Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

icenReg: Regression Models for Interval Censored Data in R

Clifford Anderson-Bergman
Sandia National Labs

Abstract

The non-parametric maximum likelihood estimator and semi-parametric regression models are fundamental estimators for interval censored data, along with standard fully-parametric regression models. The R-package **icenReg** is introduced which contains fast, reliable algorithms for fitting these models. In addition, the package contains functions for imputation of the censored response variables and diagnostics of both regression effects and baseline distribution.

Keywords: interval censoring, accelerated failure time, proportional hazards, proportional odds, survival analysis, semi-parametric regression, non-parametric.

1. Introduction

In the setting of survival analysis, interval censored data occurs when an event time is known only up to an interval. Two common forms of interval censored data are current status data (Hoel and Walburg 1972) and mixed case censoring (Schick and Yu 2000). Current status data occurs when each subject is observed at a single time and all that is recorded is whether the event of interest has occurred or not. This results in all subjects being either left or right censored. A classic current status dataset includes mice that are sacrificed at random times and inspected for lung tumors . If tumors were detected, the mice were recorded to be left censored at time of sacrifice. If no tumors were found, they were recorded as right censored. The more general type of interval censoring, called mixed case censoring , can include left censored, right censored, uncensored and observations that are censored but neither right nor left censored. The last type of censoring can occur if a subject is regularly inspected and all that is known is that the event of interest occurred between inspections. A classic mixed case interval censored data set includes semi-regular dentist visits by children, with the event of interest being emergence of permanent teeth (Vanobbergen, Lesaffre, and Declerck 2000).

By selecting the last visit without permanent teeth and the first with permanent teeth, the researchers knew the event time up to an interval. The standard assumption is that this observation time is independent of the event of interest, although the observation time may be random or fixed by design.

Standard parametric models can be used and are fairly straight forward to implement using standard algorithms. Implementations of general location-scale transformed models (the most well known being the accelerated failure time model) for interval censored data can be found in the R-package (R Core Team 2014) **survival** (Therneau 2014, Therneau and Grambsch 2000), fit with the `survreg` function. In addition, the R-package **flexsurv** (Jackson 2015, Jackson, Sharples, and Thompson 2010) can be used to fit accelerated failure time, proportional hazards and proportional odds models. These models must be used with some caution in regards to interval censored data; they are heavily influenced by the choice of parametric model, for which the model inspection can be extremely difficult.

Because of this, non-parametric models are often preferred, if at least for diagnostics. For univariate data, the non-parametric maximum likelihood estimator (NPMLE)(Turnbull 1976) is often preferred, a generalization of the Kaplan Meier curves (Kaplan and Meier 1958) for interval censored data. This is also referred to as the Turnbull estimator in the literature. This can be fit by the function `EMICM` in the package **Icens** (Gentleman and Vandal 2011, Wellner and Zhan 1997). Alternatively, this can fit by the function `computeMLE` in the package **MLEcens** (Maathuis 2013, Groeneboom, Jongbloed, and Wellner 2008). In terms of statistical properties the NPMLE is notoriously inefficient; for current status data, the convergence rate has been shown to be $n^{1/3}$ (Groeneboom and Wellner 1992) instead of the more standard $n^{1/2}$, while for mixed case interval censored data, the convergence rate has been conjectured to be $n^{1/2} - n^{1/3}$ (Groeneboom 1996, Huang 1999), depending on the severity of the censoring.

For non-parametric comparison of different strata, a log-rank test can be used (Fay 1999). This is complicated by the fact that the NPMLE is characterized by a large number of parameters, many of which may be on the boundary. Alternatively, permutation tests may be used to compare separate groups. R implementations of both these tests can be found in the R-package **interval** (Fay and Shaw 2010), called by the function `ictest`. A generalized logrank test can be found in the R-package **glrt** (X Zhao and Kim 2008, Zhao and Sun 2015).

For semi-parametric regression modeling of interval censored data, a Cox proportional hazards model (Cox 1972) can be used. Separating the estimation of the regression parameters from the estimation of the baseline parameters is not as simple as in the right censored case. One proposed method is to use the likelihood over the sum of all possible rankings in the data set (Satten 1996). An MCEM approach can also be used (Goggins, Finkelstein, Schoenfeld, and Zaslavksy 1998). Both these methods are very computationally intensive.

The model can also be kept semi-parametric by using the NPMLE as the baseline survival distribution rather than separating the baseline and regression parameters (Finkelstein 1986). Even though the rate of convergence can be as low as $n^{1/3}$ for survival estimates based on the NPMLE, it has been shown regression coefficients converge at the standard $n^{1/2}$ rate and are asymptotically normal (Huang 1995), allowing for efficient comparisons with the semi-parametric proportional hazards model. Inference on the regression parameters can be done using bootstrap standard errors (Efron 1979). A semi-parametric proportional odds model can also be used (Rossini and Tsiatis 1996, Rabinowitz, Betensky, and Tsiatis 2000). In this manuscript, these models will be referred to as the Semi-Parametric Turnbull (SPT) models.

It was shown that the SPT model can be fit with an ICM algorithm (Argaon and Eberly 1992, Pan 1999, Huang and Wellner 1997). This algorithm has been implemented in R in the package **intcox** (Henschel and Mansmann 2013). Caution must be used with this package, as it was found that the default tolerance is set too slack, often converging far from the MLE (shown later in the manuscript). In addition, it has been shown that while the ICM algorithm can preform well for purely interval censored data, it behaves very poorly in regards to required number of iterations when the data also contains a mixture of censored and uncensored response variables (Anderson-Bergman 2016b).

To the best of this author's knowledge, there is no R-package available for fitting an SPT proportional odds model for interval censored data as defined in this manuscript, other than the package being presented.

Outside the SPT model, there exist several R packages that can be used for alternative semi-parametric models. The package **coxinterval** (Boruvka and Cook 2015b) implements a Cox-Aalen model (Boruvka and Cook 2015a). The **ICsurv** (McMahan and Wang 2014) packages provides semi-parametric models that use splines for the baseline distribution (Wang, McMahan, Hudgens, and Qureshi 2016), as does the **flexsurv** package. For these methods, knot selection is still an open question. The **MIICD** (Delord 2015) provides inference through a multiple imputation approach (Delord and Génin 2015).

Outside of the R environment, there are a few options for interval censored regression models. In Stata (StataCorp 2015), the `intreg` routine fits a parametric aft model (Cameron and Trivedi 2010) that allows for interval censored data. In SAS (SAS Institute Inc 2014a), the `LIFEREG` procedure fits parametric aft models as well (SAS Institute Inc 2014b) and the `ICPHREG` fits a semi-parametric model with a spline-based estimate of the baseline distribution.

In **icenReg**, fast, reliable implementations of the fundamental tools for interval censored data are provided, intended for analysis of real data in R. This includes fitting the NPMLE through the function `ic_np` ("interval-censored non-parametric"), the SPT proportional odds and hazards model through `ic_sp` ("interval-censored semi-parametric") and the fully parametric accelerated failure time, proportional odds and proportional hazards model with a variety of choices for baseline distribution through `ic_par` ("interval-censored parametric"). The functions `diag_baseline` and `diag_covar` are provided for visual diagnostics for parametric assumptions and covariate effects which rely on the either the SPT or non-parametric models. The function `imputeCens` can be used to impute random samples of the interval-censored data, conditional on the fitted model.

In section 2, the different models fit by **icenReg** are described, along with a brief description of the algorithm used. The algorithms are tested on simulated data against competing packages, when available. In section 3, the diagnostic tools for model fitting are presented. In section 4, the imputation method used by **icenReg** is presented. In section 5, the various tools of **icenReg** are applied to a real dataset. In section 6, future plans for the package are discussed.

2. Models

Some notation and the formal definition of the generating process for interval censoring is established first. For subject i, let t_i represent the true event time of interest. The value t_i is generally not known exactly, but rather to be contained within an "observation interval", for which the left and right side are denoted by l_i and r_i. This allows for left censoring ($l_i = 0$),

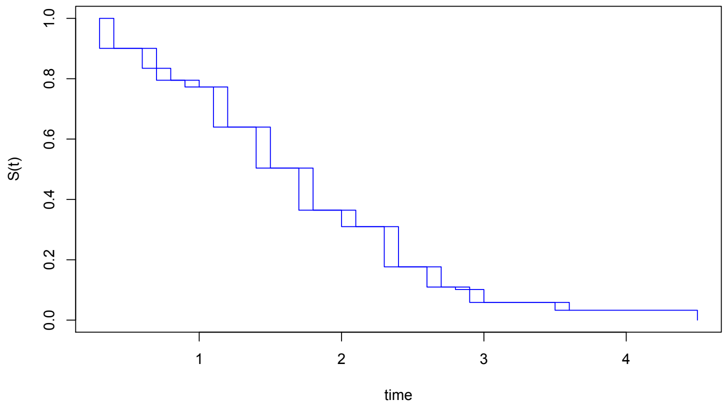

Figure 1: Sample NPMLE. Any proper survival curve between the two step functions will be an NPMLE.

right censoring ($r_i = \infty$), uncensored observations ($l_i = r_i$) and other interval censoring ($0 < l_i < r_i < \infty$). Whether these intervals are open, closed or partially open does not affect estimation for fully parametric models, but can have an effect on the non-parametric and SPT models (Ng 2002). In **icenReg**, the default behavior is to treat intervals is left-open, right-closed (*i.e.* $(l_i, r_i]$) as recommended by Ng 2002, but this can be controlled by the argument `B` in `ic_np` and `ic_sp`. Regardless of the choice of `B`, if $l_i = r_i$, the observation is treated as though it were uncensored, even though this would technically be undefined unless the interval were closed on both sides.

The generating process assumed for the censored intervals is the mixed censoring case, as defined in Schick and Yu 2000. To rephrase this definition, let K be a vector of random positive integers and let C be a random set of inspection times such that

$$C = \{C_{i,j} : i = 1, ..., n, j = 0, ..., K_i + 1 \text{ with } C_{i,0} = 0, C_{i,K_{i+1}} = \infty, C_{i,j} < C_{i,j+1}\}.$$

For each subject i, the interval such that $t_i \in (C_{i,j}, C_{i,j+1}]$ is observed and so $l_i = C_{i,j}, r_i = C_{i,j+1}$. By allowing $K_i = \infty$, all the intervals for subject i can be arbitrarily small, allowing for uncensored observations. Current status data is the special case where $K_i = 1$ for all i. An important assumption is that (K, C) are independent of t_i, *i.e.* the event time does not affect the inspection mechanism.

2.1. Non-Parametric Maximum Likelihood Estimator

For the NPMLE, the log likelihood is defined as

$$\sum_{i=1}^{n} \log(S(l_{i-}) - S(r_i))$$

such that S is a non-increasing function that maps $\mathbb{R} \to [0, 1]$. More precisely, this is the log-likelihood with closed observation intervals; the likelihood with open or partially open intervals is the same form but "clipping" the open ends of the interval for censored observations (*i.e.* replacing l_{i-} with l_i and r_i with r_{i-} if $l_i < r_i$). The NPMLE is any proper survival function S that maximizes the log likelihood function. It has been shown that the NPMLE only assigns positive probability mass to disjoint Turnbull intervals (Turnbull 1976), but how the probability mass is distributed within a Turnbull interval does not affect the likelihood. As shown on figure 1, this means the NPMLE is not necessarily unique, but rather can be defined as any function that lines between a upper and lower step function.

Computation of the NPMLE is a high dimensional constrained optimization problem, as the number of Turnbull intervals, k, can grow linearly with n. An established algorithm for fitting the NPMLE is the EMICM algorithm (Wellner and Zhan 1997). Original implementations of this algorithm calculated each iteration in $O(nk)$ time. Later algorithms considered optimization over an active set of parameters (Groeneboom *et al.* 2008, Wang 2008), where the active set is defined as the parameters with positive probability mass. This reduced computation of each iteration to $O(nk_a)$, where k_a is the size of the active set. In the case of heavily censored data, this was a massive improvement, but with lightly censored data, k_a is often close to k and in such cases these algorithms were significantly slower than the EMICM algorithm.

In **icenReg**, an efficient implementation of the EMICM algorithm is used. By taking advantage of the linear form of the data, each iteration can be calculated in $O(n)$ time, rather than $O(nk)$ (Anderson-Bergman 2016a). This provides a massive speed up compared with both the original EMICM and the active set algorithms.

The performance of `ic_np` was compared against the EMICM algorithm in **Icens** (called by `EMICM`) and the support reduction algorithm in **MLEcens** (called by `computeMLE`). Data was simulated using **icenReg**'s `simIC_weib` function as such:

```
R> simdata <- simIC_weib(n = 100, b1 = 0, b2 = 0)
```

This simulates a dataset with 100 observations. The arguments `b1` and `b2` define the regression effects; by setting them to 0, this simulates data from a Weibull(2,2) distribution. Further description of the simulation methods can be found in `?simIC_weib`. Samples size of $n = 10^2$, 10^3, 10^4, 10^5 and 10^6 were considered. Each scenario was tested 100 times. For each algorithm, the average time in seconds and average relative error were reported, where average relative error was defined as the difference in log-likelihood compared with the max likelihood achieved across all fits.

The results can be seen on table 1. All three algorithms consistently achieved a tolerable amount of error. With regards to speed, `ic_np` dominates the competing algorithms, except in the $n = 100$ case where it is virtually tied with MLEcens::computeMLE.

All simulations were run on a 2015 Macbook Air with a 2.2 GHz i7 processor.

2.2. Semi-Parametric Models

In the case of the SPT proportional hazards model, the log-likelihood can be written as

$$\sum_{i=1}^{n} \log \left(S_o(l_{i-})^{e^{X_i\beta}} - S_o(r_i)^{e^{X_i\beta}} \right),$$

	Mean Time	Mean Relative Error
n = 100		
icenReg::ic_np	0.001	6.96e-13
Icens::EMICM	0.106	6.18e-15
MLEcens::computeMLE	0.001	6.27e-10
n = 1,000		
icenReg::ic_np	0.003	3.54e-12
Icens::EMICM	8.319	4.39e-13
MLEcens::computeMLE	0.079	2.63e-9
n = 10,000		
icenReg::ic_np	0.024	0.00
MLEcens::computeMLE	12.95	7.00e-9
n = 100,000		
icenReg::ic_np	0.316	0.00
n = 1,000,000		
icenReg::ic_np	6.228	0.00

Table 1: Comparing results from icenReg, Icens and MLEcens when computing the NPMLE. Mean time is in seconds. Mean relative error is the mean difference in log-likelihood compared with the highest log-likelihood across all fits. Data was simulated using icenReg's simIC_wieb function.

where S_o is the baseline survival function, the column vector X_i contains individual's covariates without an intercept and the row vector β is a vector of coefficients. Like the definition for the NPMLE in section 2.1, this the definition of the log-likelihood with closed observation intervals. Similarly, the log likelihood function in the proportional odds model can be written as

$$\sum_{i=1}^{n} \log \left(\frac{S_o(l_{i-})e^{X_i\beta}}{S_o(l_{i-})e^{X_i\beta} - S_o(l_{i-}) + 1} - \frac{S_o(r_i)e^{X_i\beta}}{S_o(r_i)e^{X_i\beta} - S_o(r_i) + 1} \right).$$

Unfortunately, there is no established method for fitting an aft model with the NPMLE as the baseline distribution. While the necessary baseline parameters for the proportional hazards and proportional odds model can be found using the same methods as with the NPMLE (Anderson-Bergman 2016b), this cannot be generalized to the aft model, as the log-likelihood cannot be fully characterized by evaluation of S_o at a finite number of points (which can be done with the proportional hazards and odds models).

Traditional methods for computing the SPT model includes using an ICM algorithm to update the baseline parameters and conditional Newton Raphson on the regression parameters (Pan 1999, Huang and Wellner 1997). While it was found that this algorithm works well with heavily censored data, it behaves very poorly with lightly censored data (Anderson-Bergman 2016b). This is very similar to the results for the NPMLE; the ICM algorithm behaves very poorly with lightly censored data, so it is paired with EM algorithm which behaves well in that case resulting in the EMICM algorithm. In **icenReg**, a novel algorithm is used which augments the ICM algorithm with a constrained gradient ascent step, similar to augmenting the ICM with the EM algorithm in NPMLE case. Each iteration of the algorithm now includes three steps: a conditional Newton Raphson that updates the regression parameters, an ICM step that updates the baseline parameters on the log cumulative hazard scale and a constrained gradient ascent step that updates the baseline parameters on the probability mass scale. In the ICM step, a pool-adjacent violators algorithm (van Eeden 1958) is used to optimize the baseline hazard function while still respecting the monotonic constraint of the cumulative hazard. The algorithm is described in more detail in Anderson-Bergman 2016b.

To demonstrate that the algorithm finds the correct solution, results were compared with **intcox**'s `intcox` function on simulated data. The review begins with a more detailed examination of a single dataset. A simulated dataset with 500 interval censored observations from a proportional hazards model was creating using `simIC_weib` function using the default settings, other than setting the sample size to 500. This implies the simulated true event times came from a proportional hazards model with a baseline Weibull distribution with shape and scale parameters equal to 2 and regression coefficients 0.5 and -0.5.

Using **icenReg**'s `ic_sp`, the algorithm converged in 10 iterations (0.023 seconds) to the solution with final log likelihood = -347.7107 and regression parameters $\hat{\beta}_1 = 0.743, \hat{\beta}_2 = -0.550$. Using the default settings `intcox` algorithm converged in 75 iterations (0.325 seconds), but the final log likelihood was -350.7797 with regression parameters $\hat{\beta}_1 = 0.657, \hat{\beta}_2 = -0.497$. Further investigation showed this was due to slack default convergence criteria used by `intcox` (difference in log likelihood less than 10^{-4}). The `intcox` algorithm was rerun with stricter criteria (difference in log likelihood less than 10^{-10}, same as `ic_sp`), and converged in 5,026 iterations (25.27 seconds) with log likelihood -347.7107 and regression parameters $\hat{\beta}_1 = 0.743, \hat{\beta}_2 = -0.550$.

To test the reliability and speed of the algorithm, the above procedure was repeated 100 times for $n = 100, 500$, 2,000 and 10,000 (although the `intcox` algorithm was excluded from the n = 10,000 case due to speed). The computation time and relative error was recorded. Results are presented on table 2.

The results show that `ic_sp` consistently found the highest log-likelihood. Given a strict enough tolerance, it would appear that `intcox` always converged to the same solution. This provided assurance that the `ic_sp` algorithm was finding the correct solution. In addition, the new algorithm appears to be around 1,000 times faster than `intcox` given the same level of tolerance. It should be noted that these simulations are that of heavily censored data, for which the standard ICM algorithm (without the constrained gradient ascent step) does well; for lightly censored data, the `intcox` algorithm does significantly worse. For a more thorough review of performance, see Anderson-Bergman 2016b.

This author is not aware of any packages that fit an SPT proportional odds model which could be used to compare the results with. Because of this, it is instead demonstrated that the algorithm produces estimates that behave as expected. To investigate, 1,000 proportional odds datasets were simulated using `simIC_weib` for $n = 500$ and $n = 2,000$ and examined the mean and variance of the estimated regression coefficients. The true regression coefficients were $\beta_1 = 0.5, \beta_2 = -0.5$. A single dataset with $n = 500$ can be simulated according to the following code:

```
R> simdata <- simIC_weib(n = 500, model = "po")
```

For $n = 500$, $\mathbb{E}[\hat{\beta}_1] = 0.515$, $\mathbb{E}[\hat{\beta}_2] = -0.515$ and $\text{Var}[\hat{\beta}_1] = 0.031$, $\text{Var}[\hat{\beta}_2] = 0.0099$. For

	Mean Time	Mean Relative Error
n = 100		
icenReg::ic_sp	0.007	0.00
intcox default	0.21	5.41e-01
intcox strict	8.57	6.31e-06
n = 500		
icenReg::ic_sp	0.027	0.00
intcox default	0.55	3.99e+00
intcox strict	41.5	3.90e-04
n = 2,000		
icenReg::ic_sp	0.099	0.00
intcox default	2.21	2.88e+01
intcox strict	233	2.38e-02
n = 10,000		
icenReg::ic_sp	0.549	0.00

Table 2: Comparing results from icenReg and intcox on simulated data. intcox (default) refers to the function intcox::intcox with default settings, intcox (strict) refers to the same function with (epsilon = 10e-10). Mean time is in seconds. Mean relative error is the mean difference in log-likelihood compared with the highest log-likelihood across all fits. Data was simulated using icenReg's simIC_wieb function.

$n = 2,000$, $\mathbb{E}[\hat{\beta}_1] = 0.510$, $\mathbb{E}[\hat{\beta}_2] = -0.505$ and $\text{Var}[\hat{\beta}_1] = 0.0073$, $\text{Var}[\hat{\beta}_2] = 0.0023$. Thus, the estimates appear somewhat upwardly biased but consistent with variance $\propto n^{-1}$. This is consistent with results from similar simulations on the proportional hazards STP model (Pan 1999).

2.3. Fully Parametric Models

When parametric models are considered, the log likelihood function must treat uncensored observations in a distinct manner. Because of the continuous nature of the baseline distributions considered, the distinction between open and closed intervals is no longer necessary.

In the case of the proportional hazards model, the log likelihood can be written as

$$\sum_{i=1}^{n_1} \log\left(e^{X_i\beta} f_o(t_i|\alpha) S(t_i|\alpha)^{e^{X_i\beta-1}}\right) + \sum_{i=n_1+1}^{n_1+n_2} \log\left(S_o(l_i|\alpha)^{e^{X_i\beta}} - S_o(r_i|\alpha)^{e^{X_i\beta}}\right),$$

where α contains the parameters associated with the baseline distribution, the row vector β contains the regression parameters, the column X_i contains subject i's covariates, f_o and S_o are the baseline density and survival functions, with the first n_1 subjects being the uncensored subjects, and the remaining n_2 subjects are interval censored.

For the proportional odds model, the log likelihood function can be written as

$$\sum_{i=1}^{n_1} \log\left(\frac{f_o(t_i|\alpha)e^{X_i\beta}}{(S_o(t_i|\alpha)e^{X_i\beta} - S_o(t_i|\alpha) + 1)^2}\right)$$

$$+\sum_{i=n_1+1}^{n_1+n_2}\log\left(\frac{S_o(l_i|\alpha)e^{X_i\beta}}{S_o(l_i|\alpha)e^{X_i\beta}-S_o(l_i|\alpha)+1}-\frac{S_o(r_i|\alpha)e^{X_i\beta}}{S_o(r_i|\alpha)e^{X_i\beta}-S_o(r_i|\alpha)+1}\right).$$

For the accelerated failure time model, the log likelihood function can be written as

$$\sum_{i=1}^{n_1}\log\left(f_o(t_ie^{-X_i\beta}|\alpha)e^{-X_i\beta}\right)+\sum_{i=n_1+1}^{n_1+n_2}\log\left(S_o(l_ie^{-X_i\beta}|\alpha)-S_o(r_ie^{-X_i\beta}|\alpha)\right).$$

In all the parametric families considered, the parameter space α is transformed such that it is defined on $\mathbb{R}^k$, where k is the number of parameters of the parametric family. For example, for the exponential family, log(rate) is used instead of rate.

To maximize this likelihood function, a two step algorithm is used. A simple conditional Newton-Raphson step is used to update the regression parameters, as the function will be concave under standard regression conditions (non-singular design matrix). The log likelihood function is not necessarily concave as a function of the baseline parameters, and it was occasionally found to be non-locally concave for poor starting choices of α. To handle this, the algorithm would first check if the Hessian was negative definite. If so, a conditional Newton-Raphson step was used. If not, a gradient ascent step was used until the log likelihood function is locally concave.

To compare this algorithm with an established implementation, **icenReg**'s `ic_par` was compared with the results of **survival**'s `survreg` function. The default model from `survreg` is a Weibull aft model. This can be directly compared to two `ic_par` models: the Weibull aft model and the Weibull proportional hazards. This is because for the Weibull distribution, the aft and proportional hazards models are identical up to a linear transformation of variables (see appendix for derivation). As such, both models must have the same maximum likelihood. Datasets of n = 10,000 and 100,000 were simulated using `simIC_weib`, with 100 simulated datasets for each scenario. Mean time and relative error for each algorithm is presented on table 3. All three algorithms were sufficiently precise, having mean relative error on the order of 1.0e-10 to 1.0e-9. In addition, the maximum absolute difference in regression parameters across all models (after rescaling the proportional hazards model to be on the same scale as the aft parameters) was less than 6.97e-08 across all simulations. A speed advantage was held by `survreg`, being on average 6-8 times faster on the simulated data than either model fit with `ic_par`.

Even though `survreg` holds a speed advantage, `ic_par` has many model choices not available with `survreg`. As such, a proportional hazards or proportional odds model may be preferred if they fit the data better. In addition, the `ic_par` objects interact more seamlessly with the other tools provided by **icenReg**, so a user may chose to use `ic_par` with `model = "aft"`, despite the loss to speed, in order to use the other utilities provided by **icenReg**.

Currently, six parametric families are supported: exponential, gamma, weibull, log-normal, log-logistic and generalized gamma (Stacy 1962). It is worth noting that for several of these distributions, the parametric family is *not* preserved given the link function. For example, having proportional hazards to a log-normal distribution does not imply log-normality. In these cases only the baseline distribution will actually belong to the given parametric family. As such, it is very important to note that `ic_par` centers the covariates before fitting the model for numeric stability; the baseline distribution refers to subjects with mean covariate values, rather than 0. Some users may find it displeasing that the link does not preserve the

	Mean Time	Mean Relative Error
n = 10,000		
icenReg::ic_par, model = aft	0.387	9.47e-12
icenReg::ic_par, model = ph	0.336	1.04e-11
survival::survreg	0.055	2.98e-11
n = 100,000		
icenReg::ic_par, model = aft	4.84	2.10e-10
icenReg::ic_par, model = ph	4.15	2.85e-10
survival::survreg	0.598	1.60e-09

Table 3: Comparing fully-parametric results from icenReg and survival on simulated data. Mean time is in seconds.

parametric family. However, standard probability functions (*i.e.,* pdf, cdf, inverse cdf) are easily computed despite this. The package includes a function `getFitEsts` that allows easy extraction of the estimated cdf or inverse cdf from a fitted model.

The software is written in an object-oriented manner such that it requires minimal effort to add new parametric distributions; all that is needed is a C++ implementation of both the pdf and cdf function. From there, optimization is handled generically.

3. Diagnostic tools

When fitting parametric regression models, the researcher makes an assumption about the effect of the covariates and the baseline parametric model. When fitting an SPT model, the second assumption is dropped, but the covariate assumption is still required for valid inference. In either case, it is important to assess the validity of the assumptions. With interval censored data, this can be fairly difficult. The **icenReg** package includes easy to use routines for examining both sets of assumptions using the SPT model. Unfortunately, these methods only apply to proportional odds and hazards models, as there are no methods to fit the SPT aft model.

To examine the parametric baseline assumption, `diag_baseline` fits and plots the baseline survival distribution of a variety of parametric choices. It also plots the SPT estimated baseline distribution. This can help an investigator assess if there appears to be a systematic deviation from the assumed baseline distribution.

To examine the functional form of the covariates, `diag_covar` uses the fact that for both models, there is a transformation of the survival function such that differences in covariate effects will result in constant differences. For the proportional hazards model, note that

$$\log(-\log(S(t|X,\beta))) = \log(-\log(S_o(t)^{e^{X\beta}})) = X\beta + \log(-\log(S_o(t)))$$

Likewise, for the proportional hazards model, note that

$$\log\left(\frac{S(t|X,\beta)}{1-S(t|X,\beta)}\right) = \log\left(\frac{\left(\frac{S_o(t)e^{X\beta}}{S_o(t)e^{X\beta}-S_o(t)+1}\right)}{\left(\frac{S_o(t)e^{X\beta}}{S_o(t)e^{X\beta}-S_o(t)+1}\right)-1}\right) =$$

$$\log\left(\frac{\left(\frac{S_o(t)e^{X\beta}}{S_o(t)e^{X\beta}-S_o(t)+1}\right)}{\left(\frac{S_o(t)-1}{S_o(t)e^{X\beta}-S_o(t)+1}\right)}\right) = X\beta + \log\left(\frac{S_o(t)}{S_o(t)-1}\right)$$

To investigate whether the functional form is appropriate for a given covariate, `diag_covar` first stratifies the dataset on different levels of that covariate. It then fits an SPT model for each strata and plots the given transformation of the baseline survival functions. If the functional form of the covariate is correct, the difference between the two strata's transformed baseline should be approximately constant. To help visualize the difference, the average of all the strata is subtracted off of each strata by default. By subtracting off the average of all the curves, each curve should be a flat line (with stochastic noise) under the correct regression model. When the regression model is incorrect, often the different curves will converge together or cross. Users also have the option to examine the transformed survival curves without the mean subtracted (may be necessary if there is little overlap between strata) and the raw baseline survival function estimates.

Because there is no established method for calculating the aft SPT model, neither of these methods can be applied to the aft model. The author is currently working on alternative diagnostic tools for the aft model.

4. Imputation

In some cases, the analyst may wish to impute the missing data (*i.e.* exact event time). For example, this could be a step in a multiple imputation analysis (Rubin 1987). This functionality is provided in `imputeCens`.

Three imputation strategies are allowed by `imputeCens`. The simplest is median imputation (`imputeType = "median"`), in which event times are imputed with the median value, conditional on being inside the given observational intervals and the parameter values at the MLE. The next strategy (`imputeType = "fixedParSample"`) takes a random draw of the event time, conditional on being within the given observation interval and the parameter values at the MLE. Finally, the last strategy (`imputeType = "fullSample"`) takes a random sample of the parameters. Then, conditional on those parameters values, it takes a random sample of the event times, conditional on being within the observational interval and the randomly drawn parameter values. How the parameters are sampled are dependent on the model. For the fully parametric model, the asymptotic normality of the estimators motivate taking a random draw from a multivariate normal with mean and covariance provided by the point estimates and negative inverse Hessian matrix. However, this method cannot readily be applied to the NPMLE and SPT models, as the baseline parameters do not follow an asymptotically normal distribution. As such, for the SPT model the `"fullSample"` option fixes the baseline parameters and takes a random sample of the bootstrapped regression coefficients before sampling the conditional event times. For the NPMLE, `"fullSample"` is equivalent to `"fixedParSample"`.

In addition, the analyst must keep in mind that the NPMLE and SPT model only apply probability mass onto Turnbull intervals. Because of this, on most fits there will be several "gaps" $[a_j, b_j)$[1] on $[0, \infty)$ for which the SPT model estimates $P(T \in (a_j, b_j)|X) = 0$ for all

[1] Whether this interval is open, closed or partially open will depend the choice of `B` for the initial data. In

Function Name	Basic Description
Modeling Functions	
`ic_np`	Non-parametric
`ic_sp`	Semi-parametric
`ic_par`	Fully-parametric
Graphical Diagnostic Tools	
`diag_baseline`	Compares parametric baselines
`diag_covar`	Compares regression effects
Utilities	
`imputeCens`	imputes censored data
`getFitEsts`	gets p, q values from fit
`getSCurves`	gets `ic_np`, `ic_sp` baseline curve
`simIC_weib`	simulates censored data
Standard Methods	
`plot, lines, summary, predict, vcov`	

Table 4: Functions provided in **icenReg**.

values of the covariates X. In fact, consider $r_o = \min\{r_i : r_i \geq l_j,\ j \in 1, ..., n\}$. There will be 0 probability mass assigned to (r_o, ∞), as by definition r_o will be the right side of the maximum Turnbull interval. Likewise, if $l_o = \max\{l_i : l_i \leq l_j,\ j \in 1, ..., n\}$, then zero probability mass is assigned to $[0, l_o)$. These gaps can create several complications; first, they can significantly shrink the range of the imputed values, especially if r_o is relatively small (this can happen if a large percentage of the data is right censored). Secondly, it can create a problem if a user attempts to impute a response for which the observation interval is completely contained within one of the gaps. Note that this *cannot* happen with data that was used to fit the SPT model, as each observations must contain at least one Turnbull interval with positive probability mass to have a finite log-likelihood, but it could happen if a user attempted to impute data for an observation interval that was not used to fit the model.

Given these complications, it is advised that the analyst use the most appropriate parametric model for imputation unless doing so leads to clear bias in imputation.

5. Using icenReg

Table 4 provides a very quick summary of the public functions provided in **icenReg**. To help illustrate the use of the package, a sample analysis is presented.

Thee `IR_diabetes` dataset (Borch-Johnsens, K, Andersen, P and Decker, T 1985) from **icenReg** is used, which was initially imported from **glrt** (Zhao and Sun 2015). In this dataset, 731 patients (454 males and 277 females) are followed, with time from onset of diabetes to onset of diabetic nerphronpathy being the response time of interest. For many of the patients (595), the event time was known exactly but for others (136) the exact time was known only up to an interval due to limited follow up. The data set contains three variables: `left, right` and

this case, it is assumed that the default choice of `B = c(0,1)` is used, as discussed in the beginning of section 2.

gender. The variables left and right represent the observational interval. In this example, the effect of gender will be examined.

```
R> data("IR_diabetes")
R> head(IR_diabetes)
  left right gender
1   24    27   male
2   22    22 female
3   37    39   male
4   20    20   male
5    1    16   male
6    8    20 female
```

First, the NPMLE is fit to each group. The syntax for this is very similar to fitting the Kaplan Meier curves with survival::survfit, but the response must either be a Surv object with type = "interval2" or of the form cbind(l, r), where l,r are the left and right side of the observation interval for each subject. This syntax is also used for ic_sp and ic_par.

```
R> npmleFit <- ic_np(cbind(left, right) ~ gender, data = IR_diabetes)
```

Plots of the NPMLE for each group can be created using the plot function. This can be see on figure 2.

```
R> plot(npmleFit, main = "NPMLE by gender", col = c("blue", "orange"))
```

While the two NPMLE fits give a full picture of comparing the two groups, an investigator may want to use a regression model to more succinctly describe the difference between the two groups. One can begin with visually assessing which regression model appears more appropriate.

```
R> diag_covar(cbind(left, right) ~ gender,
+            data = IR_diabetes, model = "ph"
+            col = c("blue", "orange") )
R> diag_covar(cbind(left, right) ~ gender,
+            data = IR_diabetes, model = "po")
+            col = c("blue", "orange") )
```

Examining figure 3, some deviation from the regression assumptions can be seen; the transformed difference between the groups is greater early on but becomes less as time goes on. However, for the proportional odds model, the deviation is less and acceptable for inference. Because of this, it was chosen to model the data with a proportional odds effect of gender.

Given that bootstrap is required for estimating the standard errors, it is simple to use multiple cores to speed up computation. For parallel computing of bootstrap samples, **icenReg** works seamlessly with R's **doParallel** package (Revolution Analytics and Steve Weston 2014), although it is left to the user to set up the cluster (as to not interfere with other processes that may be running). This is demonstrated below. The fitting of the model and an additional 1,000 bootstrap samples took just under 30 seconds utilizing 2 cores.

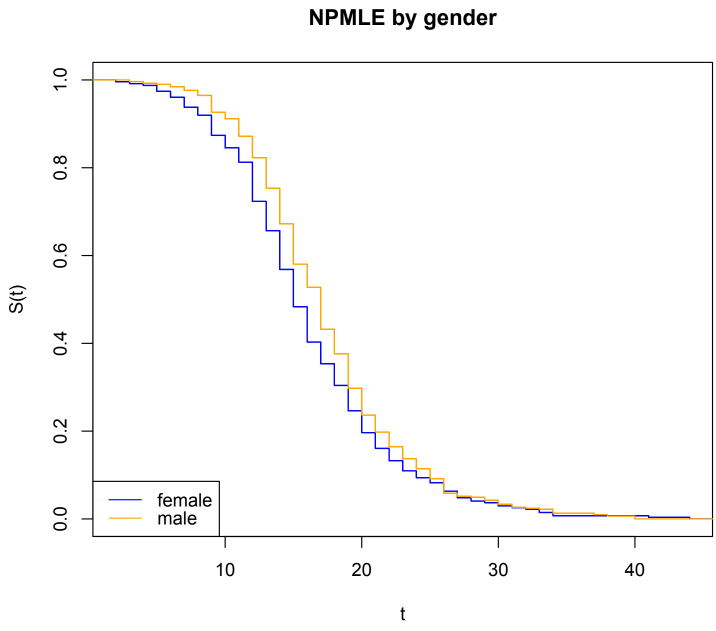

Figure 2: NPMLE for each group.

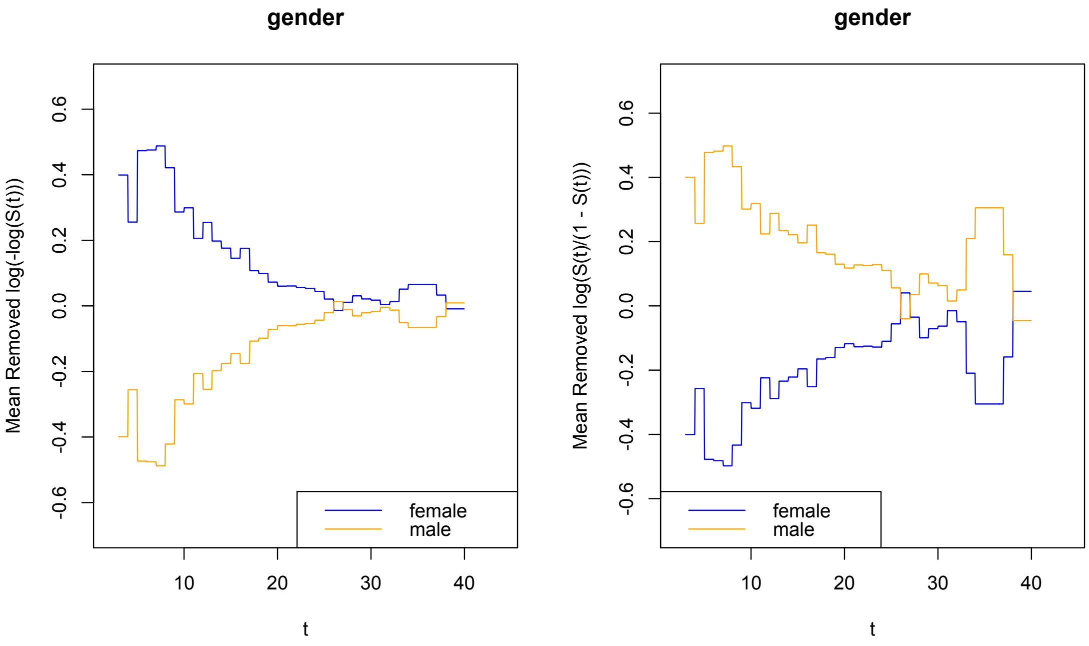

Figure 3: Checking the appropriateness of regression effects. If regression effect is correct, lines should be approximately parallel. Left plot checks the proportional hazards, right plot checks proportional odds.

```
R> library("doParallel")
R> myClust <- makeCluster(2)
R> registerDoParallel(myClust)
R>
R> sp_fit <- ic_sp(cbind(left, right) ~ gender,
+                              model = "po", data = IR_diabetes,
+                              bs_samples = 1000, useMCores = T)
R> stopCluster(myClust)
```

The summary function can be used to review the results.

```
R> summary(sp_fit)

Model:  Proportional Odds
Baseline:  semi-parametric
Call: ic_sp(formula = cbind(left, right) ~ gender, data = IR_diabetes,
    model = "po", bs_samples = 1000, useMCores = T)

           Estimate Exp(Est) Std.Error z-value        p
gendermale   0.4001    1.492    0.1491   2.683 0.007298

final llk =  -1959.716
Iterations =  23
Bootstrap Samples =  1000
```

From the summary, it can be seen that there is a statistically significant difference in the odds of not having experienced diabetic nerphronpathy at a given time after diabetes between men and women in the study. It is estimated that the odds of survival at any given time will be 1.49 times higher for men than for women (95% CI = 1.11, 2.00, p = 0.0073) under the assumption of proportional odds.

Using the plot method, the user can plot estimated survival curves between the two groups. This is done by including a new data set with the covariates for each survival curve to be plotted. If no newdata argument is included, the baseline group is plotted. The plotted curves can be found on figure 4.

```
R> newdata <- data.frame(gender = c("female", "male"))
R> rownames(newdata) <- c("Female", "Male")
R> plot(sp_fit, newdata,
+      main = "Semi-parametric Fits by Gender",
+      col = c("blue", "orange") )
```

In some cases, a parametric model may be preferred. For example, even though bootstrapping can be used for inference on the regression parameters, it cannot be used for inference on the baseline survival distribution (and thus the conditional survival probabilities as well). If a parametric model is chosen, it is important to check that the chosen baseline distribution is appropriate. Using diag_baseline with a proportional odds model, the user can view how

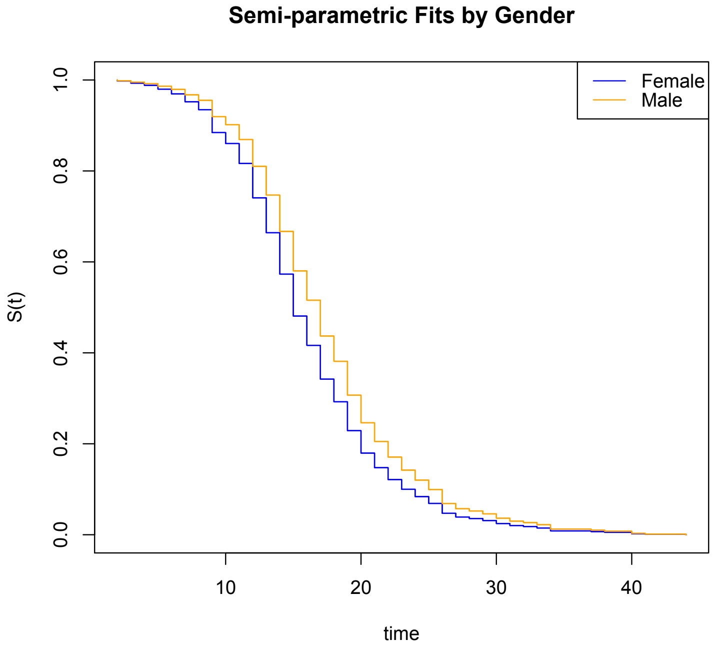

Figure 4: Plotted survival curves based on the SPT proportional odds model.

the different parametric models compare with the SPT fit. If the argument `dist` is left blank, default behavior is plot all available parametric distributions against the SPT model. This is plotted on figure 5. It was decided that the log-logistic distribution was the most appropriate, given that there appears no systematic deviation from the SPT fit.

```
R> diag_baseline(sp_fit, lgdLocation = "topright")
R> diag_baseline(sp_fit, dist = "loglogistic", lgdLocation = "topright")
```

The proportional odds model with a log-logistic baseline distribution can be fit using `ic_par`.

```
R> par_fit <- ic_par(cbind(left, right) ~ gender,
+                    data = IR_diabetes,
+                    model = "po", dist = "loglogistic")
R> summary(par_fit)

Model:  Proportional Odds
Baseline:  loglogistic
Call: ic_par(formula = cbind(left, right) ~ gender, data = IR_diabetes,
    model = "po", dist = "loglogistic")

           Estimate Exp(Est) Std.Error z-value        p
log_alpha    2.7730   16.000   0.01372 202.100 0.000000
```

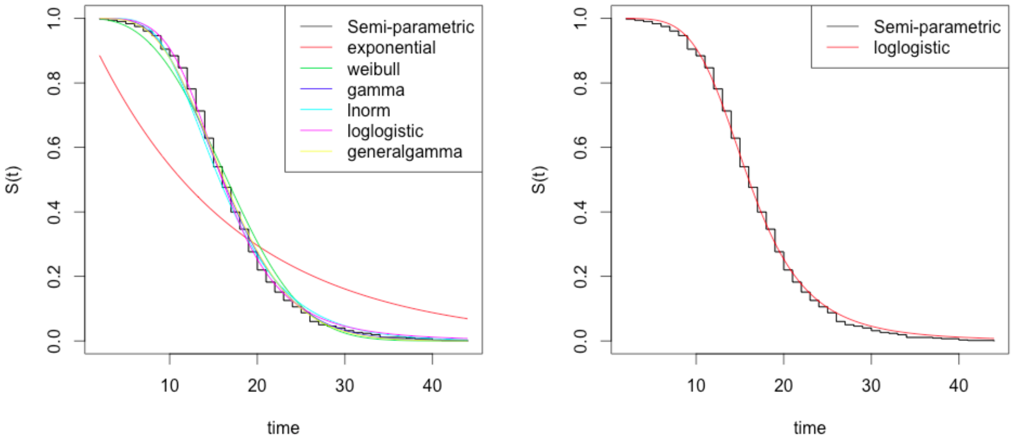

Figure 5: Examining parametric baseline fits.

```
log_beta      1.5820     4.865    0.03299  47.960 0.000000
gendermale    0.3852     1.470    0.13840   2.783 0.005385

final llk =  -2003.695
Iterations =  4
```

The results from the parametric fit closely agree with the SPT model; a statistically significant difference was found in the proportional odds of survival for males compared with females. The odds of survival were estimated to be 1.47 times higher for males than females at all times (95% CI = 1.12, 1.93, p = 0.0053).

The function `getFitEsts` can be used to get estimated event time or probabilities from a fitted model. For example, a user could use the following code the extract the median event time for males and the survival probability at `t = 10` for females.

```
R> maleCovs <- data.frame(gender = c("male"))
R> femaleCovs <- data.frame(gender = c("female"))
R> getFitEsts(par_fit, newdata = maleCovs, p = 0.5)
[1] 16.48959
R> getFitEsts(par_fit, newdata = femaleCovs, q = 10)
[1] 0.114272
```

Finally, if the user wanted to impute the censored data to be passed to another model, this could be done quite easily with `imputeCens`. This returns a matrix of imputed values, where rows correspond with subjects and columns correspond different imputations. Note that for subjects that are uncensored, the imputed values will be constant.

```
R> imputeValues <- imputeCens(par_fit, imputeType = "fullSample")
R> head(imputeValues)
         [,1]      [,2]      [,3]      [,4]      [,5]
```

```
[1,] 26.67204 24.71427 25.05204 24.74152 26.07612
[2,] 22.00000 22.00000 22.00000 22.00000 22.00000
[3,] 37.43854 37.90400 37.77349 38.81463 37.07689
[4,] 20.00000 20.00000 20.00000 20.00000 20.00000
[5,] 13.94358 13.95117 12.87287 15.12503 12.06405
[6,] 13.35249 11.88034 15.39899 18.74752 13.78116
```

6. Discussion

The author's vision for the **icenReg** package is to provide analysts with a reliable, organized set of tools for the analysis of interval censored data. As such, implementing established methods over novel methods is preferred, although there is interested in fast new algorithms, as implemented for the non-parametric and SPT model.

With this general guideline, there are several improvements to the package planned in the near future. At this time, there is work on developing diagnostic tools for the aft model. In addition, there are plans for including residuals, such as those found in Farrington 2000. Time varying coefficients, as found in Sparling, Younes, and Lachin 2006 would be another useful addition.

7. Acknowledgements

All matrix algebra done at the C++ level in **icenReg** is handled by the **Eigen** library (Guennebaud, Jacob *et al.* 2010). This library was accessed by linking to the **RcppEigen** package (Bates and Eddelbuettel 2013). The implementation for the generalized gamma distribution functions was largely guided by code found in the **flexsurv** (Jackson 2015) package. **icenReg**'s `ICNPMLE` function calls **MLEcens**'s (Maathuis 2013) `reduc` function, the HeightMap (Maathuis 2005) algorithm for finding the maximal intersections for the bivariate NPMLE. At this time, `ICNPMLE` has not been made publicly available.

The author would like to thank Ledif Grisell Diaz-Rameriz for review of the manuscript and insightful discussions on methods for applied analyses, Dirk Eddelbuettel for kindly pointing out that **Eigen** did not need to be statically included in **icenReg** and Professor Perry de Valpine and the NIMBLE team for invaluable lessons in object oriented programming. Finally, the author would like to thank the editor and anonymous reviewers for very constructive feedback.

References

Anderson-Bergman C (2016a). "An efficient implementation of the EMICM algorithm for the interval censored NPMLE." *Journal of Computational and Graphical Statistics*. (just accepted).

Anderson-Bergman C (2016b). "Revisiting the iterative convex minorant algorithm for interval censored survival regression models." *Preprint*.

Argaon J, Eberly D (1992). "On Convergence of Convex Minorant Algorithms for Distribution Estimation with Interval-Censored Data." *Journal of Computational and Graphical Statistics*, **1**, 129–140.

Bates D, Eddelbuettel D (2013). "Fast and Elegant Numerical Linear Algebra Using the RcppEigen Package." *Journal of Statistical Software*, **52**(5), 1–24. URL http://www.jstatsoft.org/v52/i05/.

Borch-Johnsens, K, Andersen, P and Decker, T (1985). "The effect of proteinuria on relative mortality in Type I (insulin-dependent) diabetes mellitus." *Diabetologia*, **28**, 590–596.

Boruvka A, Cook R (2015a). "A Cox-Aalen model for interval-censored data." *Scandinavian Journal of Statistics*, **42**, 414–426.

Boruvka A, Cook RJ (2015b). *coxinterval: Cox-Type Models for Interval-Censored Data*. R package version 1.2, URL https://CRAN.R-project.org/package=coxinterval.

Cameron A, Trivedi P (2010). "Microeconometrics Using Stata." *Stata Journal*, pp. 395–404.

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

Delord M (2015). *MIICD: Multiple Imputation for Interval Censored Data*. R package version 2.2, URL https://CRAN.R-project.org/package=MIICD.

Delord M, Génin E (2015). "Multiple imputation for competing risks regression with interval-censored data." *Journal of Statistical Computation and Simulation*, pp. 1–12.

Efron B (1979). "Bootstrap Methods: Another Look at the Jackknife." *The Annals of Statistics*, **7**, 1–26.

Farrington CP (2000). "Residuals for Proportional Hazards Models with Interval-Censored Survival Data." *Biometics*, **56**, 473–482.

Fay MP (1999). "Comparing Several Score Tests for Interval Censored Data." *Statistics in Medicine*, p. 2681.

Fay MP, Shaw PA (2010). "Exact and Asymptotic Weighted Logrank Tests for Interval Censored Data: The interval R Package." *Journal of Statistical Software*, **36**(2), 1–34.

Finkelstein DM (1986). "A Proportional Hazards Model for Interval-Censored Failure Time Data." *Biometrika*, **42**, 845–854.

Gentleman R, Vandal A (2011). *Icens: NPMLE for Censored and Truncated Data*. R package version 1.38.0.

Goggins W, Finkelstein D, Schoenfeld D, Zaslavksy A (1998). "A Markov chain Monte Carlo EM algorithm for analyzing interval-censored data under the Cox proportional hazards model." *Biometrics*, **54**, 1498–1507.

Groeneboom P (1996). *Lectures on Inverse Problems*. Lecture Notes in Mathematics, 1648, Springer-Verlag, Berlin.

Groeneboom P, Jongbloed G, Wellner J (2008). "The Support Reduction Algorithm for Computing Non-Parametric Function Estimates in Mixture Models." *Scandinavian Journal of Statitistics*, pp. 385–399.

Groeneboom P, Wellner J (1992). *Information Bounds and Non-Parametric Maximum Likelihood Estimation.* DMV Seminar, Band 19,Birkhauser, New York.

Guennebaud G, Jacob B, *et al.* (2010). "Eigen v3." http://eigen.tuxfamily.org.

Henschel V, Mansmann U (2013). *intcox: Iterated Convex Minorant Algorithm for Interval Censored Event Data.* R package version 0.9.3, URL http://CRAN.R-project.org/package=intcox.

Hoel DG, Walburg HE (1972). "Statistical Analysis of Survival Experiments." *The Annals of Statistics*, **18**, 1259– 1294.

Huang J (1995). "Efficient Estimation for the Proportional Hazards Model with Interval Censoring." *The Annals of Statistics*, **24**, 540–568.

Huang J (1999). "Asymptotic Properties of Nonparametric Estimation Based on Partly Interval-Censored Data." *Statistica Sinica*, **9**, 501, 519.

Huang J, Wellner J (1997). "Interval Censored Survival Data: A Review of Recent Progress." *Proceedings of the First Seattle Symposium in Biostatistics: Survival Analysis.*

Jackson C (2015). *flexsurv: Flexible Parametric Survival and Multi-State Models.* R package version 0.7, URL https://CRAN.R-project.org/package=flexsurv.

Jackson C, Sharples L, Thompson S (2010). "Survival models in health economic evaluations: balancing fit and parsimony to improve prediction." *International Journal of Biostatistics*, **6(1): Article 34**.

Kaplan E, Meier P (1958). "Nonparametric Estimation from Incomplete Observations." *Communication in Statistics - Theory and Methods*, **27**, 1961 – 1977.

Maathuis M (2005). "Reduction algorithm for the NPMLE for the distribution function of bivariate interval censored data." *Journal of Computational and Graphical Statistics*, **14**, 252–262.

Maathuis M (2013). *MLEcens: Computation of the MLE for Bivariate (Interval) Censored Data.* R package version 0.1-4, URL http://CRAN.R-project.org/package=MLEcens.

McMahan CS, Wang L (2014). *ICsurv: A package for semiparametric regression analysis of interval-censored data.* R package version 1.0, URL https://CRAN.R-project.org/package=ICsurv.

Ng M (2002). "A Modification of Peto's Nonparametric Estimation of Survival Curves for Interval-Censored Data." *Biometrics*, **58**, 439–442.

Pan W (1999). "Extending the Iterative Convex Minorant Algorithm to the Cox Model for Interval-Censored Data." *Journal of Computational and Graphical Statistics*, **8**, 109–120.

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

Rabinowitz D, Betensky R, Tsiatis A (2000). "Using Conditional Logistic Regression to Fit Proportional Odds Models to Interval Censored Data." *Biometrics*, **56**, 511–518.

Revolution Analytics and Steve Weston (2014). *doParallel: Foreach parallel Adaptor for the Parallel Package*. R package version 1.0.8, URL `http://CRAN.R-project.org/package=doParallel`.

Rossini A, Tsiatis A (1996). "A Semiparametric Proportional Odds Regression Model for the Analysis of Current Status Data." *Journal of the American Statistical Assocation*, **91**, 713–721.

Rubin D (1987). *Multiple imputation for nonresponse in surverys*. John Wiley, New York.

SAS Institute Inc (2014a). *SAS/STAT Software, Version 13.2*. Cary, NC. URL `http://www.sas.com/`.

SAS Institute Inc (2014b). *SAS/STAT©13.2 User's Guide*. SAS Institute Inc., Cary, NC.

Satten G (1996). "Rank-based inference in the proportional hazards model for interval censored data." *Biometrika*, **82**, 355–370.

Schick A, Yu Q (2000). "Consistency of the GMLE with Mixed Case Interval-Censored data." *Scandinavian Journal of Statistics*, pp. 45–55.

Sparling Y, Younes N, Lachin J (2006). "Parametric Survival Models for Interval-Censored Data with Time-Dependent Covariates." *Biostatistics*, **7**, 599–614.

Stacy E (1962). "A generalization of the gamma distribution." *Annals of Mathematical Statistics*, **33**, 1187–1192.

StataCorp (2015). *Stata Statistical Software: Release 14*. College Station, TX. URL `http://www.stata.com/`.

Therneau T (2014). *A Package for Survival Analysis in S*. URL `http://CRAN.R-project.org/package=survival`.

Therneau TM, Grambsch PM (2000). *Modeling Survival Data: Extending the Cox Model*. Springer-Verlag, New York. ISBN 0-387-98784-3.

Turnbull B (1976). "The Empirical Distribution with Arbitrarily Grouped and Censored Data." *Journal of the Royal Statistical Society B*, **38**, 290–295.

van Eeden C (1958). *Testing and Estimating Ordered Parameters of Probability Distributions*. PhD. thesis, University of Amsterdam.

Vanobbergen J, Lesaffre E, Declerck D (2000). "The Signal-Tandmobiel ® Project - a Longitudinal Intervention Health Promotion Study in Flanders (Belgium): Base and First Year Results." *European Journal of Pediatric Dentistry*, **2**, 87–96.

Wang L, McMahan C, Hudgens M, Qureshi Z (2016). "A flexible, computationally efficient method for fitting the proportional hazards model to interval-censored data." *Biometrics*, **72**, 222–231.

Wang Y (2008). "Dimension-Reduced Nonparametric Maximum Likelihood Computation for Interval-Censored Data." *Computational Statistics & Data Analysis*, **52**(5), 2388–2402.

Wellner JA, Zhan Y (1997). "A Hybrid Algorithm for Computation of the Nonparametric Maximum Likelihood Estimator from Censored Data." *Journal of the American Statistical Association*, **92**, 945–959.

X Zhao Q Zhao JS, Kim J (2008). "Generalized Log-rank Tests for Partly Interval-Censored Failure Time Data." *Biometrical Journal*, **50**, 375–385.

Zhao Q, Sun J (2015). *glrt: Generalized Logrank Tests for Interval-censored Failure Time Data*. R package version 2.0, URL https://CRAN.R-project.org/package=glrt.

8. Appendix

In section 2.2, it was mentioned that the Weibull proportional hazards is equivalent to the Weibull aft model, up to a change of variables. To demonstrate this, first consider the Weibull aft survival function:

$$S(t|X,\beta) = S_o(te^{-X^T\beta}) = e^{-(te^{-X^T\beta}/\gamma)^\lambda}.$$

The Weibull proportional hazards survival function can be written as

$$S(t|X,\beta) = S_o(t)^{e^{X^T\beta}} = (e^{-(t/\gamma)^\lambda})^{e^{X^T\beta}} = e^{-(te^{X^T\beta\lambda}/\gamma)^\lambda}.$$

Note that for any Weibull aft survival function, an equivalent Weibull proportional hazard function exists by transforming $\beta_{ph} = -\lambda\beta_{aft}$, where β_{ph} are the regression parameters used by the proportional hazard model and β_{aft} is the regression parameters used by the aft model.

Affiliation:

Clifford Anderson-Bergman
Sandia National Labs
7011 East Avenue
Livermore, CA
E-mail: ciande@sandia.gov

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd