Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

casebase: An Alternative Framework For Survival Analysis

Sahir Bhatnagar *
McGill University

Maxime Turgeon *
University of Manitoba

Jesse Islam
McGill University

James Hanley
McGill University

Olli Saarela
University of Toronto

Abstract

The abstract of the article. * joint co-authors

Keywords: keywords, not capitalized, Java.

1. Introduction

Survival analysis has been greatly influenced over the last 50 years by the partial likelihood approach of the Cox proportional hazard model (Cox 1972). This approach provides a flexible way of assessing the influence of covariates on the hazard function, without the need to specify a parametric survival model. This flexibility comes at the cost of decoupling the baseline hazard from the effect of the covariates. To recover the whole survival curve—or the cumulative incidence function (CIF)—we then need to separately estimate the baseline hazard (Breslow 1972). This in turn often leads to stepwise estimates of the survival function.

From the perspective of clinicians and their patients, the most relevant quantity is often the 5- or 10-year risk of having a certain event given the patient's particular circumstances, and not the hazard ratio between a treatment and control group. Therefore, to make sound clinical decisions, it is important to accurately estimate the *full* hazard function, which can then be used to estimate the cumulative incidence function (CIF). From this perspective, the CIF and the survival function estimates are now smooth functions of time.

With the goal of fitting smooth-in-time hazard functions, Hanley & Miettinen (2009) proposed a general framework for estimating fully parametric hazard models via logistic regression.

Their approach provides users that are familiar with generalized linear models with a natural way of fitting parametric survival models. Moreover, their framework is very flexible: general functions of time can be estimated (e.g. using splines or general additive models), and hence these models retain some of the flexibility of partial likelihood approaches.

In this article, we present an R package that combines the ideas of Hanley & Miettinen into a simple interface. The purpose of the **casebase** package is to provide practitioners with an easy-to-use software tool to predict the risk (or cumulative incidence) of an event, conditional on a particular patient's covariate profile. Our package retains the flexibility of case-base sampling and the familiar interface of the `glm` function.

In what follows, we first recall some theoretical details on case-base sampling and its use for estimating parametric hazard functions. We then give a short review of existing R packages that implement comparable features as **casebase**. Next, we provide some details about the implementation of case-base sampling in our package, and we give a brief survey of its main functions. This is followed by four case studies that illustrate the flexibility and capabilities of **casebase**. We show how the same framework can be used for competing risk analyses, penalized estimation, and for studies with time-dependent exposures. Finally, we end the article with a discussion of the results and of future directions.

2. Theoretical details

As discussed in Hanley & Miettinen (2009), the key idea behind case-base sampling is to discretize the study base into an infinite amount of *person moments*. These person moments are indexed by both an individual in the study and a time point, and therefore each person moment has a covariate profile, an exposure status and an outcome status attached to it. We note that there is only a finite number of person moments associated with the event of interest (what Hanley & Miettinen call the *case series*). The case-base sampling refers to the sampling from the base of a representative finite sample called the *base series*.

To describe the theoretical foundations of case-base sampling, we use the framework of counting processes. In what follows, we abuse notation slightly and omit any mention of σ-algebras; the interested reader can refer to Saarela & Arjas (2015) and Saarela (2016). First, let $N_i(t) \in \{0, 1\}$ be counting processes corresponding to the event of interest for individual $i = 1, \ldots, n$. For simplicity, we will consider Type I censoring due to the end of follow-up at time τ (the general case of non-informative censoring is treated in Saarela (2016)). We assume a continuous time model, which implies that the counting process jumps are less than or equal to one. We are interested in modeling the hazard functions $\lambda_i(t)$ of the processes $N_i(t)$, and which satisfy

$$\lambda_i(t)dt = E[dN_i(t)].$$

Next, we model the base series sampling mechanism using non-homogeneous Poisson processes $R_i(t) \in \{0, 1, 2, \ldots\}$, with the person-moments where $dR_i(t) = 1$ constituting the base series. The process $Q_i(t) = R_i(t) + N_i(t)$ then counts both the case and base series person-moments contributed by individual i. This process is typically defined by the user via its hazard function $\rho_i(t)$. The process $Q_i(t)$ is characterized by $E[dQ_i(t)] = \lambda_i(t)dt + \rho_i(t)dt$.

If the hazard function $\lambda_i(t; \theta)$ is parametrized in terms of θ, we could define an estimator $\hat{\theta}$

by maximization of the likelihood expression

$$L_0(\theta) = \prod_{i=1}^{n} \exp\left\{-\int_0^{\tau} \lambda_i(t;\theta)dt\right\} \prod_{i=1}^{n} \prod_{t\in[0,\tau)} \lambda_i(t;\theta)^{dN_i(t)} \left(1-\lambda_i(t;\theta)\right)^{1-dN_i(t)},$$

where $\prod_{t\in[0,u)}$ represents a product integral from 0 to u. However, the integral over time makes the computation and maximisation of $L_0(\theta)$ challenging.

Case-base sampling allows us to avoid this integral. By conditioning on a sampled person-moment, we get individual contributions of the form

$$P(dN_i(t) \mid dQ_i(t)=1) \overset{\theta}{\propto} \frac{\lambda_i(t;\theta)^{dN_i(t)}}{\rho_i(t)+\lambda_i(t;\theta)}.$$

Therefore, we can define an estimating equation for θ as follows:

$$L(\theta) = \prod_{i=1}^{n} \prod_{t\in[0,\tau)} \left(\frac{\lambda_i(t;\theta)^{dN_i(t)}}{\rho_i(t)+\lambda_i(t;\theta)}\right)^{dQ_i(t)}.$$

When a logarithmic link function is used for modeling the hazard function, the above expression is of a logistic regression form with an offset term $\log(1/\rho_i(t))$. Note that the sampling units selected in the case-base sampling mechanism are person-moments, rather than individuals, and the parameters to be estimated are hazards or hazard ratios rather than odds or odds ratios. Generally, an individual can contribute more than one person-moment, and thus the terms in the product integral are not independent. Nonetheless, Saarela (2016) showed that the logarithm of this estimating function has mean zero at the true value $\theta = \theta_0$, and that the resulting estimator $\hat{\theta}$ is asymptotically normally distributed.

In Hanley & Miettinen (2009), the authors suggest sampling the base series *uniformly* from the study base. In terms of Poisson processes, this sampling strategy corresponds essentially to a time-homogeneous Poisson process with hazard equal to $\rho_i(t) = b/B$, where b is the number of sampled observations in the base series, and B is the total population-time for the study base. More complex examples are also available; see for example Saarela & Arjas (2015), where the probabilities of the sampling mechanism are proportional to the cardiovascular disease event rate given by the Framingham score. With this sampling mechanism, Saarela & Arjas are able to increase the efficiency of their estimators, when compared to uniform sampling.

Let $g(t;X)$ be the linear predictor such that $\log(\lambda(t;X)) = g(t;X)$. Different functions of t lead to different parametric hazard models. The simplest of these models is the one-parameter exponential distribution which is obtained by taking the hazard function to be constant over the range of t:

$$\log(\lambda(t;X)) = \beta_0 + \beta_1 X.$$

The instantaneous failure rate is independent of t.[1]

The Gompertz hazard model is given by including a linear term for time:

[1] The conditional chance of failure in a time interval of specified length is the same regardless of how long the individual has been in the study. This is also known as the *memoryless property* (Kalbfleisch and Prentice 2011).

$$\log(\lambda(t; X)) = \beta_0 + \beta_1 t + \beta_2 X.$$

Use of $\log(t)$ yields the Weibull hazard which allows for a power dependence of the hazard on time (Kalbfleisch and Prentice 2011):

$$\log(\lambda(t; X)) = \beta_0 + \beta_1 \log(t) + \beta_2 X.$$

For competing-risk analyses with J possible events, we can show that each person-moment's contribution of the likelihood is of the form

$$\frac{\lambda_j(t)^{dN_j(t)}}{\rho(t) + \sum_{j=1}^J \lambda_j(t)},$$

where $N_j(t)$ is the counting process associated with the event of type j and $\lambda_j(t)$ is the corresponding hazard function. As may be expected, this functional form is similar to the terms appearing in the likelihood function for multinomial regression.[2]

3. Existing packages

Survival analysis is an important branch of applied statistics and epidemiology. Accordingly, there is a vast ecosystem of `R` packages implementing different methods. In this section, we describe how the functionalities of **casebase** compare to these packages.

At the time of writing, a cursory examination of CRAN's task view on survival analysis reveals that there are over 250 packages related to survival analysis (2019). For the purposes of this article, we restricted our description to packages that implement at least one of the following features: parametric modeling, non-proportional hazard models, competing risk analysis, penalized estimation, and cumulative incidence curve estimation. By searching for appropriate keywords in the `DESCRIPTION` file of these packages, we found 60 relevant packages. These 60 packages were then manually examined to determine which ones are comparable to **casebase**. In particular, we excluded packages that were focused on a different set of problems, such as frailty and multi-state models. The remaining 14 packages appear in Table 1, along with some of the functionalities they offer.

Several packages implement penalized estimation for the Cox model: **glmnet** (2011), **glmpath** (2018), **penalized** (2010), **RiskRegression** (2019). Moreover, some packages also include penalized estimation in the context of Cox models with time-varying coefficients: elastic-net penalization with **CoxRidge** (2015) and **rstpm2** (2019), while **survival** (2015) has an implementation for ridge. **casebase** provides penalized estimation in the context of parametric hazards. To our knowledge, **casebase** and **rsptm2** are the only packages to offer this functionality. I can dive deeper to determine if this is a fair statement

Parametric survival models are implemented in a handful of packages: **CFC** (2019), **flexsurv** (2016), **SmoothHazard** (2017), **rsptm2** (2019), **mets** (2014), and **survival**. The types of models

[2] Specifically, it corresponds to the following parametrization:

$$\log\left(\frac{P(Y = j \mid X)}{P(Y = J \mid X)}\right) = X^T\beta_j, \qquad j = 1, \ldots, J-1.$$

they allow vary for each package. For example, **SmoothHazard** is limited to Weibull distributions (2017), whereas both **flexsurv** and **survival** allow users to supply any distribution of their choice. Also, **flexsurv**, **smoothhazard**, **mets** and **rstpm2** also have the ability to model the effect of time using splines, which allows flexible modeling of the hazard function. Moreover, **flexsurv** has the ability to estimate both scale and shape parameters for a variety of parametric families (see Table 2). As discussed above, **casebase** can model any parametric family whose log-hazard can be expressed as a linear model of covariates (including time). Therefore, our package allows the user to model the effect of time using splines, and through interaction terms involving covariates and time, it also allows user to fit time-varying coefficient models. However, we do not explicitly model any shape parameter, unlike **flexsurv**.

Several R packages implement methodologies for competing risk analysis; for a different perspective, see Mahani & Sharabiani (2019). The package **cmprsk** provides methods for cause-specific subdistributions, such as in the Fine-Gray model (1999). On the other hand, the package **CFC** estimates cause-specific cumulative incidence functions from unadjusted, non-parametric survival functions. Our package **casebase** also provides functionalities for competing risk analysis by estimating the cause-specific hazards, and from these quantities, we can derive the cause-specific cumulative incidence functions.

Finally, several packages include functions to estimate the cumulative incidence function. The corresponding methods generally fall into two categories: transformation of the estimated hazard function, and semi-parametric estimation of the baseline hazard. The first category broadly corresponds to parametric survival models, where the full hazard is explictly modeled. Using this estimate, the survival function and the cumulative incidence function can be obtained using their functional relationships (see Equations 1 and 2 below). Packages including this functionality include **CFC**, **Flexsurv**, **mets**, and **survival**. Our package **casebase** also follows this approach for both single-event and competing-event analyses. The second category outlined above broadly corresponds to Cox models. These models do not model the full hazard function, and therefore the baseline hazard needs to be estimated separately in order to estimate the survival function. This is achieved using semi-parametric estimators (e.g. Breslow's estimator). Packages that implement this approach include **RiskRegression**, **rstpm2**, and **survival**. As mentioned in the introduction, a key distinguishing factor between these two approaches is that the first category leads to smooth estimates of the cumulative incidence function, whereas the second category produces estimates in the form of step-wise functions. This was one of the main motivations for introducting case-base sampling in survival analysis.

Package	Competing risks	Non-proportional	Penalization	Splines	Parametric	Semi-parametric	Interval/left censoring	Absolute risk
casebase Hanley and Miettinen (2009)	x	x	x	x	x			x
CFC Mahani and Sharabiani (2019)	x	x			x			x
textbfcmprsk Gray (2019)	x					x		x
coxRidge Perperoglou (2015)		x	x			x		
crrp Fu (2015)	x		x					
fastcox Yi and Zou (2017)			x			x		
Flexrsurv Clerc-Urmès, Grzebyk, and Hédelin (2017)		x		x	x			x
Flexsurv Jackson (2016)	x	x		x	x			x
glmnet Simon *et al.* (2011)			x			x		
glmpath Park and Hastie (2018)			x			x		
mets Scheike *et al.* (2014)	x			x		x		x
penalized Goeman, Meijer, Chaturvedi, and Lueder (2019)			x			x		
RiskRegression Gerds *et al.* (2019)			x			x		x
Rstpm2 Clements *et al.* (2019)		x	x	x	x	x	x	x
SmoothHazard Touraine *et al.* (2017)		x		x	x		x	
Survival Therneau (2015)	x	x			x	x	x	x

Table 1: Different features of interest in various survival packages.

	Parameters (location in `red`)	Density R function	`dist`
Exponential	`rate`	`dexp`	`"exp"`
Weibull (accelerated failure time)	`shape, scale`	`dweibull`	`"weibull"`
Weibull (proportional hazards)	`shape, scale`	`dweibullPH`	`"weibullPH"`
Gamma	`shape, rate`	`dgamma`	`"gamma"`
Log-normal	`meanlog, sdlog`	`dlnorm`	`"lnorm"`
Gompertz	`shape, rate`	`dgompertz`	`"gompertz"`
Log-logistic	`shape, scale`	`dllogis`	`"llogis"`
Generalized gamma (Prentice 1975)	`mu, sigma, Q`	`dgengamma`	`"gengamma"`
Generalized gamma (Stacy 1962)	`shape, scale, k`	`dgengamma.orig`	`"gengamma.orig"`
Generalized F (stable)	`mu, sigma, Q, P`	`dgenf`	`"genf"`
Generalized F (original)	`mu, sigma, s1, s2`	`dgenf.orig`	`"genf.orig"`

Table 2: Built-in parametric survival distributions in **flexsurv**.

4. Implementation details

The functions in the casebase package can be divided into two categories: 1) data visualization, in the form of population-time plots; and 2) parametric modeling. We explicitly aimed at being compatible with both `data.frame`s and `data.table`s. This is evident in some of the coding choices we made, and it is also reflected in our unit tests.

4.1. Population-time plots

4.2. Parametric modeling

The parametric modeling step was separated into three parts:

1. case-base sampling;
2. estimation of the smooth hazard function;
3. calculation of the risk function.

By separating the sampling and estimation functions, we allowed the possibility of users implementing more complex sampling scheme, as described in Saarela (2016).

The sampling scheme selected for `sampleCaseBase` was described in Hanley & Miettinen (2009): we first sample along the “person” axis, proportional to each individual’s total follow-up time, and then we sample a moment uniformly over their follow-up time. This sampling scheme is equivalent to the following picture: imagine representing the total follow-up time of all individuals in the study along a single dimension, where the follow-up time of the next individual would start exactly when the follow-up time of the previous individual ends. Then the base series could be sampled uniformly from this one-dimensional representation of the overall follow-up time. In any case, the output is a dataset of the same class as the input, where each row corresponds to a person-moment. The covariate profile for each such person-moment is retained, and an offset term is added to the dataset. This output could then be used to fit a smooth hazard function, or for visualization of the base series.

The fitting function `fitSmoothHazard` starts by looking at the class of the dataset: if it was generated from `sampleCaseBase`, it automatically inherited the class `cbData`. If the dataset supplied to `fitSmoothHazard` does not inherit from `cbData`, then the fitting function starts by calling `sampleCaseBase` to generate the base series. In other words, users can bypass `sampleCaseBase` altogether and only worry about the fitting function `fitSmoothHazard`.

The fitting function retains the familiar formula interface of `glm`. The left-hand side of the formula should be the name of the column corresponding to the event type. The right-hand side can be any combination of the covariates, along with an explicit functional form for the time variable. Note that non-proportional hazard models can be achieved at this stage by adding an interaction term involving time (cf. Case Study 4 below). The offset term does not need to be specified by the user, as it is automatically added to the formula before calling `glm`.

To fit the hazard function, we provide several approaches that are available via the `family` parameter. These approaches are:

- `glm`: This is the familiar logistic regression.
- `glmnet`: This option allows for variable selection using Lasso or elastic-net. This functionality is provided through the `glmnet` package (Friedman, Hastie, and Tibshirani 2010).
- `gam`: This option provides support for *Generalized Additive Models* via the `gam` package (Hastie and Tibshirani 1987).
- `gbm`: This option provides support for *Gradient Boosted Trees* via the `gbm` package. This feature is still experimental.

In the case of multiple events, the hazard is fitted via multinomial regression as performed by the **VGAM** package. We selected this package for its ability to fit multinomial regression models with an offset.

Once a model-fit object has been returned by `fitSmoothHazard`, all the familiar summary and diagnostic functions are available: `print`, `summary`, `predict`, `plot`, etc. Our package provides one more functionality: it computes risk functions from the model fit. For the case of a single event, it uses the familiar identity

$$S(t) = \exp\left(-\int_0^t \lambda(u;X)du\right). \tag{1}$$

The integral is computed using either the `stats::integrate` function or Monte-Carlo integration. The risk function (or cumulative incidence function) is then defined as

$$CI(t) = 1 - S(t). \tag{2}$$

For the case of a competing-event analysis, the event-specific risk is computed using the following procedure: first, we compute the overall survival function (i.e. for all event types):

$$S(t) = \exp\left(-\int_0^t \Lambda(u;X)du\right), \qquad \Lambda(t;X) = \sum_{j=1}^{J} \lambda_j(t;X).$$

From this, we can derive the event-specific subdensities:

$$f_j(t) = \lambda_j(t)S(t).$$

Finally, by integrating these subdensities, we obtain the event-specific cumulative incidence functions:

$$CI_j(t) = \int_0^t f_j(u)du.$$

The integrals are computed using either numerical integration (via the trapezoidal rule) or Monte Carlo integration. This option is controlled by the argument `method` of the `absoluteRisk` function.

We created `absoluteRisk` as an `S3` generic, with methods that match the different types of outputs of `fitSmoothHazard`. The method dispatch system of R then takes care of matching the correct output to the correct methodology for calculating the cumulative incidence function, without requiring any intervention from the user.

In the following sections, we illustrate these functionalities in the context of four case studies.

5. Case study 1–European Randomized Study of Prostate Cancer Screening

For the first case study, we will use of the European Randomized Study of Prostate Cancer Screening data. This dataset is available through the **casebase** package:

```
R> data(ERSPC)
R> ERSPC$ScrArm <- factor(ERSPC$ScrArm,
R>   levels = c(0, 1),
R>   labels = c("Control group", "Screening group")
R> )
```

The results of this study were published by (Schröder, Hugosson, Roobol, Tammela, Ciatto, Nelen, Kwiatkowski, Lujan, Lilja, Zappa *et al.* 2009), and the dataset itself was obtained using the approach described in (Liu, Rich, and Hanley 2014).

Population time plots can be extremely informative graphical displays of survival data. They should be the first step in an exploratory data analysis. We facilitate this task in the **casebase** package using the `popTime` function. We first create the necessary dataset for producing the population time plots, and we can generate the plot by using the corresponding `plot` method:

```
R> pt_object <- casebase::popTime(ERSPC, event = "DeadOfPrCa")
R> plot(pt_object)
```

We can also create exposure stratified plots by specifying the `exposure` argument in the `popTime` function:

```
R> pt_object_strat <- casebase::popTime(ERSPC,
R>   event = "DeadOfPrCa",
R>   exposure = "ScrArm"
R> )
R> plot(pt_object_strat)
```

We can also plot them side-by-side using the `ncol` argument:

```
R> plot(pt_object_strat, ncol = 2)
```

ADD PARAGRAPH ABOUT WHAT WE CAN CONCLUDE FROM THESE GRAPHS.

Next, we investigate the differences between the control and the screening arms. A common choice for this type of analysis is to use a Cox regression model and estimate the hazard ratio for the screening group (relative to the control group). In R, it can be done as follows:

```
R> cox_model <- survival::coxph(Surv(Follow.Up.Time, DeadOfPrCa) ~ ScrArm,
R>   data = ERSPC
R> )
R> summary(cox_model)
```

We can also plot the cumulative incidence function (CIF) for each group. However, this involves estimating the baseline hazard, which Cox regression treated as a nuisance parameter.

```
R> new_data <- data.frame(
R>   ScrArm = c("Control group", "Screening group"),
R>   ignore = 99
R> )
R>
R> plot(survfit(cox_model, newdata = new_data),
R>   xlab = "Years since Randomization",
R>   ylab = "Cumulative Incidence (%)",
R>   fun = "event",
R>   xlim = c(0, 15), conf.int = FALSE, col = c("red", "blue"),
R>   main = sprintf(
R>     "Estimated Cumulative Incidence (risk) of Death from Prostate
R>                    Cancer Screening group Hazard Ratio: %.2g (%.2g, %.2g)",
R>     exp(coef(cox_model)),
R>     exp(confint(cox_model))[1],
R>     exp(confint(cox_model))[2]
R>   ),
R>   ylim = c(0, 2), yscale = 100
R> )
R> legend("topleft",
R>   legend = c("Control group", "Screening group"),
R>   col = c("red", "blue"),
R>   lty = c(1, 1),
R>   bg = "gray90"
R> )
```

Next, we show how case-base sampling can be used to carry out a similar analysis. We fit several models that differ in how we choose to model time.

First, we will fit an exponential model. Recall that this corresponds to excluding time from the linear predictor.

```
R> casebase_exponential <- casebase::fitSmoothHazard(DeadOfPrCa ~ ScrArm,
R>   data = ERSPC,
R>   ratio = 100
R> )
R>
R> summary(casebase_exponential)
R> exp(coef(casebase_exponential)[2])
R> exp(confint(casebase_exponential)[2, ])
```

We can then use the `absoluteRisk` function to get an estimate of the cumulative incidence curve for a specific covariate profile. In the plot below, we overlay the estimated CIF from the exponential model on the Cox model CIF:

```
R> smooth_risk_exp <- casebase::absoluteRisk(
R>   object = casebase_exponential,
R>   time = seq(0, 15, 0.1),
R>   newdata = new_data
R> )
R>
R> plot(survfit(cox_model, newdata = new_data),
R>   xlab = "Years since Randomization",
R>   ylab = "Cumulative Incidence (%)",
R>   fun = "event",
R>   xlim = c(0, 15), conf.int = FALSE, col = c("red", "blue"),
R>   main = sprintf(
R>     "Estimated Cumulative Incidence (risk) of Death from Prostate
R>                     Cancer Screening group Hazard Ratio: %.2g (%.2g, %.2g)",
R>     exp(coef(cox_model)),
R>     exp(confint(cox_model))[1],
R>     exp(confint(cox_model))[2]
R>   ),
R>   ylim = c(0, 2), yscale = 100
R> )
R> lines(smooth_risk_exp[, 1], smooth_risk_exp[, 2], col = "red", lty = 2)
R> lines(smooth_risk_exp[, 1], smooth_risk_exp[, 3], col = "blue", lty = 2)
R>
R>
R> legend("topleft",
R>   legend = c(
R>     "Control group (Cox)", "Control group (Casebase)",
R>     "Screening group (Cox)", "Screening group (Casebase)"
R>   ),
R>   col = c("red", "red", "blue", "blue"),
R>   lty = c(1, 2, 1, 2),
R>   bg = "gray90"
R> )
```

As we can see, the exponential model gives us an estimate of the hazard ratio that is similar to the one obtained using Cox regression. However, the CIF estimates are quite different. Based on what we observed in the population time plot, where more events are observed later on in time, we do not expect a constant hazard would be a good description for this data. In other words, this poor fit for the exponential hazard is expected. A constant hazard model would overestimate the cumulative incidence earlier on in time, and underestimate it later on; this is what we see on the cumulative incidence plot. This example demonstrates the benefits of population time plots as an exploratory analysis tool.

For the next model, we include time as a linear term. Recall that this corresponds to a Weibull model.

```
R> casebase_time <- fitSmoothHazard(DeadOfPrCa ~ Follow.Up.Time + ScrArm,
R>   data = ERSPC,
R>   ratio = 100
R> )
R>
R> summary(casebase_time)
R> exp(coef(casebase_time))
R> exp(confint(casebase_time))
```

Again, the estimate of the hazard ratio is similar to that obtained from Cox regression. We then look at the estimate of the CIF:

```
R> smooth_risk_time <- casebase::absoluteRisk(
R>   object = casebase_time,
R>   time = seq(0, 15, 0.1),
R>   newdata = new_data
R> )
R>
R> plot(survfit(cox_model, newdata = new_data),
R>   xlab = "Years since Randomization",
R>   ylab = "Cumulative Incidence (%)",
R>   fun = "event",
R>   xlim = c(0, 15), conf.int = FALSE, col = c("red", "blue"),
R>   main = sprintf(
R>     "Estimated Cumulative Incidence (risk) of Death from Prostate
R>                     Cancer Screening group Hazard Ratio: %.2g (%.2g, %.2g)",
R>     exp(coef(cox_model)),
R>     exp(confint(cox_model))[1],
R>     exp(confint(cox_model))[2]
R>   ),
R>   ylim = c(0, 2), yscale = 100
R> )
R> lines(smooth_risk_time[, 1], smooth_risk_time[, 2], col = "red", lty = 2)
R> lines(smooth_risk_time[, 1], smooth_risk_time[, 3], col = "blue", lty = 2)
R>
R> legend("topleft",
R>   legend = c(
R>     "Control group (Cox)", "Control group (Casebase)",
R>     "Screening group (Cox)", "Screening group (Casebase)"
R>   ),
R>   col = c("red", "red", "blue", "blue"),
R>   lty = c(1, 2, 1, 2),
R>   bg = "gray90"
R> )
```

We see that the Weibull model leads to a better fit of the CIF than the Exponential model, when compared with the semi-parametric approach.

Finally, we can model the hazard as a smooth function of time using the `splines` package:

```
R> casebase_splines <- fitSmoothHazard(DeadOfPrCa ~ bs(Follow.Up.Time) + ScrArm,
R>   data = ERSPC,
R>   ratio = 100
R> )
R>
R> summary(casebase_splines)
R> exp(coef(casebase_splines))
R> exp(confint(casebase_splines))
```

Once again, we can see that the estimate of the hazard ratio is similar to that from the Cox regression. We then look at the estimate of the CIF:

```
R> smooth_risk_splines <- absoluteRisk(
R>   object = casebase_splines,
R>   time = seq(0, 15, 0.1),
R>   newdata = new_data
R> )
R>
R> plot(survfit(cox_model, newdata = new_data),
R>   xlab = "Years since Randomization",
R>   ylab = "Cumulative Incidence (%)",
R>   fun = "event",
R>   xlim = c(0, 15), conf.int = FALSE, col = c("red", "blue"),
R>   main = sprintf(
R>     "Estimated Cumulative Incidence (risk) of Death from Prostate
R>                     Cancer Screening group Hazard Ratio: %.2g (%.2g, %.2g)",
R>     exp(coef(cox_model)),
R>     exp(confint(cox_model))[1],
R>     exp(confint(cox_model))[2]
R>   ),
R>   ylim = c(0, 2), yscale = 100
R> )
R> lines(smooth_risk_splines[, 1], smooth_risk_splines[, 2], col = "red", lty = 2)
R> lines(smooth_risk_splines[, 1], smooth_risk_splines[, 3], col = "blue", lty = 2)
R>
R> legend("topleft",
R>   legend = c(
R>     "Control group (Cox)", "Control group (Casebase)",
R>     "Screening group (Cox)", "Screening group (Casebase)"
R>   ),
R>   col = c("red", "red", "blue", "blue"),
R>   lty = c(1, 2, 1, 2),
R>   bg = "gray90"
R> )
```

Qualitatively, the combination of splines and case-base sampling produces the parametric model that most closely resembles the Cox model. In the following table, we see that the confidence intervals are also similar across all four models. In other words, this reinforces the idea that, under proportional hazards, we do not need to model the full hazard to obtain reliable estimates of the hazard ratio. Of course, different parametric models for the hazards give rise to qualitatively different estimates for the CIF.

As noted above, the usual asymptotic results hold for likelihood ratio tests built using case-base sampling models. Therefore, we can easily test the null hypothesis that the exponential model is just as good as the larger (in terms of number of parameters) splines model.

```
R> anova(casebase_exponential, casebase_splines, test = "LRT")
```

As expected, we see that splines model provides a better fit. Similarly, we can compare the AIC for all three case-base models:

```
R> c(
R>   "Exp." = AIC(casebase_exponential),
R>   "Weib." = AIC(casebase_time),
R>   "Splines" = AIC(casebase_splines)
R> )
```

Once again, we have evidence that the splines model provides the best fit.

With this first case study, we saw how population-time plots can provide a useful summary of the incidence for our event of interests. In particular, we saw how it can provide evidence against an exponential hazard model. We then explored how different parametric models can provide different estimates of the CIF, even though they all give similar estimates for the hazard ratios.

6. Case study 2–Bone-marrow transplant

In the next case study, we show how case-base sampling can be used in the context of a competing risk analysis. For illustrative purposes, we will use the same data that was used in Scrucca *et al* (2010). The data was downloaded from the first author's website, and it is also available as part of the **casebase** package.

```
R>
R> data(bmtcrr)
```

The data contains information on 177 patients who received a stem-cell transplant for acute leukemia. The event of interest is relapse, but other competing causes (e.g. transplant-related death) were also recorded. Several covariates were captured at baseline: sex, disease type (acute lymphoblastic or myeloblastic leukemia, abbreviated as ALL and AML, respectively), disease phase at transplant (Relapse, CR1, CR2, CR3), source of stem cells (bone marrow and peripheral blood, coded as BM+PB, or only peripheral blood, coded as PB), and age. A summary of these baseline characteristics appear in Table **??**. We note that the statistical summaries were generated differently for different variable types: for continuous variables, we gave the range, followed by the mean and standard deviation; for categorical variables, we gave the counts for each category.

First, we can look at a population-time plot to visualize the incidence density of both relapse and the competing events. In Figure **??**, failure times are highlighted on the plot using red dots for the event of interest (panel A) and blue dots for competing events (panel B). In both panels, we see evidence of a non-constant hazard function: the density of points is larger at the beginning of follow-up than at the end.

Our main objective is to compute the absolute risk of relapse for a given set of covariates. We start by fitting a smooth hazard to the data using a linear term for time:

```
R> model_cb <- fitSmoothHazard(
R>   Status ~ ftime + Sex + D + Phase + Source + Age,
R>   data = bmtcrr,
R>   ratio = 100,
R>   time = "ftime"
R> )
```

We will compare our hazard ratio estimates to that obtained from a Cox regression. To do so, we need to treat the competing event as censoring.

```
R> library(survival)
R> # Treat competing event as censoring
R> model_cox <- coxph(Surv(ftime, Status == 1) ~ Sex + D + Phase + Source + Age,
R>   data = bmtcrr
R> )
```

From the fit object, we can extract both the hazard ratios and their corresponding confidence intervals. These quantities appear in Table **??**. As we can see, the only significant hazard ratio identified by case-base sampling is the one associated with the phase of the disease at transplant. More precisely, being in relapse at transplant is associated with a hazard ratio of 3.89 when compared to CR1.

Given the estimate of the hazard function obtained using case-base sampling, we can compute the absolute risk curve for a fixed covariate profile. We perform this computation for a 35 year old woman who received a stem-cell transplant from peripheral blood at relapse. We compared the absolute risk curve for such a woman with acute lymphoblastic leukemia with that for a similar woman with acute myeloblastic leukemia. We will estimate the curve from 0 to 60 months.

```
R> # Pick 100 equidistant points between 0 and 60 months
R> time_points <- seq(0, 60, length.out = 50)
R>
R> # Data.frame containing risk profile
R> newdata <- data.frame(
R>   "Sex" = factor(c("F", "F"),
R>     levels = levels(bmtcrr[, "Sex"])
R>   ),
R>   "D" = c("ALL", "AML"), # Both diseases
R>   "Phase" = factor(c("Relapse", "Relapse"),
R>     levels = levels(bmtcrr[, "Phase"])
R>   ),
R>   "Age" = c(35, 35),
R>   "Source" = factor(c("PB", "PB"),
R>     levels = levels(bmtcrr[, "Source"])
R>   )
R> )
R>
R> # Estimate absolute risk curve
R> risk_cb <- absoluteRisk(
R>   object = model_cb, time = time_points,
R>   method = "numerical", newdata = newdata
R> )
```

We will compare our estimates to that obtained from a corresponding Fine-Gray model (1999). The Fine-Gray model is a semiparametric model for the cause-specific *subdistribution*, i.e. the function $f_k(t)$ such that

$$CI_k(t) = 1 - \exp\left(-\int_0^t f_k(u)\mathrm{d}u\right),$$

where $CI_k(t)$ is the cause-specific cumulative indicence. The Fine-Gray model allows to directly assess the effect of a covariate on the subdistribution, as opposed to the hazard. For the computation, we will use the **timereg** package (Scheike and Zhang 2011):

```
R> library(timereg)
R> model_fg <- comp.risk(Event(ftime, Status) ~ const(Sex) + const(D) +
R>   const(Phase) + const(Source) + const(Age),
R> data = bmtcrr, cause = 1, model = "fg"
R> )
R>
R> # Estimate absolute risk curve
R> risk_fg <- predict(model_fg, newdata, times = time_points)
```

Figure **??** shows the absolute risk curves for both case-base sampling and the Fine-Gray model. As we can see, the two approaches agree quite well for acute myeloblastic leukemia; however, there seems to be a difference of about 5% between the two curves for acute lymphoblastic leukemia. This difference does not appear to be significant: the curve from case-base sampling is contained within a 95% confidence band around the Fine-Gray absolute risk curve (figure not shown).

7. Case study 3–SUPPORT Data

In the first two case studies, we described the basic functionalities of the **casebase** package: creating population-time plots, fitting parametric models for hazard functions, and estimating the corresponding cumulative incidence curves. In the next two case studies, we explore more advanced features.

For the third case study, we will look at regularized estimation for the hazard function. This can be performed using a combination of case-base sampling and regularized logistic regression. To demonstrate this capability, we examine the Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT) dataset (Knaus, Harrell, Lynn, Goldman, Phillips, Connors, Dawson, Fulkerson, Califf, Desbiens *et al.* 1995). The SUPPORT dataset tracks death for individuals who are considered seriously ill within 5 hospitals in the United States of America. Imputation was conducted using the **mice** package in R with default settings (van Buuren and Groothuis-Oudshoorn 2011). Before imputation, there were a total of 9105 individuals and 47 variables. Of those variables, a few were removed before imputation. We removed three response variables: hospital charges, patient ratio of costs to charges and patient micro-costs. Next, we removed hospital death as it was directly informative of our event of interest, death. We also removed functional disability and income as they are ordinal covariates, because ordinal variables are treated as factors in our current model. Finally, we removed 8 covariates relating to the results of previous findings: we removed SUPPORT day 3 physiology score (sps), APACHE III day 3 physiology score (aps), SUPPORT model 2-month survival estimate, SUPPORT model 6-month survival estimate, Physician's 2-month survival estimate for pt., Physician's 6-month survival estimate for pt., Patient had Do Not Resuscitate (DNR) order and Day of DNR order (<0 if before study). Of these, sps and aps were added on after imputation, as they were missing only 1 observation and permitted us to compare findings between our model and their's.

First we impute manually using recommended normal values for physiological measures (Harrell 2020). Next, we impute a single dataset using **mice** with default settings. After imputation, we noted that the covariate for surrogate activities of daily living was not imputed. This is due to colinearity between the other two covariates for activities of daily living. As such, surrogate activites of daily living was removed from our model. After these imputations, we have a total of 9104 observations and 30 variables, two of which are sps and aps, which

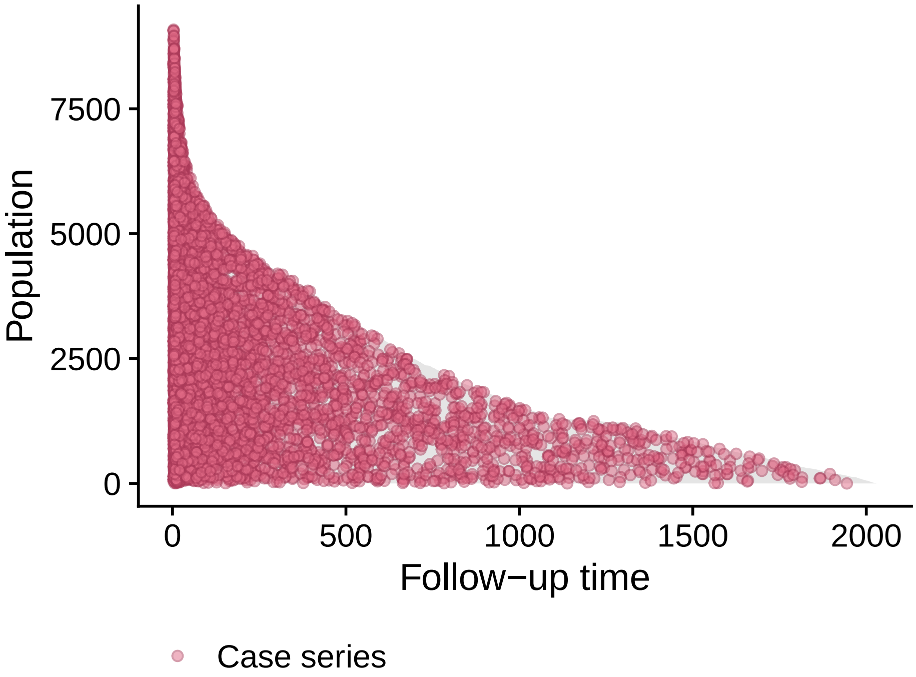

Figure 1: Population-time plot for the SUPPORT data.

will work as a baseline for our comparison to previous findings. This dataset is available Its worth noting that these scores were developed for a period of 180 days, which makes our comparison over 5.56 years exploratory. Also, our single imputation introduces bias in the data. As the goal here is purely demonstrative, this bias will be consistent with any of the following models. As such, their comparison is still valid, though any interpretation requires an awareness of said bias. A description of used variables can be found in Table 3 and a breakdown of each categorical variable in Table 4. The original data was obtained from the Department of Biostatistics at Vanderbilt University (Harrell 2020). The imputed dataset is available as part of the **casebase** package.

As before, the first step is to visualize the incidence density. The population-time plot for the SUPPORT data appears in Figure 1. As we can see, the incidence density is not constant.

Before generating an absolute risk curve using regularized logistic regression, we will explore three options as a baseline comparison, beyond a Kaplan-meier curve. The original study compared these scores over a span of 180 days, which may not be comparable over the full 5.56 years. Therefore, We will examine three cox regression scenarios: sps only, aps only and all covariates excluding sps and aps. Next, we will compare each modeling using the Brier score (Graf, Schmoor, Sauerbrei, and Schumacher 1999). The Brier score is able to demonstrate both discrimination and calibration in a single model (Graf *et al.* 1999) and provide a comparable metric between semi-parametric and parametric methods in the survival context. For all the following analyses in this case study, we will use 80% of the observations to train our models, while 20% will be used to generate absolute risk curves, all of which will be averaged to demonstrate each model's predictive ability. Over 5.56 years, 2 demonstrates that all the cox models have similar, if not identical average predictions. Figure 3 examines

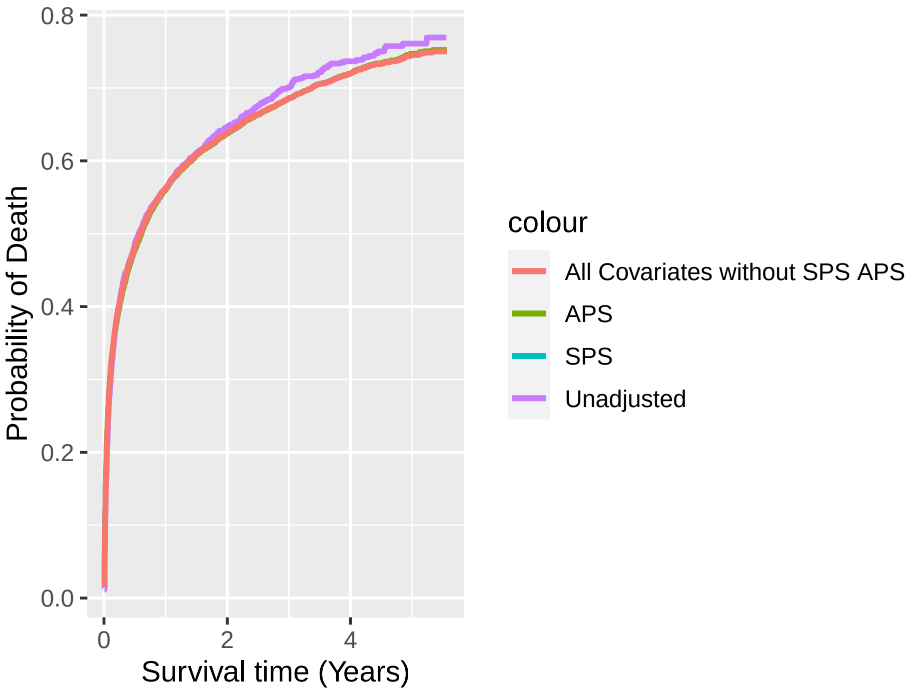

Figure 2: AbsoluteRisk curves from hazard models. Pink: All covariates without SPS nor APS. Green: APS only model. Blue: SPS only model. Purple: unadjusted curve. All three models underestimate the risk in comparison to the unadjusted curve.

the brier scores as they change over time. As the full model has the lowest brier score on overall, it will be used in the following comparison to regularized cox, regularized logistic regression and the unadjusted Kaplan-Meier curve.

To get the absolute risk following regularized cox regression, this required that we work around limitations in existing packages. To make use of the **survival** package, we first get our hazard coefficients using **glmnet**'s coxnet implementation for lasso penalization (Simon *et al.* 2011). Next, we fit a cox model using the survival package with the selected variables only, from our previous fit. Finally, we replaced the coefficients in the survival object with those from our glmnet fit, allowing us to make use of the absolute risk tools available in the **survival** package.

For our penalized logistic regression model, we opted for a logistic function of time, resulting in a Weibull distribution assumption. As casebase makes use of the glmnet package, we interact with the fitsmoothhazard.fit function using a matrix interface, where y contains the time and event variables and x contains all other variables we would like to include in the model. For the SUPPORT dataset, all factor variables were converted into dummy variables. Here, we used lasso penalization by setting alpha to 1, leading to our smooth hazard function. We then take our hazard and use it to predict the absolute risk of each individual in our test set, over time. The probabilites over time for each individual are averaged, resulting in the smooth absolute risk curve in figure 4. On average, the Brier score is lower in our penalized logistic regression model, compared to the other models (fig. 6).

This case study serves to demonstrate how regularization can be used in a survival context, using logistic regression. Our approach performs comparably to Cox regression, demonstrating

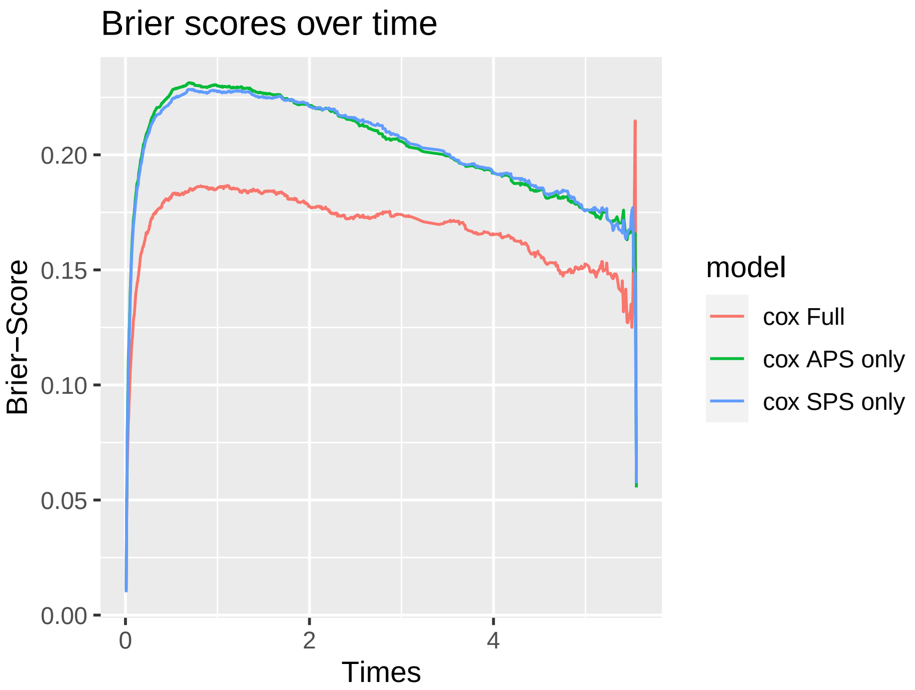

Figure 3: Brier scores over time. Full model (Cox Full) performs best, while the other two (APS only and SPS only) are comparable.

its validity in an applied setting.

```
R>
R> cbFull <- casebase::fitSmoothHazard.fit(x, y, family = "glmnet", time = "d.time", event
```

```
R> # Estimating the absolute risk curve using the newData parameter.
R> abCbFull <- casebase::absoluteRisk(cbFull, time = times, newdata = newData, s = "lambda
```

8. Case study 4–Stanford Heart Transplant Data

In the previous case studies, we only considered covariates that were fixed at baseline. In this next case study, we will use the Stanford Heart Transplant data (Clark, Stinson, Griepp, Schroeder, Shumway, and Harrison 1971, Crowley and Hu (1977)) to show how case-base sampling can also be used in the context of time-varying covariates. As an example that already appeared in the literature, case-base sampling was used to study vaccination safety, where the exposure period was defined as the week following vaccination (Saarela and Hanley 2015). Hence, the main covariate of interest, i.e. exposure to the vaccine, was changing over time. In this context, case-base sampling offers an efficient alternative to nested case-control designs or self-matching.

Recall the setting of Stanford Heart Transplant study: patients were admitted to the Stanford program after meeting with their physician and determining that they were unlikely to respond

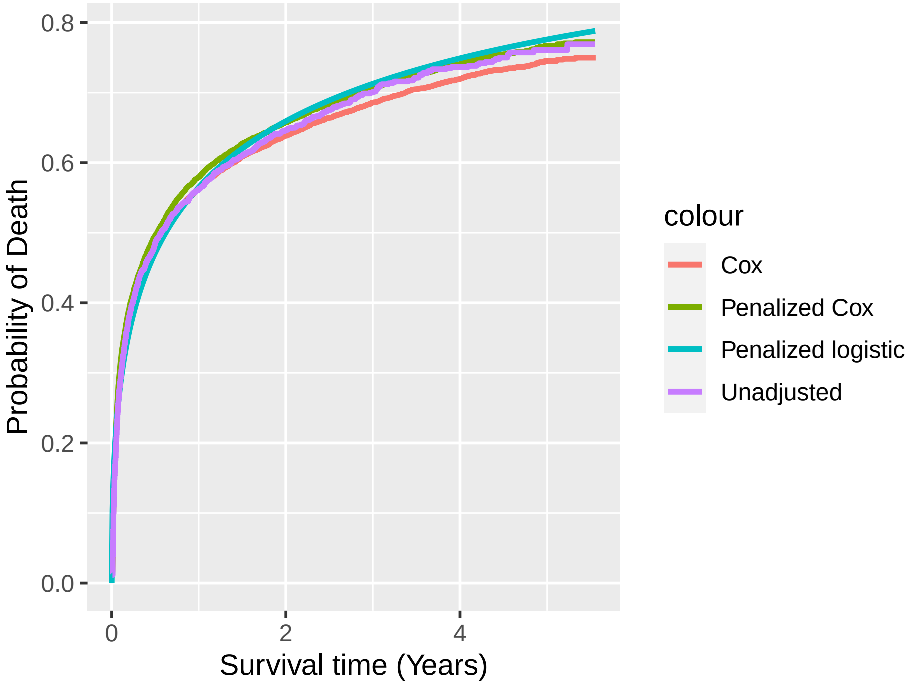

Figure 4: Four absolute risk curves, integrated from the following hazard models: Cox (pink), penalized Cox (green), penalized logistic (blue) and Kaplan-Meier (purple).

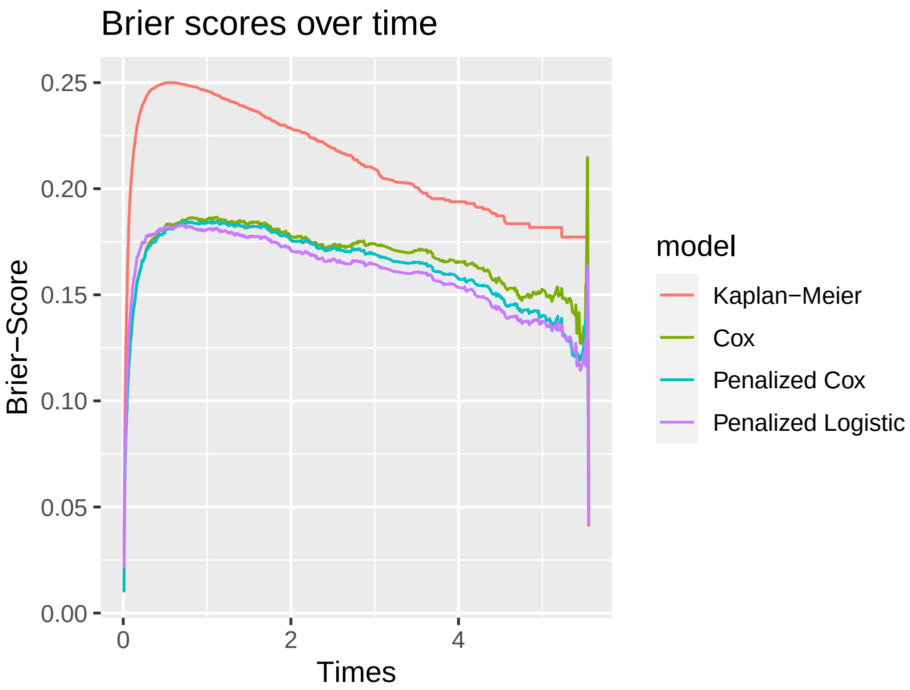

Figure 5: 4 curves comparing the brier scores as they change over time between a null model (Kaplan-Meier, black), Cox regression (yellow), penalized Cox regression (blue) and penalized logistic regression (green).

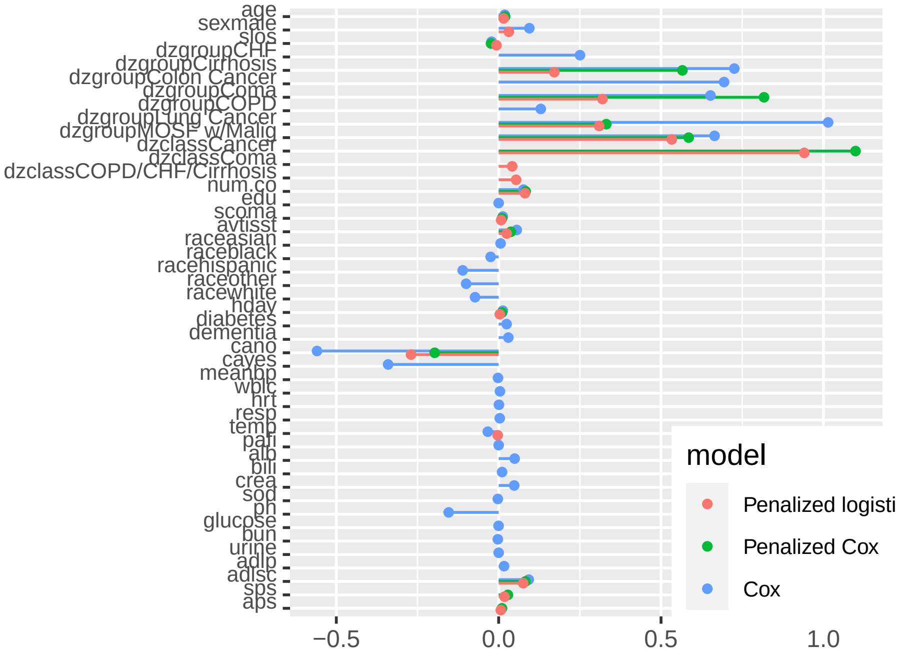

Figure 6: A lollipop plot demonstrating the estimated coefficients between the following three models: Cox (blue), penalized Cox (Green), penalized logisitic (red). If a coefficient is estimated to be 0, it is removed from the plot

to other forms of treatment. After enrollment, the program searched for a suitable donor for the patient, which could take anywhere between a few days to almost a year. We are interested in the effect of a heart transplant on survival; therefore, the patient is considered exposed only after the transplant has occurred.

As before, we can look at the population-time plot for a graphical summary of the event incidence. As we can see, most events occur early during the follow-up period, and therefore we do not expect the hazard to be constant.

```
R> library(survival)
R> library(casebase)
R>
R> stanford_popTime <- popTime(jasa,
R>   time = "futime",
R>   event = "fustat"
R> )
R> plot(stanford_popTime)
```

Since the exposure is time-dependent, we need to manually define the exposure variable *after* case-base sampling and *before* fitting the hazard function. For this reason, we will use the `sampleCaseBase` function directly.

```
R> library(tidyverse)
R> library(lubridate)
R>
R> cb_data <- sampleCaseBase(jasa,
R>   time = "futime",
R>   event = "fustat", ratio = 10
R> )
```

Next, we will compute the number of days from acceptance into the program to transplant, and we use this variable to determine whether each population-moment is exposed or not.

```
R> # Define exposure variable
R> cb_data <- mutate(cb_data,
R>   txtime = time_length(accept.dt %--% tx.date,
R>     unit = "days"
R>   ),
R>   exposure = case_when(
R>     is.na(txtime) ~ 0L,
R>     txtime > futime ~ 0L,
R>     txtime <= futime ~ 1L
R>   )
R> )
```

Finally, we can fit the hazard using various linear predictors.

```
R> library(splines)
R> # Fit several models
R> fit1 <- fitSmoothHazard(fustat ~ exposure,
R>   data = cb_data, time = "futime"
R> )
R> fit2 <- fitSmoothHazard(fustat ~ exposure + futime,
R>   data = cb_data, time = "futime"
R> )
R> fit3 <- fitSmoothHazard(fustat ~ exposure + bs(futime),
R>   data = cb_data, time = "futime"
R> )
R> fit4 <- fitSmoothHazard(fustat ~ exposure * bs(futime),
R>   data = cb_data, time = "futime"
R> )
```

Note that the fourth model (i.e. `fit4`) includes an interaction term between exposure and follow-up time. In other words, this model no longer exhibit proportional hazards. The evidence of non-proportionality of hazards in the Stanford Heart Transplant data has been widely discussed (Arjas 1988).

We can then compare the goodness of fit of these four models using the Akaike Information Criterion (AIC).

```
R> # Compute AIC
R> c(
R>   "Model1" = AIC(fit1),
R>   "Model2" = AIC(fit2),
R>   "Model3" = AIC(fit3),
R>   "Model4" = AIC(fit4)
R> )
```

As we can, the best fit is the fourth model. By visualizing the hazard functions for both exposed and unexposed individuals, we can more clearly see how the hazards are no longer proportional.

```
R> # Compute hazards---
R> # First, create a list of time points for both exposure status
R> hazard_data <- expand.grid(
R>   exposure = c(0, 1),
R>   futime = seq(0, 1000,
R>     length.out = 100
R>   )
R> )
R> # Set the offset to zero
R> hazard_data$offset <- 0
R> # Use predict to get the fitted values, and exponentiate to
R> # transform to the right scale
```

```
R> hazard_data$hazard <- exp(predict(fit4,
R>   newdata = hazard_data,
R>   type = "link"
R> ))
R> # Add labels for plots
R> hazard_data$Status <- factor(hazard_data$exposure,
R>   labels = c("NoTrans", "Trans")
R> )
R>
R> ggplot(hazard_data, aes(futime, hazard, colour = Status)) +
R>   geom_line() +
R>   theme_minimal() +
R>   theme(legend.position = "top") +
R>   ylab("Hazard") +
R>   xlab("Follow-up time")
```

The non-proportionality seems to be more pronounced at the beginning of follow-up than the end. Finally, we can turn these estimates of the hazard function into estimates of the cumulative incidence functions.

```
R> # Compute absolute risk curves
R> newdata <- data.frame(exposure = c(0, 1))
R> absrisk <- absoluteRisk(fit4,
R>   newdata = newdata,
R>   time = seq(0, 1000, length.out = 100)
R> )
R>
R> colnames(absrisk) <- c("Time", "NoTrans", "Trans")
R>
R> # Rearrange the data
R> absrisk <- gather(
R>   as.data.frame(absrisk),
R>   "Status", "Risk", -Time
R> )
R>
R> ggplot(absrisk, aes(Time, Risk, colour = Status)) +
R>   geom_line() +
R>   theme_minimal() +
R>   theme(legend.position = "top") +
R>   expand_limits(y = 0.5) +
R>   xlab("Follow-up time") +
R>   ylab("Cum. Incidence")
```

Note that we can easily adapt the code above to the situation where a patient receives a heart transplant at a point in time of interest, for example after 30 days.

```
R> # Compute hazards---
R> # First, create a list of time points for both exposure status
```

```
R> one_yr_haz <- data.frame(futime = seq(0, 365,
R>   length.out = 100
R> ))
R> one_yr_haz <- mutate(one_yr_haz,
R>   offset = 0,
R>   exposure = if_else(futime < 30, 0, 1)
R> )
R> one_yr_haz$hazard <- exp(predict(fit4,
R>   newdata = one_yr_haz,
R>   type = "link"
R> ))
R> ggplot(one_yr_haz, aes(futime, hazard)) +
R>   geom_line() +
R>   theme_minimal() +
R>   theme(legend.position = "top") +
R>   ylab("Hazard") +
R>   xlab("Follow-up time") +
R>   geom_vline(xintercept = 30, linetype = "dashed")
```

We can then compare the 1-year mortality risk without transplant and with transplant at 30 days.

```
R> absoluteRisk(fit4,
R>   newdata = data.frame(exposure = 0),
R>   time = 365
R> )
R>
R> # Use the trapezoidal rule to estimate the cumulative hazard
R> cumhaz_est <- pracma::trapz(one_yr_haz$futime, one_yr_haz$hazard)
R> 1 - exp(-cumhaz_est)
```

As we can see, the risk estimate at 1-year is about 30% lower if the patient receives a heart transplant at 30 days.

9. Discussion

In this article, we presented the R package **casebase**, which provides functions for fitting smooth parametric hazards and estimating cumulative incidence functions using case-base sampling. We outlined the theoretical underpinnings of the approach, we provided details about our implementation, and we illustrated the merits of the approach and the package through four case studies.

As a methodological framework, case-base sampling is very flexible. This flexibility has been explored before in the literature: for example, Saarela and Hanley (2015) used case-base sampling to model a time-dependent exposure variable in a vaccine safety study. As another example, Saarela and Arjas (2015) combined case-base sampling and a Bayesian non-parametric framework to compute individualized risk assessments for chronic diseases. In the case studies above, we explored this flexibility along two fronts. On the one hand, we showed how splines could be used as part of the linear predictor to model the effect of time on the hazard. This strategy yielded estimates of the survival function that were qualitatively similar to semiparametric estimates derived from Cox regression; however, case-base sampling led to estimates of the survival function that *vary smoothly in time*. On the other hand, we also displayed the flexibility of case-base sampling by showing how it could be combined with penalized logistic regression to perform variable selection. Even though we did not illustrate it in this article, case-base sampling can also be combined with the framework of *generalized additive models*. This functionality has been implemented in our package **casebase**. Similarly, case-base sampling can also be combined with quasi-likelihood estimation to fit survival models that can account for the presence of over-dispersion. All of these examples illustrate how the case-base sampling framework in general, and the package **casebase** in particular, allows the user to fit a broad and flexible family of survival functions.

As presented in Hanley & Miettinen (2009), case-base sampling is comprised of three steps: 1) sampling a case series and a base series from the study; 2) fit the log-hazard as a linear function of predictors (including time); and 3) use the fitted hazard to estimate the cumulative incidence curve. Accordingly, our package provides functions for each step. Moreover, the simple interface of the `fittingSmoothHazard` function resembles the `glm` interface. This interface should look familiar to new users. Our modular approach also provides a convenient way to extend our package for new sampling or fitting strategies.

In the case studies above, we compared the performance of case-base sampling with that of Cox regression and Fine-Gray models. In terms of function interface, **casebase** uses a formula interface that is closer to that of `glm`, in that the event variable is the only variable appearing on the left-hand side of the formula. By contrast, both `survival::coxph` and `timereg::comp.risk` use arrays that captures both the event type and time. Both approaches to modeling yield user-friendly code. However, in terms of output, both approaches differ significantly. Case-base sampling produces smooth hazards and smooth cumulative incidence curves, whereas Cox regression and Fine-Gray models produce step-wise cumulative incidence functions and never explicitly model the hazard function. Qualitatively, we showed that by using splines in the linear predictor, all three models yielded similar curves. However, the smooth nature of the output of **casebase** provides a more intuitive interpretation for consumers of these predictions. In Table 5, we provide a side-by-side comparison between the Cox model and case-base sampling.

Our choice of modeling the log-hazard as a linear function of covariates allows us to develop a simple computational scheme for estimation. However, as a downside, it does not allow us to model location and scale parameters separately like the package **flexsurv**. For example, if we look at the Weibull distribution as parametrised in `stats::pweibull`, the log-hazard function is given by

$$\log \lambda(t;\alpha,\beta) = [\log(\alpha/\beta) - (\alpha-1)\log(\beta)] + (\alpha-1)\log t,$$

where α, β are shape and scale parameters, respectively. Unlike **casebase**, the approach taken by **flexsurv** allows the user to model the scale parameter as a function of covariates. Of course, this added flexibility comes at the cost of interpretability: by modeling the log-hazard directly, the parameter estimates from **casebase** can be interpreted as estimates of log-hazard ratios. To improve the flexibility of **casebase** at capturing the scale of a parametric family, we could replace the logistic regression with its quasi-likelihood counterpart and therefore model over- and under-dispersion with respect to the logistic likelihood. We defer the study of the properties and performance of such a model to a future article.

Future work will look at some of the methodological extensions of case-base sampling. First, to assess the quality of the model fit, we want to study the properties of the residuals (e.g. Cox-Snell, martingale). More work needs to be done to understand these residuals in the context of the partial likelihood underlying case-base sampling. The resulting diagnostic tools would then be integrated in this package. Also, we are interested in extending case-base sampling to account for interval censoring. This type of censoring is very common in longitudinal studies, and many packages (e.g. **SmoothHazard**, **survival** and **Rstpm2**) provide functions to account for it. Again, we hope to include any resulting methodology as part of this package.

In future versions of the package, we also want to increase the complement of diagnostic and inferential tools that are currently available. For example, we would like to include the ability to compute confidence intervals for the cumulative incidence curve. The delta method or parametric bootstrap are two different strategies we can use to construct approximate confidence intervals. Furthermore, we would like to include functions to compute calibration and discrimination statistics (e.g. Brier score, AUC) for our models. Saarela and Arjas (2015) also describe how to obtain a posterior distribution for the AUC from their model. Their approach could also be included in **casebase**. Finally, we want to provide more flexibility in how the case-base sampling is performed. This could be achieved by adding a `hazard` argument to the function `sampleCaseBase`. In this way, users could specify their own sampling mechanism. For example, they could provide a hazard that gives sampling probabilities that are proportional to the cardiovascular disease event rate given by the Framingham score (Saarela and Arjas 2015).

In conclusion, we presented the R package **casebase** which implements case-base sampling for fitting parametric survival models and for estimating smooth cumulative incidence functions using the framework of generalized linear models. We strongly believe that its flexibility and its foundation on the familiar logistic regression model will make it appealing to new and established practioners.

10. Environment Details

This report was generated on 2020-04-09 13:28:46 using the following computational environment and dependencies:

```
R> # which R packages and versions?
R> # devtools::session_info()
R> sessionInfo()

#> R version 3.6.3 (2020-02-29)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 10 x64 (build 18363)
#>
#> Matrix products: default
#>
#> locale:
#> [1] LC_COLLATE=English_United States.1252
#> [2] LC_CTYPE=English_United States.1252
#> [3] LC_MONETARY=English_United States.1252
#> [4] LC_NUMERIC=C
#> [5] LC_TIME=English_United States.1252
#>
#> attached base packages:
#> [1] splines   stats     graphics  grDevices utils     datasets  methods
#> [8] base
#>
#> other attached packages:
#>  [1] broom_0.5.5               dotwhisker_0.5.0
#>  [3] prodlim_2019.11.13        riskRegression_2020.03.27
#>  [5] pracma_2.2.9              kableExtra_1.1.0
#>  [7] data.table_1.12.8         glmnet_3.0-2
#>  [9] Matrix_1.2-18             casebase_0.2.1.9001
#> [11] survival_3.1-11           magrittr_1.5
#> [13] forcats_0.5.0             stringr_1.4.0
#> [15] dplyr_0.8.5               purrr_0.3.3
#> [17] readr_1.3.1               tidyr_1.0.2
#> [19] tibble_3.0.0              ggplot2_3.3.0
#> [21] tidyverse_1.3.0
#>
#> loaded via a namespace (and not attached):
#>  [1] TH.data_1.0-10      VGAM_1.1-2          colorspace_1.4-1
#>  [4] ellipsis_0.3.0      ggstance_0.3.4      htmlTable_1.13.3
#>  [7] base64enc_0.1-3     fs_1.4.1            rstudioapi_0.11
#> [10] farver_2.0.3        MatrixModels_0.4-1  fansi_0.4.1
#> [13] mvtnorm_1.1-0       lubridate_1.7.8     xml2_1.3.1
#> [16] codetools_0.2-16    knitr_1.28          Formula_1.2-3
#> [19] jsonlite_1.6.1      cluster_2.1.0       dbplyr_1.4.2
#> [22] png_0.1-7           compiler_3.6.3      httr_1.4.1
#> [25] backports_1.1.6     assertthat_0.2.1    cli_2.0.2
#> [28] acepack_1.4.1       htmltools_0.4.0     quantreg_5.55
#> [31] tools_3.6.3         gtable_0.3.0        glue_1.4.0
#> [34] Rcpp_1.0.4.6        cellranger_1.1.0    vctrs_0.2.4
#> [37] nlme_3.1-145        iterators_1.0.12    xfun_0.12
#> [40] rvest_0.3.5         lifecycle_0.2.0     polspline_1.1.17
#> [43] MASS_7.3-51.5       zoo_1.8-7           scales_1.1.0
#> [46] hms_0.5.3           sandwich_2.5-1      SparseM_1.78
#> [49] RColorBrewer_1.1-2  rticles_0.14        yaml_2.2.1
#> [52] gridExtra_2.3       rms_5.1-4           rpart_4.1-15
#> [55] latticeExtra_0.6-29 stringi_1.4.6       foreach_1.5.0
#> [58] checkmate_2.0.0     lava_1.6.7          shape_1.4.4
#> [61] mets_1.2.7.1        rlang_0.4.5         pkgconfig_2.0.3
#> [64] evaluate_0.14       lattice_0.20-41     htmlwidgets_1.5.1
#> [67] labeling_0.3        cmprsk_2.2-9        tidyselect_1.0.0
#> [70] plyr_1.8.6          R6_2.4.1            generics_0.0.2
#> [73] Hmisc_4.4-0         multcomp_1.4-13     DBI_1.1.0
#> [76] pillar_1.4.3        haven_2.2.0         foreign_0.8-76
#> [79] withr_2.1.2         mgcv_1.8-31         nnet_7.3-13
#> [82] modelr_0.1.6        crayon_1.3.4        rmarkdown_2.1
#> [85] timereg_1.9.4       jpeg_0.1-8.1        grid_3.6.3
#> [88] readxl_1.3.1        reprex_0.3.0        digest_0.6.25
#> [91] webshot_0.5.2       numDeriv_2016.8-1.1 stats4_3.6.3
#> [94] munsell_0.5.0       viridisLite_0.3.0
```

The current Git commit details are:

```
R> # what commit is this file at?
R> git2r::repository(here::here())
```

```
#> Local:    master C:/Users/jesse/Documents/gits/cbpaper
#> Remote:   master @ origin (https://github.com/Jesse-Islam/cbpaper.git)
#> Head:     [b31e195] 2020-03-31: Fixed penal. on time and Brier
```

References

Allignol A, Latouche A (2019). "CRAN Task View: Survival Analysis." URL https://cran.r-project.org/web/views/Survival.html.

Arjas E (1988). "A graphical method for assessing goodness of fit in Cox's proportional hazards model." *Journal of the American Statistical Association*, **83**(401), 204–212.

Breslow N (1972). "Discussion of the paper by DR Cox cited below." *Journal of the Royal Statistical Society, Series B*, **34**, 187–220.

Clark DA, Stinson EB, Griepp RB, Schroeder JS, Shumway NE, Harrison D (1971). "Cardiac transplantation in man." *Annals of Internal Medicine*, **75**(1), 15–21.

Clements M, Liu XR, Lambert P, Jakobsen LH, Gasparini A, Smyth G, Alken P, Wood S, Ulerich R (2019). "Smooth Survival Models, Including Generalized Survival Models [R package rstpm2 version 1.5.1]." URL https://cran.r-project.org/web/packages/rstpm2/index.html.

Clerc-Urmès I, Grzebyk M, Hédelin G (2017). *flexrsurv: An R package for relative survival analysis.* R package version 1.4.1, URL https://CRAN.R-project.org/package=flexrsurv.

Cox DR (1972). "Regression models and life-tables." *Journal of the Royal Statistical Society: Series B (Methodological)*, **34**(2), 187–202.

Crowley J, Hu M (1977). "Covariance analysis of heart transplant survival data." *Journal of the American Statistical Association*, **72**(357), 27–36.

Fine JP, Gray RJ (1999). "A proportional hazards model for the subdistribution of a competing risk." *Journal of the American statistical association*, **94**(446), 496–509.

Friedman J, Hastie T, Tibshirani R (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software*, **33**(1). ISSN 1548-7660. doi:10.18637/jss.v033.i01.

Fu Z (2015). "Package crrp." URL https://cran.r-project.org/web/packages/crrp/index.html.

Gerds TA, Blanche P, Mortersen R, Tollenaar N, Mogensen UB, Ozenne B (2019). "Risk Regression Models and Prediction Scores for Survival Analysis with Competing Risks [R package riskRegression version 2019.11.03]." URL https://CRAN.R-project.org/package=riskRegression.

Goeman J, Meijer R, Chaturvedi N, Lueder M (2019). "Package penalized." URL https://cran.r-project.org/web/packages/penalized/index.html.

Goeman JJ (2010). "L1 penalized estimation in the Cox proportional hazards model." *Biometrical Journal*, (52), –14.

Graf E, Schmoor C, Sauerbrei W, Schumacher M (1999). "Assessment and comparison of prognostic classification schemes for survival data." *Statistics in medicine*, **18**(17-18), 2529–2545.

Gray B (2019). "Package cmprsk." URL https://cran.r-project.org/web/packages/cmprsk/index.html.

Hanley JA, Miettinen OS (2009). "Fitting smooth-in-time prognostic risk functions via logistic regression." *The International Journal of Biostatistics*, **5**(1).

Harrell F (2020). "SupportDesc < Main < Vanderbilt Biostatistics Wiki." URL http://biostat.mc.vanderbilt.edu/wiki/Main/SupportDesc.

Hastie T, Tibshirani R (1987). "Generalized additive models: some applications." *Journal of the American Statistical Association*, **82**(398), 371–386.

Jackson C (2016). "flexsurv: A Platform for Parametric Survival Modeling in R." *Journal of Statistical Software*, **70**(8), 1–33. doi:10.18637/jss.v070.i08.

Kalbfleisch JD, Prentice RL (2011). *The statistical analysis of failure time data*, volume 360. John Wiley & Sons.

Knaus WA, Harrell FE, Lynn J, Goldman L, Phillips RS, Connors AF, Dawson NV, Fulkerson WJ, Califf RM, Desbiens N, *et al.* (1995). "The SUPPORT prognostic model: objective estimates of survival for seriously ill hospitalized adults." *Annals of internal medicine*, **122**(3), 191–203.

Liu Z, Rich B, Hanley JA (2014). "Recovering the raw data behind a non-parametric survival curve." *Systematic reviews*, **3**(1), 151.

Mahani A, Sharabiani M (2019). "Bayesian, and Non-Bayesian, Cause-Specific Competing-Risk Analysis for Parametric and Nonparametric Survival Functions: The R Package CFC." *Journal of Statistical Software, Articles*, **89**(9), 1–29. ISSN 1548-7660. doi:10.18637/jss.v089.i09. URL https://www.jstatsoft.org/v089/i09.

Park MY, Hastie T (2018). "Package glmpath." URL https://CRAN.R-project.org/package=glmpath.

Perperoglou A (2015). "Package CoxRidge." URL https://CRAN.R-project.org/package=CoxRidge.

Saarela O (2016). "A case-base sampling method for estimating recurrent event intensities." *Lifetime data analysis*, **22**(4), 589–605.

Saarela O, Arjas E (2015). "Non-parametric Bayesian Hazard Regression for Chronic Disease Risk Assessment." *Scandinavian Journal of Statistics*, **42**(2), 609–626.

Saarela O, Hanley JA (2015). "Case-base methods for studying vaccination safety." *Biometrics*, **71**(1), 42–52.

Scheike TH, Holst KK, Hjelmborg JB (2014). "Estimating twin concordance for bivariate competing risks twin data." *Statistics in medicine*, **33**(7), 1193–1204.

Scheike TH, Zhang MJ (2011). "Analyzing Competing Risk Data Using the R timereg Package." *Journal of Statistical Software*, **38**(2), 1–15. URL http://www.jstatsoft.org/v38/i02/.

Schröder FH, Hugosson J, Roobol MJ, Tammela TL, Ciatto S, Nelen V, Kwiatkowski M, Lujan M, Lilja H, Zappa M, *et al.* (2009). "Screening and prostate-cancer mortality in a randomized European study." *New England Journal of Medicine*, **360**(13), 1320–1328.

Scrucca L, Santucci A, Aversa F (2010). "Regression modeling of competing risk using R: an in depth guide for clinicians." *Bone marrow transplantation*, **45**(9), 1388.

Simon N, Friedman J, Hastie T, Tibshirani R (2011). "Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent." *Journal of Statistical Software*, **39**(5), 1–13. URL http://www.jstatsoft.org/v39/i05/.

Therneau TM (2015). *A Package for Survival Analysis in S*. Version 2.38, URL https://CRAN.R-project.org/package=survival.

Touraine C, Gerds TA, Joly P (2017). "SmoothHazard: An R Package for Fitting Regression Models to Interval-Censored Observations of Illness-Death Models." *Journal of Statistical Software*, **79**(7), 1–22. doi:10.18637/jss.v079.i07.

van Buuren S, Groothuis-Oudshoorn K (2011). "mice: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software*, **45**(3), 1–67. URL https://www.jstatsoft.org/v45/i03/.

Yi Y, Zou H (2017). "Package fastcox." URL https://cran.r-project.org/web/packages/fastcox/index.html.

Affiliation:

Sahir Bhatnagar *
McGill University
1020 Pine Avenue West Montreal, QC, Canada H3A 1A2
E-mail: sahir.bhatnagar@mail.mcgill.ca
URL: http://sahirbhatnagar.com/

Maxime Turgeon *
University of Manitoba
186 Dysart Road Winnipeg, MB, Canada R3T 2N2
E-mail: max.turgeon@umanitoba.ca
URL: https://maxturgeon.ca/

Jesse Islam
McGill University
1020 Pine Avenue West Montreal, QC, Canada H3A 1A2
E-mail: jesse.islam@mail.mcgill.ca

James Hanley
McGill University
1020 Pine Avenue West Montreal, QC, Canada H3A 1A2
E-mail: james.hanley@mcgill.ca
URL: http://www.medicine.mcgill.ca/epidemiology/hanley/

Olli Saarela
University of Toronto
Dalla Lana School of Public Health, 155 College Street, 6th floor, Toronto, Ontario M5T 3M7, Canada
E-mail: olli.saarela@utoronto.ca
URL: http://individual.utoronto.ca/osaarela/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd

Name	Labels	Levels	Storage	NAs
age	Age		double	0
death	Death at any time up to NDI date:31DEC94		double	0
sex		2	integer	0
slos	Days from Study Entry to Discharge		double	0
d.time	Days of Follow-Up		double	0
dzgroup		8	integer	0
dzclass		4	integer	0
num.co	number of comorbidities		double	0
edu	Years of Education		double	202
income		4	integer	349
scoma	SUPPORT Coma Score based on Glasgow D3		double	0
avtisst	Average TISS, Days 3-25		double	6
race		5	integer	5
sps	support physiology score day 3		double	1
aps	APS III no coma, imp bun,uout for ph1,D3		double	1
hday	Day in Hospital at Study Admit		double	0
diabetes	Diabetes (Com 27-28, Dx 73)		double	0
dementia	Dementia (Comorbidity 6)		double	0
ca	Cancer State	3	integer	0
meanbp	Mean Arterial Blood Pressure Day 3		double	0
wblc	White Blood Cell Count Day 3		double	24
hrt	Heart Rate Day 3		double	0
resp	Respiration Rate Day 3		double	0
temp	Temperature (Celsius) Day 3		double	0
pafi	PaO2/(.01*FiO2) Day 3		double	253
alb	Serum Albumin Day 3		double	378
bili	Bilirubin Day 3		double	297
crea	Serum creatinine Day 3		double	3
sod	Serum sodium Day 3		double	0
ph	Serum pH (arterial) Day 3		double	250
glucose	Glucose Day 3		double	470
bun	BUN Day 3		double	455
urine	Urine Output Day 3		double	517
adlp	Activiites of Daily Living (ADL) Patient Day 3		double	634
adlsc	Imputed ADL Calibrated to Surrogate		double	0

Table 3: A description of each variable in the SUPPORT dataset.

Variable	**Levels**
sex	female
	male
dzgroup	ARF/MOSF w/Sepsis
	COPD
	CHF
	Cirrhosis
	Coma
	Colon Cancer
	Lung Cancer
	MOSF w/Malig
dzclass	ARF/MOSF
	COPD/CHF/Cirrhosis
	Coma
	Cancer
	>$50k
race	white
	black
	asian
	other
	hispanic
ca	Yes
	No
	Metastatic

Table 4: A description of each level within each categorical variable in the dataset.

Table 5: Comparison between the Cox model and case-base sampling

Feature	Cox model	Case-base sampling
Model type	Semi-parametric	Fully parametric
Time	Left hand side of the formula	Right hand side (allows flexible modeling of time)
Cumulative incidence	Step function	Smooth-in-time curve
Non-proportional hazards	Interaction of covariates with time	Interaction of covariates with time
Model testing		Use GLM framework (e.g. LRT, AIC, BIC)
Competing risks	Difficult	Cause-specific CIFs