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
Gladstone Institutes, UCSF

Abstract

Algorithms for fitting a semi-parametric Cox-PH model for interval censored data have existed in the literature for over 15 years at the time of this writing. However, currently there are no reliable R packages for fitting such models for the analysis of real data. The established algorithms are by no means trivial, so it is unreasonable to think that a researcher would rewrite these algorithms to analyze their own data set. To fill this gap, we created the R-package **icenReg** whose centerpiece is an algorithm for computing the semi-parametric Cox-PH or proportional odds model based on a combination of established algorithms for the Cox-PH model and similar problems. Standard errors are estimated via bootstrapping, which is set up to take advantage of multiple cores with minimal effort by the user. In addition, we provide functions for fitting Cox-PH and proportional odds parametric models for interval censored data and functions for easy diagnosis of both parametric assumptions and covariate effects.

Keywords: interval censoring, Cox-PH, proportional odds, survival analysis, semi-parametric regression, multiple imputations.

1. Introduction

In the setting of survival analysis, interval censored data occurs when an event time is known only up to an interval. There are two common forms of interval censored data: case I, or current status data, occurs when each subject is observed at a single (random or fixed by design) 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 censored. A classic dataset includes mice that are sacrificed at random times and inspected for lung tumors (Hoel and Walburg (1972)). 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. Case II, or general interval 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 case II 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.

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) can be found in the R-package (R Core Team (2014)) **survival** (Therneau (2014), Therneau and Grambsch (2000)), fit with the `survreg` function. These models must be used with some caution: they are heavily influenced by the choice of parametric model, for which the model inspection can be extremely difficult due to the interval censoring.

Because of this, non-parametric models are often used, at least for diagnosistics. For univariate data, the non-parametric maximum likelihood estimator (NPMLE) is often preferred (Turnbull (1976)), which can be thought of as a generalization of the Kaplan Meier curves (Kaplan and Meier (1958)) for interval censored data. 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)) which, in the current implementations, is much faster. In terms of statistical properties NPMLE is notoriously inefficient; for case I interval censored 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 case II the convergence rate has been conjectured to be $n^{1/2} - n^{1/3}$ (Groeneboom (1996), Huang (1999)), depending on the severity of the interval censoring.

For non-parametric comparison of different stratum, a log-rank test can be used (Fay (1999)). However, this is complicated by the fact that the NPMLE is characterized by a large number of parameters, many of which many 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`.

For semi-parametric regression modeling of interval censored data, a Cox-PH model (Cox (1972)) can be used (Finkelstein (1986)). Unlike the case of right censored data, the regression parameters cannot be estimated without estimating baseline survival distribution. The model can still be kept semi-parametric by using the NPMLE as the baseline survival distribution. Interestingly, 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 Cox PH model. Inference on the regression parameters can be done using bootstrap standard errors (Efron (1979)). Alternatively, a semi-parametric proportional odds model can be used (Rossini and Tsiatis (1996), Rabinowitz, Betensky, and Tsiatis (2000)).

It was shown that the model can be fit with an ICM algorithm (Argaon and Eberly (1992), Pan (1999a)). This algorithm has been implemented in R in the package **intcox** (Henschel and Mansmann (2013)). However, there are several shortcomings in this package which make it inadequate for general data analysis: first and foremost, the algorithm does appear to be able handle uncensored observations in the data, implying it would not be able to handle general case II interval censoring. We are not sure what causes this problem. Similarly, the algorithm would often immediately fail in the case of discrete censoring, even if all the observation intervals have strictly positive interval lengths. Furthermore, no standard errors are provided for the estimates (although this can be overcome via bootstrapping). Finally, under the standard settings the algorithm prematurely converges quite often, leading to regression estimates that can be off by a significant amount. For example, in a recent general tutorial on interval censored data (Gomez, Calle, Oller, and Langohr (2009)), the authors use `intcox` on a sample dataset, which results in final log likelihood of -5368.38 and regression parameters 0.288 and 0.316. However, rerunning this analysis, we see that `intcox` reports an error after the first iteration. We found that at the MLE, the log likelihood was in fact -5138.57, with regression estimates of 0.322 and 0.337, using the software we provide. Given these issues, we would not consider **intcox** reliable for general analysis of real data in it's current form.

To the best of our knowledge, there is no R-package available for fitting a semi-parametric proportional odds model for interval censored data.

In **icenReg**, we provide tools intended for analysis of real data. In particular, we present a more reliable and considerably faster algorithm for fitting the semi-parametric Cox-PH and proportional odds models through the function `ic_sp`. The algorithm iterates between a conditional active set step to update the baseline parameters and a conditional Newton Rhapson algorithm to update the regression parameters. This more advanced algorithm allows for much faster and more reliable computation, allowing for quick bootstrapping for moderately sized ($n \leq 10,000$) datasets. We also provide routines for fully parametric Cox-PH and proportional odds regression models via `ic_par`. Finally, we provide functions `diag_baseline` and `diag_covar` for easy visual review of both parametric assumptions and covariate effects which relies on our semi-parametric models.

In section 2, we describe our algorithm to fit the semi-parametric model and compare results to those found in **intcox**.

2. Description of algorithm for semi-parametric models

For characterization of the semi-parametric Cox-PH and proportional odds models, we start from the assumption of independence of the event time and censoring mechanism. We also use the standard interval notation that for subject i, l_i and r_i are the left and right side of their "observation interval" (*i.e.* the interval for which the event is known to have occurred). This allows for left censoring ($l_i = 0$), right censoring ($r_i = \infty$), uncensored observations ($l_i = r_i$) and general interval censoring. From here, we construct the Cox-PH likelihood function as

$$\sum_{i=1}^{n} \log \left(S_o(l_i)^{e^{X_i \beta}} - S_o(r_{i+})^{e^{X_i \beta}} \right)$$

where S_o is the baseline survival function, X_i are an individual's covariates with no intercept and β is a vector of coefficients. Similarly, we can define the likelihood function in the proportional odds model as

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

It is worth noting that while for continuous distributions, censored and uncensored observations must be handled differently (*i.e.* with the survival distribution and probability density function respectively), in the case of the semi-parametric estimator, it has been shown that the baseline distribution can treated as a discrete distribution and so $S_o(t) - S_o(t_+) > 0$ if probability mass is placed at t. In fact, it has been shown that the MLE places positive probability only inside *Turnbull intervals* (Turnbull (1976)), *i.e.* intervals such that the left side of the interval is the left side of an observation interval, the right side of the interval is the right side of (possibly a different) observation interval, with no ends of any other intervals in between. Uncensored observations are considered to be intervals of length 0, and so automatically become Turnbull intervals. With this in mind, the above likelihood functions can be written as a function of a finite number of parameters by replacing $S_o(x_i)$ with

$$1 - \sum_{T_j < x_i} p_j$$

$$\text{such that } p_j \geq 0, \sum p_j = 1$$

where p_j is the probability mass assigned to Turnbull interval T_j. This is a slight abuse of notation, as comparing an interval (T_j) to a point (l_i or r_i) is typically ill defined. In our case, by definition of the Turnbull intervals, if l_i or r_i (*i.e.* the ends of an observation interval) are greater than or equal to any single point in a Turnbull interval, they are greater than or equal to all points in the Turnbull interval. Likewise for less than or equal to. Under this parameterization, we can characterize the MLE with a finite number of parameters; the regression parameters β and baseline distribution probability masses p.

To add in computation, we optimize the likelihood on the cumulative hazard scale rather probability masses. Reparameterizing the discrete survival function as a cumulative hazard function, we can describe the baseline cumulative hazard as

$$\Lambda(x_i, H) = \begin{cases} -\infty & \text{if } x_i < T_1 \\ H_j & \text{if } x_i \geq T_j \cap x_i < T_{j+1} \\ \infty & \text{if } x_i > T_k \end{cases}$$

$$\text{such that } H_j \leq H_{j+1}$$

where k is the number of Turnbull intervals.

We also define the function which links the baseline cumulative hazard to the survival function conditional on the covariates as $f_m(h|X, \beta)$, where m represents the model selected. For the Cox-PH model, we have

$$f_{ph}(h|X, \beta) = e^{-e^{h+X\beta}}$$

For the proportional odds model, we have

$$f_{po}(h|X, \beta) = \frac{e^{-e^h + X\beta}}{e^{-e^h + X\beta} - e^{-e^h} + 1}$$

Using this, our algorithm maximizes the following function:

Proportional Hazards		
	Mean Time	Mean $\hat{lk} - lk$
$n = 100$		
`ic_sp`	0.006	0.000
`intcox`: default	0.155	0.253
`intcox`: strict	1.142	0.010
$n = 500$		
`ic_sp`	0.032	0.000
`intcox`: default	0.351	1.797
`intcox`: strict	2.754	0.078
$n = 1,000$		
`ic_sp`	0.082	0.000
`intcox`: default	0.744	5.753
`intcox`: strict	6.036	0.165
$n = 5,000$		
`ic_sp`	0.787	0.000
`intcox`: default	4.258	87.65
`intcox`: strict	86.06	0.953

Table 1: Average computation times in seconds and average difference in final log likelihood from MLE. `intcox`: default refers to default settings of the `intcox` function, while `intcox`: strict refers to the same function but with a stricter convergence criteria.

$$\sum_{i=1}^{n} \log\left(f_m(\Lambda(l_i, H)|X_i, \beta) - f_m(\Lambda(r_i, H)|X_i, \beta)\right)$$

$$\text{such that } H_j \leq H_{j+1}$$

Each iteration of our algorithm included a conditional Newton-Raphson step which updates the regression parameters and an active set step which updated the baseline hazard function. In active set step, a pool-adjacent violators algorithm is used to optimize the baseline hazard function while still respecting the monotonic constraint of the cumulative hazard.

To examine the speed of the algorithms, we simulated interval censored data (via the function `simIC_weib`) with two covariates and sample sizes n = 100, 500, 1,000 and 5,000, where the true baseline distribution was weibull(2,2). For the censoring mechanism, we used a case

Proportional Odds	
$n = 100$	0.005
$n = 500$	0.018
$n = 1000$	0.034
$n = 5000$	0.198

Table 2: Average computation times in seconds for the `ic_sp` function with proportional odds model. Data simulated by `simIC_weib`.

II interval censoring mechanism in which two inspections were used, with inspection time simulated according to a uniform(0, 2.5) distrbution.

Fair comparison of the speed of our algorithm with that found in **intcox** is somewhat confounded by the fact that the algorithm called by `intcox` typically prematurely terminates, biasing the speed estimates. Even still, it is quite clear that our algorithm is both much faster and more reliable. For a complete comparison, we used both the default settings and also supplied a more strict convergence criteria to `intcox`. With each dataset, we fit `ic_sp`, `intcox` with default settings and `intcox` with stricter convergence criteria (referred to as `intcox`:default and `intcox`:strict respectively). For the `intcox`:strict, we set the argument `epsilon` $= 10^{-6}$ (this is refers to the required difference in likelihood between iterations for termination. Default is 10^{-4}). For each sample size and algorithm configuration, we recorded mean time to algorithm termination and mean difference of final likelihood for each configuration compared with the maximum likelihood of all three configurations. At each configuration, 100 simulated datasets were fit. The algorithm used in `intcox` very frequently failed with discrete censoring, reporting algorithm failure in the first iteration over half the time. Because of this, the speed of the `intcox` algorithm was not examined for discrete censoring. Results are presented on table 1. We found `ic_sp` significantly outperformed the other configurations in all situations. It appeared that the competitive speed advantage of `ic_sp` technically decreased as n increased (in fact, with $n = 5,000$, `intcox`:default stops much faster `ic_sp`), but this seems to be a result of the stopping criteria of `intcox` being too lax and preforming worse as n increases: even for `intcox`:strict, at $n = 5,000$, the average difference in final log likelihood implies that the results should not be trusted.

We also clocked the algorithm for proportional odds under similar scenarios, but simulated proportional odds event times rather than proportional hazards. There is no existing algorithm that we are aware of to clock against. Results are shown on table 2.

All simulations were run on a 2014 Macbook Pro with a 2.2 GHz i7 processor.

3. Algorithm for Parametric Models

When parametric models are considered, the likelihood function must treat uncensored observations in a distinct manner. In the case of the Cox-PH model, the 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 are the parameters associated with the baseline distribution, β are the regression parameters, X_i are an individual'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 censored.

For the proportional hazards model, the likelihood function can be written as

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

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

In all the parametric families we consider, 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, 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 mild conditions. 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 conjugate gradient step was used.

To compare this algorithm with an established implementation, we simulated 100 datasets with our function `simIC_weib` with $n = 100,000$ and fit the data with both our algorithm with a Cox-PH weibull model and **survival**'s `survreg` function. Although `survreg` fits a weibull aft model by default, this leads to the same maximum likelihood, as for the weibull distribution, the aft and Cox-PH models are identical up to a linear transformation of variables. Slight modification of the simulated data was required as `survreg` does not allow for the left side of intervals to be 0 or the right side to be ∞. The two implementations reliably found the same estimate; the maximum absolute difference in final log likelihood observed was 3.5×10^{-9}. It was found that `survreg` was slightly faster, with an average computation time of 0.79 seconds vs 3.88 seconds. Currently, our algorithm uses numeric derivatives and we believe that replacing them with analytically calculated derivatives could lead to similar speed as `survreg`.

4. 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 a semi-parametric model, the second assumption is dropped. Either way, it is important to assess the validity of the assumptions. With interval censored data, this can be fairly difficult. Our package includes easy to use routines for examining both sets of assumptions, both of which use the semi-parametric 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 semi-parametric 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, we take use 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 a semi-parametric model for each strata and plots the given transformation of the baseline survival functions. If 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.

5. Using icenReg

The core functions in **icenReg** are `ic_sp` and `ic_par` for fitting the semi- and fully-parametric Cox-PH models respectively. Results of the fitted models can be viewed with the standard `summary` method. Plots of estimated survival curves can be viewed via the `plot` method.

Alternatively, the estimated survival curves can be extracted by the `getSCurves` function. For the sake of demonstration, **icenReg** includes the function `simICPH_beta`, which simulates proportional hazards data with a beta baseline distribution and `mdata`, a real current status data set from Hoel and Walburg (1972).

The functions `ic_ph` and `impute_ic_ph` shares many arguments. In particular, they both include the standard arguments `formula` and `data`. In the `formula` argument, the response must be a `Surv` object with `type = "interval2"`. These functions also include the arguments `useMCores = F`, a logical indicator for whether multiple cores should be used for the imputation or bootstrap step and `seed = NULL`, a numeric value for which seed will be passed to `set.seed` before the bootstrap or imputation samples are taken. If `seed = NULL`, a random seed is generated and passed to `set.seed`. The arguments `bs_samples = 20` and `imps = 100` refer to the number of bootstrap or imputation samples to be taken. The default values are lower than recommended for inference; this is so users can estimate the time needed to run the necessary samples. Unique to `impute_ic_ph` are the arguments `eta = 1e-10` and `rightCenVal = 10000`. This is because the **survival** function `survreg` used for the imputation model cannot handle values of 0 or `Inf`, so values less than `eta` are replaced with `eta` and values greater than `rightCenVal` are replaced with `rightCenVal`. In some cases, it may be necessary to supply a larger value of `rightCenVal`.

The `summary` function is used in the standard manner. For the `plot` function for `ic_ph` fits, if only the fit is provided, the baseline survival distribution will be plotted. Alternatively, the second argument supplied to `plot` can be a new data.frame, with matching variables names to those found in the original dataset. In this case, `plot` will plot the survival curve for each set of covariates. A legend will be plotted using the rownames of the data.frame. For custom plots, the function `getSCurves` can be used to extract the survival curves, with similar arguments to plot (*i.e.* first argument is the fit, optional second argument can be a new data.frame).

For demonstrative purposes, the package includes a function simICPH_beta, which simulates data from a Cox-PH model with a beta baseline distribution. The arguments accepted are n, the number of subjects simulated, b1 and b2, the coefficients for the covariates, inspections = 1, the number of inspection times (default is one inspection, resulting in current status data), shape1 and shape2, the parameters of the baseline distribution and rndDigits, the number of digits that the inspection times will be round to (rndDigits = NULL implies no rounding). The covariates associated with b1 and b2 are distributed rnorm(n) and rbinom(n, 1, 0.5) - 0.5 respectively. The reason a beta distribution was chosen is that this will lead to the imputation model assuming a very inappropriate baseline distribution (Weibull) and one can see that if the number of inspections is low, the imputation model will lead to heavy bias. However, if the number of inspections is high, we will see the bias is fairly small. We leave implementation of this to the curious reader.

In order to use parallel computation of either bootstrap or imputation samples, we take advantage of the **foreach** package (Revolution Analytics and Steve Weston (2014b)). The user must make and register a cluster via **doParallel**'s makeCluster and registerDoParallel (Revolution Analytics and Steve Weston (2014a)). We demonstrate this in the next section.

6. Example Analysis

For our example, we will use the dataset mdata provided in our package. This dataset borrows from Hoel and Walburg (1972). In this study, RFM mice were sacrificed and examined for lung tumors. If tumors were found, the mouse was considered left censored with the inspection time being equal to age at sacrifice. If no tumors were found, the mouse was considered right censored. Mice were placed in two different environments: conventional environments and germ free environments. In the mdata dataset, l represents the left side of the observation interval, u represents the right side and grp is a factor representing the environment the mouse was placed in.

Below we analyze the data using both ic_ph and impute_ic_ph. To begin with, we will use the package **doParallel** to set up a cluster.

```
> library(doParallel)
> myCluster <- makeCluster(3, type = 'FORK')
> registerDoParallel(myCluster)
```

Next, we will load the data and fit the two models.

```
data(mdata)
sp_fit <- ic_ph(Surv(l, u, type = 'interval2') ~ grp,
        data = mdata, bs_samples = 500, useMCores = T)
mi_fit <- impute_ic_ph(Surv(l, u, type = 'interval2') ~ grp,
        data = mdata, imps = 500, useMCores = T)
```

We viewed the results of the two fits with the summary function.

```
> summary(sp_fit)

Semi Parameteric Cox PH model for interval censored data
Call =
ic_ph(formula = Surv(l, u, type = "interval2") ~ grp, data = mdata,
    bs_samples = 500)

    Estimate Exp(Est) Std. Error z-value      p
grp   0.6277    1.873     0.4087   1.536 0.1246

final llk =  -76.53436
Iterations =  26
Bootstrap samples =  500
> summary(mi_fit)

Multiple Imputations Cox PH model for interval censored data
Call =
impute_ic_ph(formula = Surv(l, u, type = "interval2") ~ grp,
    data = mdata, imps = 500)

      Estimate Exp(Est) Std. Error z-value       p
grpge   0.7409    2.098     0.3135   2.363 0.01811

number of imputations =  500
```

For visualization, we plot the `ic_ph` fit for the two groups.

```
> newdata <- data.frame(grp = c('ce', 'ge'))
> rownames(newdata) = c('Conventional Environment', 'Germ Free Environment')
> plot(sp_fit, newdata)
> sCurves <- getSCurves(sp_fit, newdata)
```

We note that the imputation model results in slightly different estimates than the semi-parametric model, and slightly higher standard errors. This is due to the relative wide observational intervals, leading to heavy influence from the imputation model.

Based on our findings, we would conclude that while it was estimated that mice in the germ free environment were at higher risk to develop tumors than those in the conventional environment, we did not find enough evidence to reject the notation that the hazard rates might be equal. This implies we trusted the results of the semi-parametric model over the parametric imputation model. We decided on this because the observation intervals were relatively wide, and so the parametric imputation model could result in heavy biases.

7. Future Expansions

We have several plans to expand **icenReg** to allow for easier analysis of interval censored data.

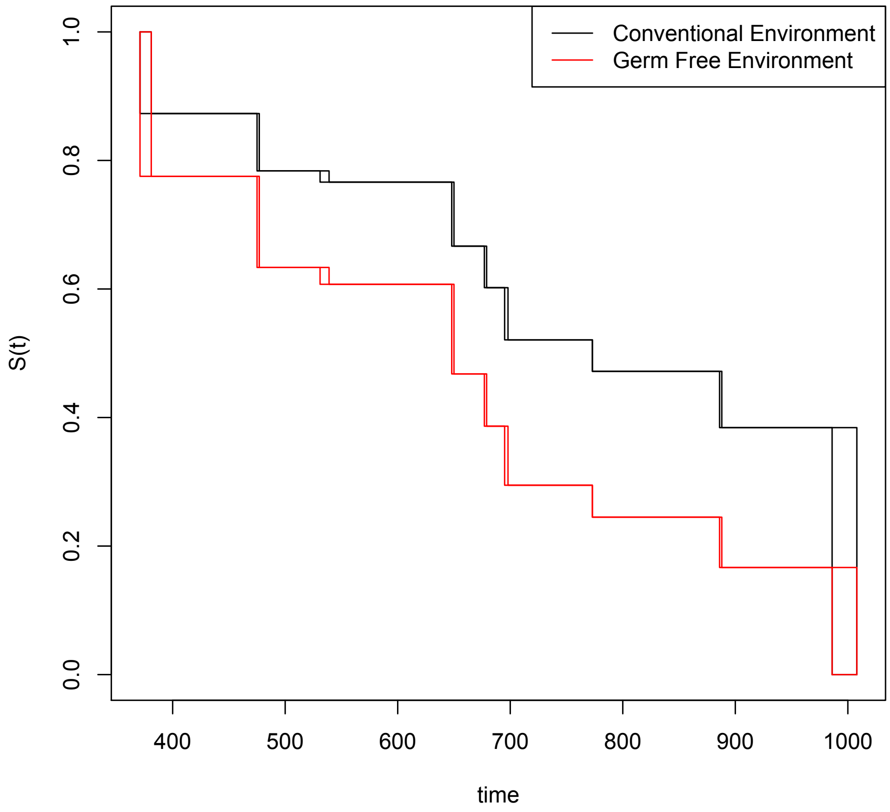

Figure 1: Plotted survival curves for `mdata` dataset

In particular, we would like to provide user-friendly tools for model assessment, such as single function for stratifying the data on different levels of a covariate to examine the proportional hazards assumption. We are also interested in adding a semi-parametric proportional odds model, which should be able to use a similar algorithm as the proportional hazards model.

References

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.

Bohning D (1986). "A Vertex-exchange-method in D-optimal Design Theory." *Metrika*, **33**, 337–347.

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

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

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.

Gomez G, Calle M, Oller R, Langohr K (2009). "Tutorial on methods for interval-censored data and their implementation in R." *Statistical Modeling*, **9(4)**, 259.

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.

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.

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

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.

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

Pan W (1999b). "A multiple imputations approach to Cox regression with interval-censored data." *Biometrics*, **12**, 133–146.

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 (2014a). *doParallel: Foreach parallel adaptor for the parallel package*. R package version 1.0.8, URL http://CRAN.R-project.org/package=doParallel.

Revolution Analytics and Steve Weston (2014b). *foreach: Foreach looping construct for R*. R package version 1.4.2, URL http://CRAN.R-project.org/package=foreach.

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 DB (1976). "Inference and Missing Data." *Biometrika*, **63**, 581?592.

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, 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, Series B*, **38**, 290–295.

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.

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.

Yu Y (2011). "D-optimal designs via a cocktail algorithm." *Statistics and Computing*, **21**, 475–481.

Affiliation:

Clifford Anderson-Bergman
Gladstone Institutes
University of California, San Francisco
1650 Owens Street
San Francisco, CA
E-mail: cliff.andersonbergman@gladstone.ucsf.edu