covariate in the model.

Rabinowitz, Betensky, and Tsiatis (2000) present an approach to survival analysis using a conditional logistic regression to fit a proportional odds model to interval-censored data. Shih and Fay (1999) consider permutation tests for stratified survival data by permuting scores based on a functional of estimated distribution functions.

Methodology

The Cox proportional hazards model is the most widely applied model for time-to-event data. One advantage of the Cox model is that both continuous and categorical predictors can be considered in the same model. Another advantage is that the model is semi-parametric, meaning that the baseline hazard function, $h_0(t)$, does not need to be known. The hazard function for continuous time survival models can be interpreted as the instantaneous rate of failure and, for the Cox model, can be expressed as

$$h(t) = h_0(t)\exp[\beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p]$$

Where $h_0(t)$ is the baseline hazard with p predictors $(X_1, X_2, \cdots, X_p)$.

The estimates for β_i are obtained through a so-called partial likelihood function. These estimates are obtained in a similar manner to traditional maximum likelihood estimates, and assume that characteristics of the partial likelihood function are likewise asymptotically normally distributed (Singer and Willett, 2003). Wald-type confidence intervals are thus applied to regression coefficients and are typically the default in software programs such as R and SAS. When sample sizes are small and/or there are few observed events, asymptotic properties of the partial likelihood function may not hold. To handle this issue within the framework of a Cox model, profile likelihood intervals have been suggested and tend to yield more robust results over Wald confidence intervals (Heinze and Schemper, 2002). This is of particular consequence for adaptive design clinical trials, where sample sizes can be small due to the choice of design. For example, in a group-sequential clinical trial design, the earlier phases will have a small number of events, and then increase at pre-specified times during the course of the trial. During the earlier phases, the Type I error rate may be inflated if a conventional large-sample method is used.

When the event rate is low, discrete survival models are another good option for analyzing survival data. The proposed method in this paper is an exact method for discrete survival analysis (DT-E). In order to analyze the data properly, it must be in person-period format. This is done in R using the longFormat function from the discSurv package. In this format, each subject will have multiple observations. Because the data are gathered over time (logitudinal) rather than just a snapshot in time (cross-sectional), this format will appropriately allow each subject to contribute to the risk set for each time period until the subject is censored or experiences the event. The probability of an event at time t is conditional on the subject surviving up until time t, resulting in conditional independence.

With the data appropriately formatted, the usual method for discrete survival analysis is to then perform logistic regression, with the model expressed as

$$\log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \gamma_i + x'_{ij}\boldsymbol{\beta},$$

where $i = 1, 2, \dots, N$ indicate the strata, and $j = 1, 2, \dots, n_i$ indicates each of the n_i subjects within stratum i. Let $Y_{ij} = 1$ if the jth subject in the ith stratum experienced the event, and $Y_{ij} = 0$ otherwise. π_{ij} is then defined as $\Pr(Yij = 1|x_{ij})$ where x_{ij} is a vector of covariates with p dimensions for the jth subject in the ith stratum, γ_i is a scalar parameter of the ith strata, and $\boldsymbol{\beta}$ is a $(p \times 1)$ vector of parameters. When the time measurements are considered to be discrete, but rounded due to measurement restrictions, the logit model estimates converge to a Cox proportional hazards model (Thompson, 1977). In the discrete time model, the odds ratio is used; while in the Cox Proportional Hazards model, we use the hazard ratio.

Based on this logistic regression model, two separate approaches are explored. The first approach, which will be referred to as DT-A throughout the paper, will rely on stratifying on all parameters that are not of interest, including time. This method conditions out the nuisance parameters on their sufficient statistics to calculate estimates only for the parameters of interest and relies on asymptotic assumptions of the MLE.

The second approach, which will be referred to as DT-E throughout the paper, uses exact conditional logistic regression. Like DT-A, all nuisance parameters are conditioned out, but rather than relying on the asymptotic properties of the MLE, this method relies on the exact distribution for only the parameter of interest to obtain the estimate for β. All simulations and examples were analyzed using Cytel Studio LogXact software. LogXact is a statistical package for regression procedures featuring exact methods. LogXact is an industry standard in clinical trials because they use highly efficient algorithms for exact methods. However, other widely used software packages such as R or SAS also have the capability to fit an exact conditional regression model for discrete time survival data if the data are formatted properly.

Simulation

A variety of packages have been developed for simulating survival data in R. Common approaches for simulating survival data involve sampling from specified distributions (e.g. exponential, Weibull, Gompertz) that assume a particular shape for the baseline hazard. Note that with the Cox model, we assume that the baseline hazard is from a family of parametric distributions that yield proportional hazards. However, the Cox model allows you to model data without assuming a particular shape for the baseline hazard. For this simulation study, rather than drawing from a specific parametric distribution, a randomly generated baseline hazard is created for each simulated dataset, using the coxed package (Harden and Kropko, 2018). Applying this method makes the simulation applicable to a wider variety of data, increasing the generalizability of the simulation study. Harden and Kropko (2018) describe their method (noting that T = maximum duration value possible):

> This function employs the flexible hazard method to generate a baseline failure CDF: it plots points at (0, 0) and (T+1, 1), and it plots knots additional points with x-coordinates drawn uniformly from integers in [2, T] and y-coordinates drawn

from U[0, 1]. It sorts these coordinates in ascending order (because a CDF must be non-decreasing) and if spline=TRUE it fits a spline using Hyman's (1983) cubic smoothing function to preserve the CDF's monotonicity. Next it constructs the failure-time PDF by computing the first differences of the CDF at each time point. It generates the survivor function by subtracting the failure CDF from 1. Finally, it computes the baseline hazard by dividing the failure PDF by the survivor function.

Two examples of these randomly generated hazard functions used for simulation our simulation study in Figures 1 and 2 are shown below with their accompanying PDFs, CDFs, and Survivor functions.

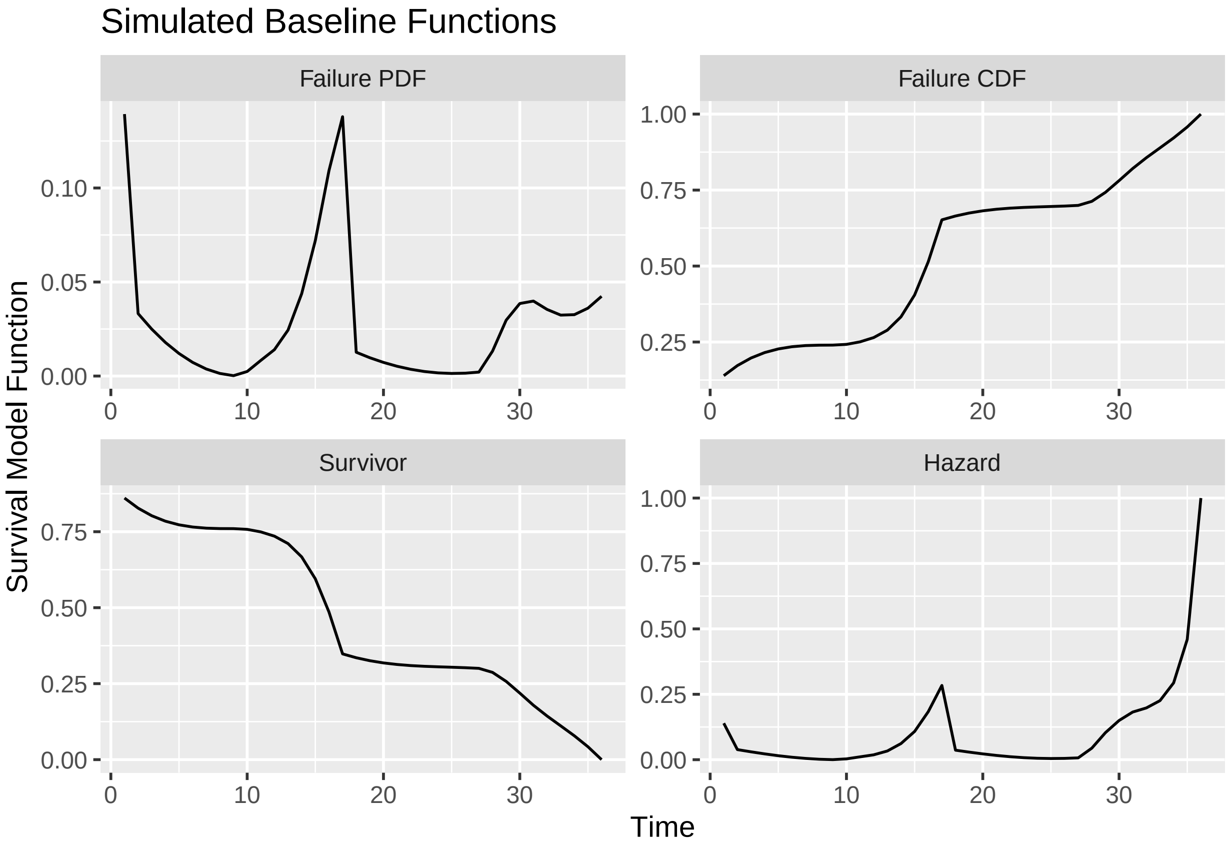

Figure 1: Example 1

For our study, we used three sample sizes (N = 40, 80, 200), two levels of right censoring (50% and 80%), and two binary predictor variables. The first predictor variable could assume one of four values: $\beta_1 = \log(\theta) = 0, 0.6, 1.2, 1.8$, where θ is the true hazard. The second

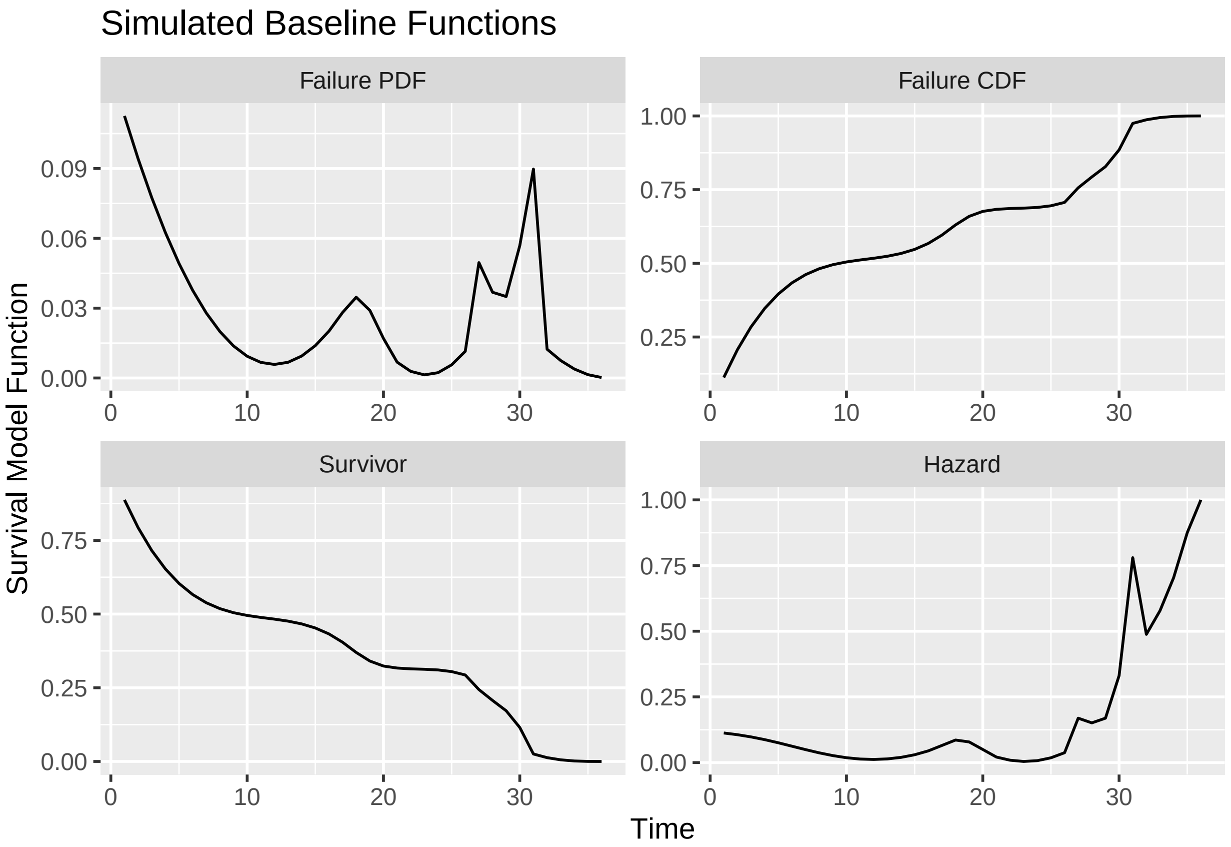

Figure 2: Example 2

predictor was assigned $\beta_2 = \log(\theta) = 0$. For each sample size, level of censoring, and value for $\log(\theta)$, 5000 datasets were simulated. The censoring was assumed to be uninformative, i.e., independent of failure time. The times are assumed continuous, but rounded due to coarseness of the measurements. The first predictor, x_1, has $\frac{N}{2}$ observations belonging to group 1, where N is the sample size. The second predictor, x_2, was randomly selected, with probability $p = 0.5$ for each of the two groups.

Three methods were applied: Cox porportional hazards model, DT-A, DT-E. The parameter estimates for the Cox proportional hazards model are obtained by maximizing the partial likelihood. Theoretically, event times are considered continuous so no two event times will be the same. However, in practice, measurement rounding can produce ties. The partial likelihood equation is valid only when there are no tied event times. When ties do occur, the two methods most commonly considered for approximation come from Breslow (1974) and Efron (1977). The data in this simulation study are analyzed using a Cox Proportional Hazards model with Efron's approximation. Efron's approximations are used in this study, as Breslow's approximation does not perform as well when there are a relatively large number of ties in the risk set but, when there are few to moderate ties, results tend to yield similar coefficients (Allison, 2010). Given that the sample sizes are relatively small, we used profile likelihood confidence intervals.

Results

The DT-E method retains 95% coverage in nearly all data sets, only dropping slightly below in three instances of the larger sample (N = 200) simulations. For the 50% censoring level and $\beta = 1.2$ and $\beta = 1.8$, the coverage drops just below the desired coverage level, to 94.96%. For the 80% censoring level and $\beta = 1.8$, the coverage drops to 93.82%. See figure 3 and table 1 for the coverage for each method, at each level of censoring, beta, and sample size.

Table 1: Percent Coverage

		Censoring = 50%				Censoring = 80%			
N	Method	0.0	0.6	1.2	1.8	0.0	0.6	1.2	1.8
40	DT-E	96.76	97.26	97.60	96.92	97.94	97.90	98.28	97.96
	DT-A	95.06	95.82	96.38	95.68	98.02	97.86	97.76	95.34
	Coxph	93.90	94.84	94.34	93.14	95.26	95.44	95.90	94.54
80	DT-E	96.64	96.68	96.76	96.34	97.84	97.04	96.82	96.38
	DT-A	95.16	95.08	95.12	94.62	96.74	95.92	95.40	93.68
	Coxph	94.68	94.00	93.14	91.80	95.26	94.38	93.66	92.22
200	DT-E	96.52	96.16	94.96	94.96	96.68	96.14	95.42	93.82
	DT-A	95.60	95.10	93.74	93.30	95.46	94.84	93.70	91.00
	Coxph	95.42	94.00	90.92	86.94	94.94	94.10	92.82	89.60

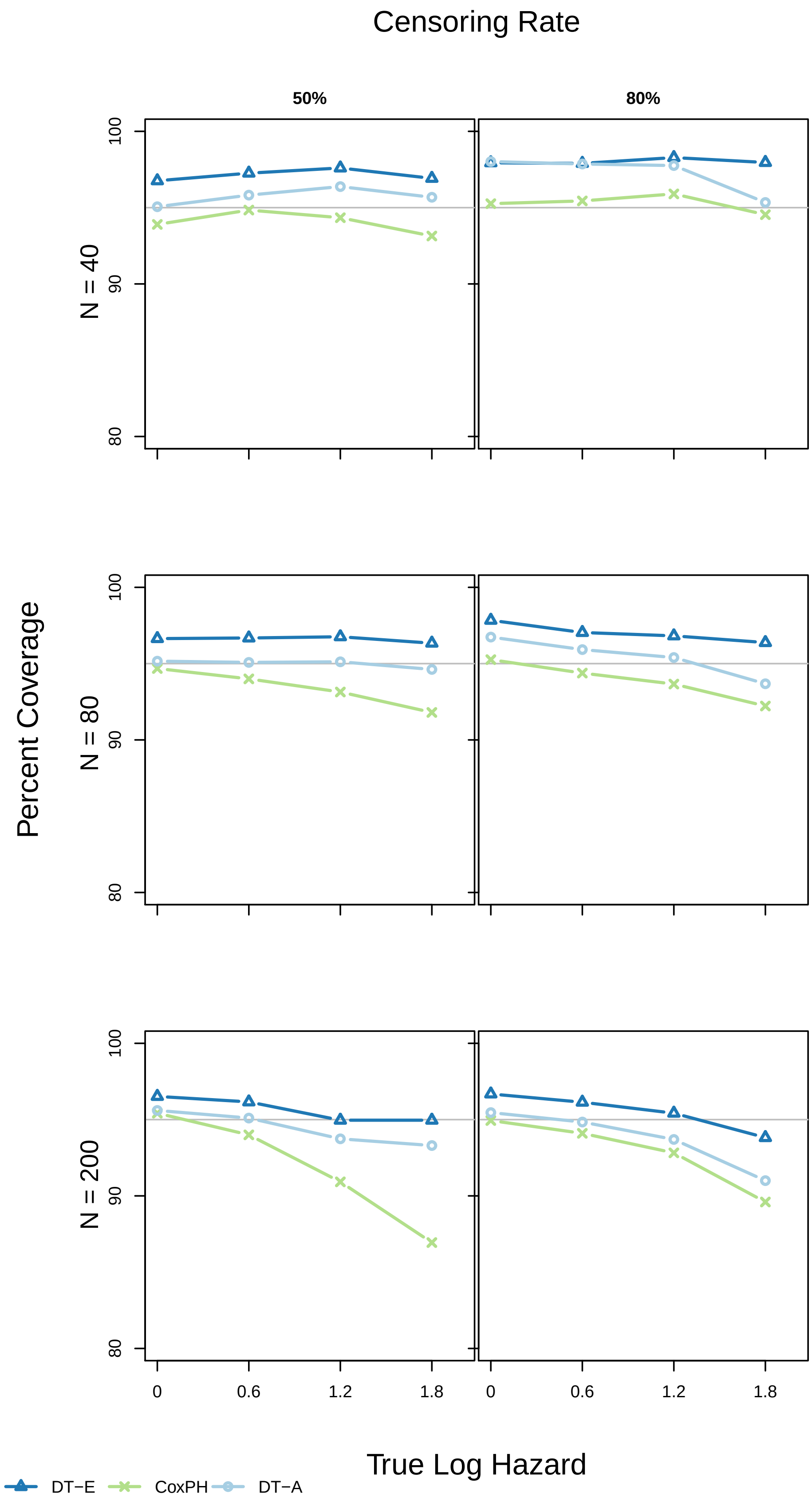

Figure 3: Coverage

One thing to note about performance is that both the DT-A and Cox model have issues with nonconvergence of the likelihood function when sample sizes are small (N = 40) and censoring is high (80%). The most notable instances are when $\beta = 1.2$, the DT-A model does not achieve convergence 8.26% of the time, and the Cox model does not achieve convergence 7.84% of the time. When $\beta = 1.8$, the DT-A model does not converge in 14.84% of instances, and the Cox model does not converge in 15.2% of instances. Under these circumstances, the estimate is not reliable. Tests that do not converge are excluded from the results in figure 3, and table 1. Percentages of nonconvergence for sample sizes N = 40, 80 are reported in table 2. Convergence is a non-issue for all values of β and censoring levels when N = 200, and are consequently not reported in table 2.

Figure 4 illustrates that average bias for all three tests are similar.

To handle issues with nonconvergence, SAS offers an option to run the Firth correction, which uses the penalized partial likelihood to calculate estimates (Heinze & Dunkler, 2008). However, this option is not available with Efron's approximation. As an initial look into the estimates, 1000 datasets were simulated with a sample size of 40, $\beta = 1.8$, and an 80% censor rate. Of the 1000 datasets, roughly 15% of the tests did not converge using the Cox Proportional Hazards model. This is as expected, based on table 2. Average estimates and coverage were then calculated on the 15%.

- Exact test: Mean $\beta = 2.326671$, coverage = 99.34%
- Firth Corrected: Mean $\beta = 2.604367$, coverage = 97.37%

Coverage is conservative, which is not surprising given the large confidence intervals of both the Firth method and the DT-E method. In the cases where nonconvergence is an issue, the DT-E test yields one-sided confidence intervals. Both tests give beta estimates that are high, but the exact tests perform slightly better in terms of bias. The average exact test gives a beta value of 2.3, which translates to a hazard ratio of roughly 10; while the average Firth test gives a beta value of 2.6, which translates to a hazard ratio of roughly 13.5. This is compared to the target value for beta of 1.8, or a hazard ratio of 6.

There are two notable limitations of this initial look into the Firth option. First, the methods are not strictly comparable since Breslow's method is used in conjunction with the Firth option, but all other tests use Efron's. Second, the sample size is too small to reach any definite conclusions. Further analyses are recommended in order to better understand how the Firth correction compares to the results of the exact test.

Table 2: Percentage of Nonconvergence

		$\log(\theta) = \beta$							
		N = 40				N = 80			
Censoring	Method	0.0	0.6	1.2	1.8	0.0	0.6	1.2	1.8
50%	DT-E	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00
	DT-A	0.00	0.00	0.20	1.24	0	0.00	0.00	0.00
	Coxph	0.00	0.00	0.18	0.92	0	0.00	0.00	0.00
80%	DT-E	0.00	0.00	0.00	0.00	0	0.00	0.00	0.00
	DT-A	3.08	3.10	8.26	14.84	0	0.04	0.38	1.70
	Coxph	3.20	3.24	7.84	15.20	0	0.04	0.38	1.64

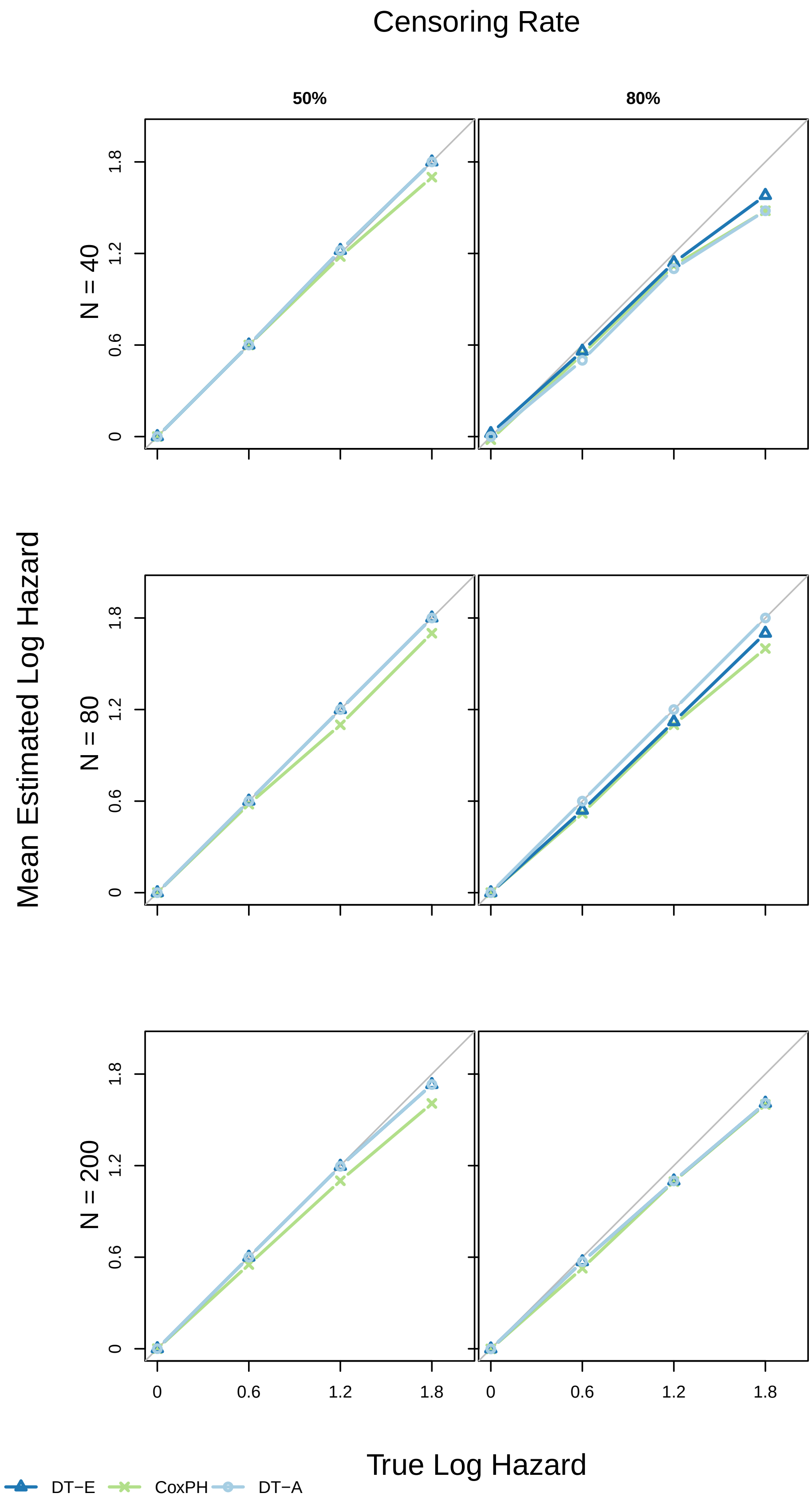

Figure 4: Bias

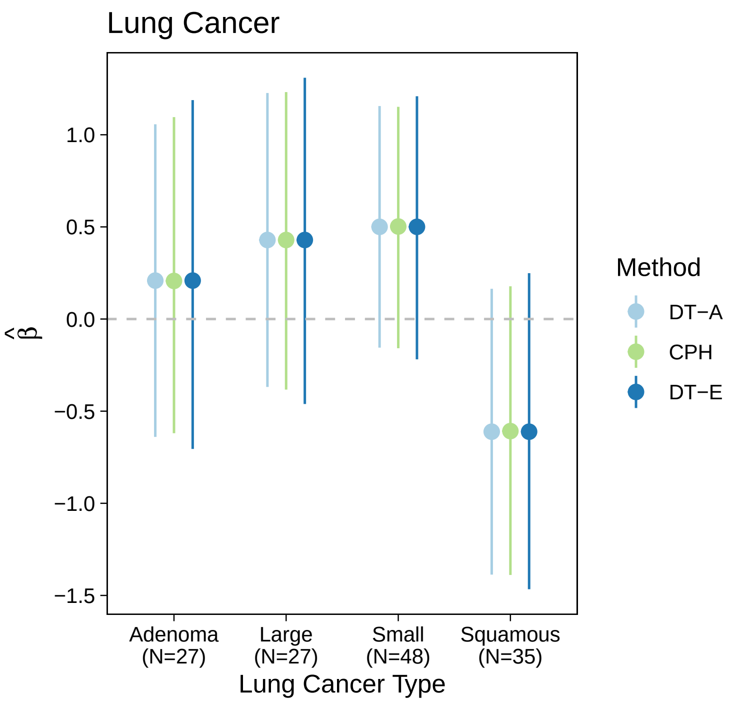

Figure 5: Estimates by Method and Tumor Type

Application

Three datasets from are analyzed using the Cox proportional hazards method, DT-A method, and DT-E method.

Lung Cancer

The dataset used comes from a study on lung cancer (Kalbfleisch and Prentice, 1980) and can be accessed through the R package ncvreg. There are 137 observations and 9 predictors, where 69 subjects were assigned to the control group, and 68 patients assigned to the treatment group. The type of tumor was recorded for each patient: adenoma, large cell, small cell, and squamous. Time until death was recorded in days.

For the purpose of analysis, the observations were divided into groups based on their tumor type. The two methods give similar estimations (see figure 5), which is consistent with what was observed in the simulations when β is close to zero.

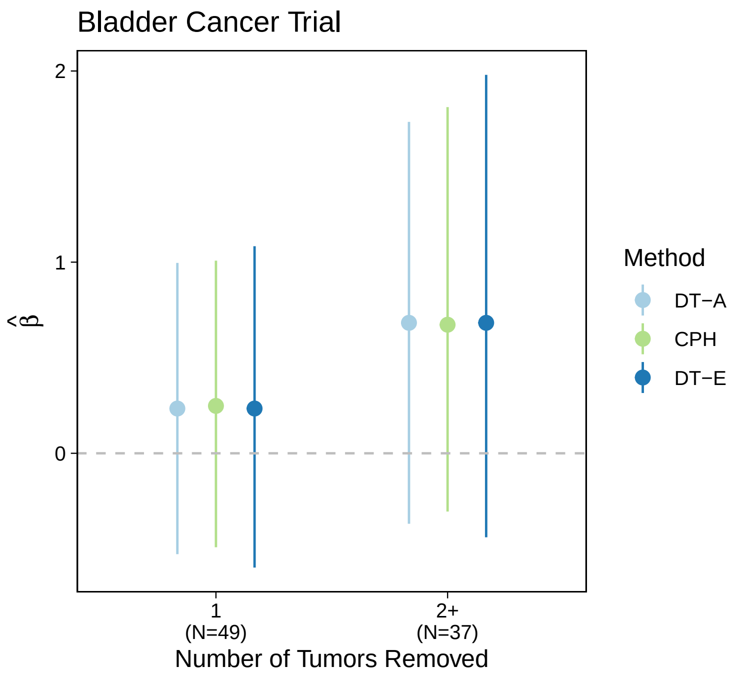

Figure 6: Estimates by Method and Number of Tumors Removed

Bladder Cancer

Data for this example come from a study of bladder cancer discussed by Pagano and Gauvreau (2000) After surgery to have one or more tumors removed, 86 patients were either treated with chemotherapy (Thiotepa), or given a placebo. Of the 86 patients, 48 were assigned to the placebo group, and 38 were given chemotherapy. Patients were then divided into two groups based on how many tumors were removed: one tumor removed vs. multiple tumors removed. Time until reocurrence is recorded in months. Table 3 gives a breakdown of patients by treatment and number of tumors removed.

Table 3:

	Placebo	Chemotherapy
One Tumor	26	23
Two Or More Tumors	22	15

Three methods were then applied to each of two subgroups based on number of tumors

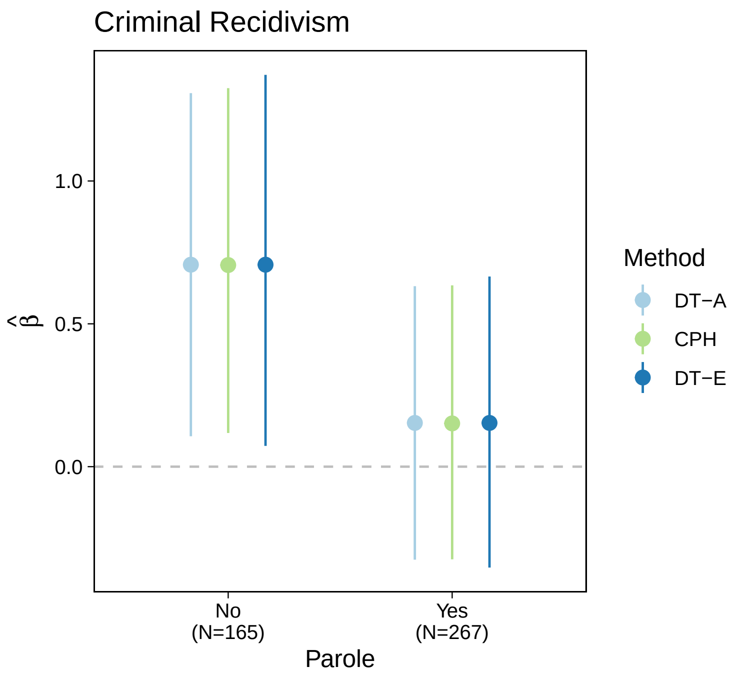

Figure 7: Estimates by Method and Parole Status

removed. All three methods give similar estimates for the log of the hazard (see figure 6), which is what we would expect based on the simulations.

Recidivism

The recidivism dataset used here is obtained from the carData package in R, and originally published by Rossi et. all (1980). The dataset contains information from 432 prisoners released from prison, of which half were given financial aid. Subjects were followed up for one year after their release date. Time of first arrest after release is recorded in weeks.
The observations were split into two groups: those who were on parole (N = 267), and those who were not (N = 165). The censoring rates of those who are on parole is 75%, and 72% for those who were not on parole. The aim of the analysis is to determine if the financial aid program reduced recidivism rates. As shown in figure 7, all three methoded give similar estimates for β, with the exact method producing slightly larger confidence intervals.

Conclusion

The DT-E method appears to perform well in the presence of high rates of censoring, small samples, and situations with non-convergence. While not the main objective of this project, we also conducted a small comparative investigation of bias where the Cox Proportional Hazards model ran into problems with non-convergence. Some limited results indicated that, in the presence of a high rate of censoring and/or small samples, DT-E also yielded less biased estimates than the Cox model. However, a more definitive conclusion about this would require a larger simulation study. Simulations and applications for the compared methods consider only data with binary covariates and type I censoring.

References

- Allison, Paul D. (2010). *Survival analysis using SAS: A practical guide, second edition.* Cary, NC: SAS Institute Inc.

- Breslow, N. (1974). Covariance analysis of censured survival data. *Biometrics*, 30, 89–99.

- Efron, B. (1977). The Efficiency of Cox's Likelihood Function for Censored Data, *Journal of the American Statistical Association.* 76, 312-319.

- Fox, J.; Weisberg, S.; and Price, B. (2018). *carData: Companion to Applied Regression Data Sets.* R package version 3.0-2. https://CRAN.R-project.org/package=carData

- Gehan, E. A. (1965). A generalized Wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika*, 52, 203-223.

- Harden, J. and Kropko, J. (2018) Simulating Duration Data for the Cox Model, *Political Science Research and Methods*, 7(4), 921-928, doi: 10.1017/psrm.2018.19.

- Heinze, G. and Dunkler, D. (2008). Avoiding Infinite Estimates of Time-Dependent Effects in Small-Sample Survival Studies. *Statistics in Medicine*, 27(30), 6455–6469. doi: 10.1002/sim.3418.

- Heinze, G. and Schemper, M. (2002). A solution to the problem of separation in logistic regression. *Statistics In Medicine*, 21(16):2409-2419. http://search.ebscohost.com.dist.lib.usu.edu/login.aspx?direct=true&db=cmedm&AN=12210625&site=eds-live. Accessed May 15, 2019.

- Kleinbaum, D. G. and Klein, M. (2012). *Survival analysis: A self-learning text, third edition.* New York, NY: Springer.

- Pagano, M. and Gauvreau, K. (2000). *Principles of Biostatistics, second edition.* Duxbury. Chapter 21, exercise 9, page 512.

- R Core Team (2018). **R:** *A language and environment for statistical computing.* R

Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

- Rossi, P.H.; Berk, R.A.; and Lenihan, K.J. (1980). *Money, Work, and Crime: Some Experimental Results.* New York: Academic Press.
- Schemper, M. (1984). A survey of permutation tests for censored survival data. *Communications in Statistics—Theory and Methods*, 13, 1655–1665.
- Singer, J. D. and Willett, J. B. (2003). *Applied longitudinal data analysis modeling change and event occurrence.* New York: Oxford University Press.
- Thompson, W. (1977). On the Treatment of Grouped Observations in Life Studies. *Biometrics*, 33(3), 463-470. doi:10.2307/2529360
- Tutz, G. and Schmid, M.(2018). *Modeling Discrete Time-to-Event Data.* Cham: Springer International Publishing.
- Welchowski, T. and Schmid, M. (2018). **discSurv:** *Discrete Time Survival Analysis.* R package version 1.3.4. https://CRAN.R-project.org/package=discSurv
- Xu, R.; Shaw, P.A.; and Mehrotra, D.V. (2018) Hazard Ratio Estimation in Small Samples, *Statistics in Biopharmaceutical Research*, 10(2), 139-149, doi: 10.1080/19466315.2017.1369899

Code Appendix

The following code generates 1000 datasets for beta = 0 at 50% censoring with 200 observations in each dataset.

```
seed <-  47347 # seed_N200_5
cens <- 80 # change this when running for 50% censoring

n <- 200 # sample size (n/2 is the size in each group)
top <- 36 # possible duration time limit
size <- 1000 # number of data frames
betaslist <- list(
  'b00' = c(0, 0),
  'b06' = c(0.6, 0),
  'b12' = c(1.2, 0),
  'b18' = c(1.8, 0)
)

# User Created Function:
longFormat <- function(x){
  longdf <- lapply(x, function(x) dataLong(x, 'time', 'event', timeAsFactor = FALSE))
  longdf_all <- bind_rows(longdf, .id = 'group')
  longdf_all$group <- as.numeric(longdf_all$group)
  # longdf_all$timeInt <- as.numeric(longdf_all$timeInt)
  longdf_all <- longdf_all[, -9]
  return(longdf_all)
}
```

```
# create the groups for x1 (binary covariate)
# do this way to get the groups ordered together
x1 <- c(rep(0, n/2), rep(1, n/2))
set.seed(2) # don't need to reset this one for each of the 5 iterations of simulations
x2 <- sample(c(c(rep(0, n/2), rep(1, n/2))), size = n, replace = FALSE)
x <- data.frame(x1, x2)

# the following code returns a list of lists, one list for each level of beta, and
# each of those lists has a list of the generated info that includes the betas, data
# frame, simulated values, baseline hazard, etc.

set.seed(seed)
simdata200 <- list()
for(i in 1:length(betaslist)) {
  simdata200[[i]] <- sim.survdata(
    N = n,
    T = top,
    num.data.frames = size,
    X = x,
    beta = betaslist[[i]],
    censor = cens/100
  )
}

### create lists of 1000 dataframes ###

simdata200_b00 <- list()
for (i in 1:size) {
  # list of dataframes
  simdata200_b00[[i]] <- simdata200[[1]][[i]]$data
  simdata200_b00[[i]]$group <- i

  # clean up to prepare for formatting
  colnames(simdata200_b00[[i]]) <-
    c('x1', 'x2', 'time', 'event', 'group')
  simdata200_b00[[i]]$event <-
    ifelse(simdata200_b00[[i]]$event == FALSE, 0, 1)
}

# format for SAS
simdata200_b00_ph <- bind_rows(simdata200_b00, .id = 'group')
simdata200_b00_ph$group <- as.numeric(simdata200_b00_ph$group)

# write to csv
write.csv(simdata200_b00_ph, file = 'simdata200_b00_ph.csv')

# convert to longFormat
simdata200_b00_long <- longFormat(simdata200_b00)

# write to csv
write.csv(simdata200_b00_long, file = 'simdata200_b00_long.csv')


#####################################
####     PHREG done in SAS       ####
####     Profile Likelihood      ####
```

```
####     Confidence Intervals    ####
#####################################

# import results back into R using the sas7bdat package

names1 <-
  c('group',
    'pl_betax1',
    'pl_sex1',
    'pl_pvaluex1',
    'pl_lowerx1',
    'pl_upperx1')
PLresults200_cens50_x1 <- list()
for (i in 1:length(betaslist)) {
  PLresults200_cens50_x1[[i]] <-
    read.sas7bdat(paste0('selected200_', names(betaslist)[[i]], '_ph.sas7bdat'))
  PLresults200_cens50_x1[[i]] <- PLresults200_cens50_x1[[i]][,-2]

  colnames(PLresults200_cens50_x1[[i]]) <- names1

  PLresults200_cens50_x1[[i]] <- PLresults200_cens50_x1[[i]] %>%
    mutate(
      pl_coveragex1 = case_when(
        pl_lowerx1 <= exp(betaslist[[i]][[1]]) &
          pl_upperx1 >= exp(betaslist[[i]][[1]]) ~ 1,
        pl_betax1 == 'NaN' |
          pl_sex1 == 'NaN' | pl_lowerx1 == 'NaN' ~ 0,
        TRUE ~ 0
      ),
      pl_usefulx1 = case_when(
        pl_betax1 == 'NaN' | pl_upperx1 == 'NaN' | pl_lowerx1 == 'NaN' ~ 0,
        TRUE ~ 1
      )
    )
}

#####################################################
####    Analysis done in LogXact through SAS    ####
#####################################################


# Stratified ELR - stratify on timeInt and x2
# interested in asymptotic (MLE) and exact (CMLE) estimates
# import results back into R


names <- c('group', 'betax1', 'sex1', 'Type', 'lowerx1', 'upperx1', 'pvaluex1')
SASresults200_cens50_x1 <- list()

for (i in 1:length(betaslist)) {
  setwd('~/Documents/SimulationData/cens50')
  SASresults200_cens50_x1[[i]] <-
    read.sas7bdat(paste0('selected200_', names(betaslist)[[i]], '_strat.sas7bdat'))
  SASresults200_cens50_x1[[i]] <- SASresults200_cens50_x1[[i]] %>%
    select(-Effect) %>%
    filter(Type2 != 'Ex-MidP')
  SASresults200_cens50_x1[[i]]$LowerCI <-
    exp(SASresults200_cens50_x1[[i]]$LowerCI)
  SASresults200_cens50_x1[[i]]$UpperCI <-
    exp(SASresults200_cens50_x1[[i]]$UpperCI)

  colnames(SASresults200_cens50_x1[[i]]) <- names

  SASresults200_cens50_x1[[i]] <- SASresults200_cens50_x1[[i]] %>%
    mutate(
      coverage = case_when(
        lowerx1 <= exp(betaslist[[i]][[1]]) &
          upperx1 >= exp(betaslist[[i]][[1]]) ~ 1,
        betax1 == 'NaN' | sex1 == 'NaN' | lowerx1 == 'NaN' ~ 0,
        TRUE ~ 0
      ),
      useful = case_when(betax1 == 'NaN' |
                           upperx1 == 'NaN' | lowerx1 == 'NaN' ~ 0,
                         TRUE ~ 1)
    )

}


### coverage
coverage200_cens50_elr <- vector()
coverage200_cens50_strat <- vector()
coverage200_cens50_pl <- vector()
for (i in 1:length(betaslist)) {
  coverage200_cens50_elr[i] <-
    mean(SASresults200_cens50_x1[[i]]$coverage[SASresults200_cens50_x1[[i]]$useful == 1
                                             & SASresults200_cens50_x1[[i]]$Type ==
                                               'Exact']) * 100
  coverage200_cens50_pl[i] <-
    mean(PLresults200_cens50_x1[[i]]$pl_coverage
         [PLresults200_cens50_x1[[i]]$pl_useful == 1]) * 100
  coverage200_cens50_strat[i] <-
    mean(SASresults200_cens50_x1[[i]]$coverage[SASresults200_cens50_x1[[i]]$useful == 1
                                             & SASresults200_cens50_x1[[i]]$Type ==
                                               'AS']) * 100
}

coverage200_cens50_elr
round(coverage200_cens50_strat, 1)
round(coverage200_cens50_pl, 1)




### Bias
bias200_cens50_elr <- vector()
bias200_cens50_strat <- vector()
bias200_cens50_pl <- vector()
for (i in 1:length(betaslist)) {
  a <- PLresults200_cens50_x1[[i]] %>%
    filter(pl_usefulx1 == 1)
  bias200_cens50_pl[i] <- mean(a$pl_betax1)
  a <- SASresults200_cens50_x1[[i]] %>%
    filter(Type == 'Exact', useful == 1)
  bias200_cens50_elr[i] <- mean(a$betax1)

  a <- SASresults200_cens50_x1[[i]] %>%
    filter(Type == 'AS', useful == 1)
  bias200_cens50_strat[i] <- mean(a$betax1)
}

round(bias200_cens50_elr, 1)
round(bias200_cens50_strat, 1)
round(bias200_cens50_pl, 1)


## SAS code

## /* 50% CENSORING RATE */


PROC IMPORT OUT= WORK.SIMDATA40_b00_PH
            DATAFILE=
              "\\Mac\Home\Documents\SimulationData\AllSims\cens50\simdata40_b00_ph.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;
ODS HTML;
title 'EFRON WITH PROFILE LIKELIHOOD CI';
```

```
title2 'N = 40, Beta = 0';

PROC PHREG data = WORK.SIMDATA40_B00_ph;

    model time*event(0) = x1 x2 / ties=efron risklimits=pl;

    by group NOTSORTED;

    ods output ParameterEstimates=WORK.PLestimates40;

    hazardratio x1 / cl=pl;

    hazardratio x2 / cl=pl;

run;

DATA CENS50.selected40_b00_ph;

    set WORK.PLestimates40

        (keep = group parameter estimate stderr hrlowerplcl hrupperplcl probchisq);

    if parameter = 'x2' then delete;

RUN;


PROC DATASETS LIB=WORK NOlist kill; RUN; QUIT;

ODS HTML CLOSE;


## /* firth's correction (for nonconvergence). have to use ties=breslow with firth's */




PROC IMPORT OUT= WORK.SIMDATA40_b18_PH

            DATAFILE=

              "\\Mac\Home\Documents\SimulationData\AllSims\cens80\simdata40_b18_ph.csv"

            DBMS=CSV REPLACE;

     GETNAMES=YES;

     DATAROW=2;

RUN;
```

```
ODS HTML;

title 'EFRON WITH PROFILE LIKELIHOOD CI';

title2 'N = 40, Beta = 1.8';

PROC PHREG data = WORK.SIMDATA40_B18_ph;

    model time*event(0) = x1 x2 / firth ties=breslow risklimits=pl;

    by group NOTSORTED;

    ods output ParameterEstimates=WORK.PLestimates40;

    hazardratio x1 / cl=pl;

    hazardratio x2 / cl=pl;

run;

DATA CENS80.selected40_b18_ph;

    set WORK.PLestimates40

        (keep = group parameter estimate stderr hrlowerplcl hrupperplcl probchisq);

    if parameter = 'x2' then delete;

RUN;

PROC DATASETS LIB=WORK NOlist kill; RUN; QUIT;

ODS HTML CLOSE;


## Exact Test: call LogXact through SAS


PROC IMPORT OUT= WORK.data

            DATAFILE=

        "\\Mac\Home\Documents\SimulationData\AllSims\cens50\simdata40_b00_long.csv"

            DBMS=CSV REPLACE;

     GETNAMES=YES;

     DATAROW=2;

RUN;

PROC LOGXACT data = WORK.data seed = 902 CL=0.95;
```

```
    STRATUM timeInt x2;
    MODEL y = x1;
    ES/EX ESTIMATEFILE = WORK.estimates x1;
    by group NOTSORTED;
TITLE  'beta = 0';
RUN;
DATA CENS50.selected40_b00_strat;
    set WORK.estimates (keep = group effect Beta LowerCI UpperCI SEBeta Pvalue Type2);
RUN;
PROC DATASETS LIB=WORK NOlist MEMTYPE=data kill; RUN; QUIT;
```