casebase: An Alternative Framework For Survival Analysis and Comparison of Event Rates

by Sahir Rai Bhatnagar, Maxime Turgeon*, Jesse Islam, James A. Hanley, and Olli Saarela*

Abstract In epidemiological studies of time-to-event data, a quantity of interest to the clinician is their patient's risk of an event. However, methods relying on time matching or risk-set sampling (including Cox regression) eliminate the baseline hazard from the estimating function. As a consequence, the focus has been on reporting hazard ratios instead of survival curves. Indeed, reporting patient risk requires a separate estimation of the baseline hazard. Using case-base sampling, Hanley & Miettinen (2009) explained how parametric hazard functions can be estimated in continuous-time using logistic regression. Their approach naturally leads to estimates of the survival function that are smooth-in-time.

In this paper, we present the **casebase** R package, a comprehensive and flexible toolkit for parametric survival analysis. We describe how the case-base framework can be used in more complex settings: non-linear functions of time and non-proportional hazards, competing risks, and variable selection. Our package also includes an extensive array of visualization tools to complement the analysis. We illustrate all these features through three different case studies.

* SRB and MT contributed equally to this work.

Introduction

The semiparametric Cox model has become the default approach to survival analysis even though Cox himself later suggested he would prefer to model the hazard function directly. In a 1994 interview with Professor Nancy Reid, Sir David Cox was asked how he would model a set of censored survival data, to which he responded: "I think I would normally want to tackle problems parametrically ... and if you want to do things like predict the outcome for a particular patient, it's much more convenient to do that parametrically" (**?**). Indeed, the most relevant quantity in a clinical setting is often the 5- or 10-year risk of experiencing a certain event given the patient's particular circumstances. Unfortunately, the most reported metric from a Cox model is the (potentially time-dependent) hazard ratio (HR), which ignores the duration of follow-up and is subject to selection bias (**?**). The covariate-adjusted survival curve overcomes these limitations, and it is arguably a more important summary measure to report than the HR. While stepwise survival curves can be computed with the Cox model, they require a second step to separately estimate the baseline hazard (**?**).

Several authors have since pursued fully parametric approaches that made the fitting of smooth survival curves much more transparent and intuitive through generalized linear models. The key feature of these procedures is splitting the time axis into discrete intervals. Whitehead (**?**) showed the equivalence between a Cox model and a Poisson regression with a parameter for each event time; Carstensen (**?**) provides a nice exposition of this equivalence with a real data example and supporting R code for computing standard errors. Arjas & Haara (**?**) and Efron (**?**) treated each patient-day as a Bernoulli random variable with probability equal to the discrete hazard rate. A potential issue with these approaches is that the number of time bins need to be chosen by the data analyst. On the one hand, a fine grouping of times may result in few (or none) events per interval, which then leads to instability in the Newton-Raphson procedure for estimation (**?**, Section 4.8). On the other hand, a coarse grouping could potentially mask nonlinear trends in the hazard function.

Rather than discretizing time, Hanley & Miettinen (**?**) selected a discrete set of person-time coordinates ('person-moments') in continuous time from all observed follow-up experience constituting the study base. By doing so, they obtained a likelihood expression for the hazard function that is equivalent to that of logistic regression. More specifically, all person-moments when the event of interest occurred are selected as the case series, complemented by a randomly sampled base series of person-moments serving as controls. This approach allows flexible modeling of the hazard function by including time as a covariate (e.g. using splines or general additive models). Furthermore, time-dependent covariates can be modeled through interactions with time. In short, Hanley & Miettinen (**?**) use the well-understood logistic regression for directly modeling the hazard function, without requiring a discrete-time model.

In this article, we present the **casebase** R package (**?**) which implements the Hanley & Miettinen (**?**) approach for fitting fully parametric hazard models and covariate-adjusted survival curves using the familiar interface of the `glm` function. Our implementation allows for straightforward extensions

to other models such as penalized regression for variable selection and competing-risk analysis. In addition, we provide functions for exploratory data analysis and visualizing the estimated quantities such as the hazard function, survival curve, and their standard errors. The ultimate goal of our package is to make fitting flexible hazards accessible to more end users with the hope that they will favor reporting absolute risks over hazard ratios.

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 three case studies that illustrate the flexibility and capabilities of **casebase**. We show how the same framework can be used for non-linear functions of time and non-proportional hazards, competing risks, and variable selection via penalized regression. Finally, we end the article with a discussion of the results and of future directions.

Theoretical details

As discussed in Hanley & Miettinen (**?**), the key idea behind case-base sampling is to consider the entire study base as an infinite collection 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 (i.e. whether the event happened) attached to it. By comparing person moments at which an event occurred with person moments at which no event occurred, we can extract information about hazards and survival.

Therefore, we start by sampling all person moments at which the event occurred; this collection of person moments is what Hanley & Miettinen call the *case series*. The incidence of the case series is dictated by the hazard function of interest. Next, we sample a finite number of person moments at which no event occurred; this second collection of person moment is what Hanley & Miettinen call the *base series*. The sampling mechanism for the base series is left at the discretion of the user, but in practice we find that sampling uniformly from the study base provides both simplicity and good performance. This is the default sampling mechanism in the package.

Likelihood and estimating function

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. Instead, following Aalen *et al* (**?**), we use the placeholder "past" to denote the past history of the corresponding process. The reader interested in more details can refer to Saarela & Arjas (**?**) and Saarela (**?**). 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 (**?**)). 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

$$\lambda_i(t)dt = E[dN_i(t) \mid \text{past}].$$

The processes $N_i(t)$ count the person moments from the *case series*.

To complement the case series, we sample person moments for the base series. To do so, we model the base series sampling mechanism using non-homogeneous Poisson processes $R_i(t) \in \{0, 1, 2, \ldots\}$; the person-moments where $dR_i(t) = 1$ constitute the base series. We note that the same individual can contribute multiple person-moments to 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. As mentioned above, the processes $R_i(t)$ are typically defined by the user via its intensity function $\rho_i(t)$. The process $Q_i(t)$ is characterized by $E[dQ_i(t) \mid \text{past}] = \lambda_i(t)dt + \rho_i(t)dt$.

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

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

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

Case-base sampling allows us to avoid this integral. By conditioning on a sampled person-moment,

we get individual likelihood contributions of the form

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

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

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

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 (**?**) 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 (**?**), the authors suggest sampling the base series *uniformly* from the study base. In terms of Poisson processes, their 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 (e.g. the sum of all individual follow-up times). More complex examples are also possible; see for example Saarela & Arjas (**?**), where the intensity functions for the sampling mechanism are proportional to the cardiovascular disease event rate given by the Framingham score. Non-uniform sampling mechanisms can increase the efficiency of the case-base estimators.

Common parametric models

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:

$$\log(\lambda(t;X)) = \beta_0 + \beta_1 X. \tag{2}$$

In this model, the instantaneous failure rate is independent of t.[1]

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

$$\log(\lambda(t;X)) = \beta_0 + \beta_1 t + \beta_2 X. \tag{3}$$

Use of $\log(t)$ yields the Weibull hazard which allows for a power dependence of the hazard on time (**?**):

$$\log(\lambda(t;X)) = \beta_0 + \beta_1 \log(t) + \beta_2 X. \tag{4}$$

Competing-risk analysis

Case-base sampling can also be used in the context of competing-risk analysis. Assuming there are J competing events, we can show that each person-moment's contribution to 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 cause-specific hazard function. As may be expected, this functional form is similar to the terms appearing in the likelihood function for multinomial regression.[2]

[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* (**?**).

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

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

Variable selection

To perform variable selection on the regression parameters $\theta \in \mathbb{R}^p$ of the hazard function, we can add a penalty to the likelihood and optimise the following equation:

$$\min_{\theta \in \mathbb{R}^p} \; -\ell(\theta) + \sum_{j=1}^{p} w_j P(\theta_j; \lambda, \alpha) \tag{5}$$

where $\ell(\theta) = \log L(\theta)$ is the log of the likelihood function given in (1), $P(\theta_j; \lambda, \alpha)$ is a penalty term controlled by the non-negative regularization parameters λ and α, and w_j is the penalty factor for the jth covariate. These penalty factors serve as a way of allowing parameters to be penalized differently. For example, we could set the penalty factor for time to be 0 to ensure it is always included in the selected model.

Comparison with existing packages

Survival analysis is an important branch of applied statistics and epidemiology. Accordingly, there is already a vast ecosystem of R packages implementing different methodologies. 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 *Survival* Task View reveals that there are over 250 packages related to survival analysis (**?**). For the purposes of this article, we restricted our review 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 (CI) 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 multistate models. The remaining 14 packages appear in Table 1, along with some of the functionalities they offer.

Parametric survival models are implemented in several packages, each differing in the parametric distributions available: **CFC** (**?**), **flexsurv** (**?**), **SmoothHazard** (**?**), **rstpm2** (**?**), **mets** (**?**), and **survival** (**?**). For example, **SmoothHazard** is limited to Weibull distributions (**?**), whereas both **flexsurv** and **survival** allow users to supply any distribution of their choice. **flexsurv**, **SmoothHazard**, **mets** and **rstpm2** can model the effect of time using splines, which allows flexible modeling of the hazard function. As discussed above, **casebase** can model any parametric family whose log-hazard can be expressed as a linear combination of covariates (including time). Therefore, our package is more general in that it allows the user to model any linear or non-linear transformation of time including splines and higher order polynomials. Also, by including interaction terms between covariates and time, it also allows users to fit (non-proportional) time-varying coefficient models. However, unlike **flexsurv**, we do not explicitly model any shape parameter.

Several packages implement penalized estimation for the Cox model: **glmnet** (**?**), **glmpath** (**?**), **penalized** (**?**), **riskRegression** (**?**). Moreover, some packages also include penalized estimation in the context of Cox models with time-varying coefficients: elastic-net penalization with **rstpm2** (**?**), while **survival** (**?**) has an implementation of ridge-penalized estimation. On the other hand, our package **casebase** provides penalized estimation of the hazard function. To our knowledge, **casebase** and **rstpm2** are the only packages to offer this functionality.

Next, several R packages implement methodologies for competing risk analysis; for a different perspective on this topic, see Mahani & Sharabiani (**?**). The package **survival** provides functionality for competing-risk analysis and multistate modelling. The package **cmprsk** provides methods for cause-specific subdistribution hazards, such as in the Fine-Gray model (**?**). On the other hand, the package **CFC** estimates cause-specific CIs from unadjusted, non-parametric survival functions. Our package **casebase** also provides functionalities for competing risk analysis by estimating parametrically the cause-specific hazards. From these quantities, we can then estimate the cause-specific CIs.

Finally, several packages include functions to estimate the survival function and the CI. 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 explicitly modeled. Using this estimate, the survival function and the CI can be obtained using their functional relationships (see Equations 6 and 7 below). Packages providing this functionality include **CFC**, **flexsurv**, **mets**, and **survival**. Our package **casebase** also follows this approach for both single-event and competing-risk analyses. The second category outlined above broadly corresponds to semi-parametric 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) or parametric estimators (e.g. spline functions). Packages that implement this approach include **riskRegression**, **rstpm2**, **survival**, and **glmnet**. As mentioned in the introduction, a key distinguishing factor between these two approaches is that the first category leads to smooth estimates of the survival function, whereas the second category often produces estimates in the form of stepwise functions.

Package	Competing Risks	Allows Non PH	Penalized Regression	Splines	Parametric	Semi Parametric	Interval/Left Censoring	Risk Estimates
casebase	✓	✓	✓	✓	✓			✓
CFC	✓	✓			✓			✓
cmprsk	✓					✓		✓
crrp	✓		✓			✓		
fastcox			✓			✓		
flexrsurv		✓		✓	✓			✓
flexsurv	✓	✓		✓	✓			✓
glmnet			✓			✓		✓
glmpath			✓			✓		
mets	✓			✓		✓		✓
penalized			✓			✓		
riskRegression	✓		✓			✓		✓
rstpm2		✓		✓	✓	✓	✓	✓
SmoothHazard		✓		✓	✓		✓	
survival	✓	✓			✓	✓	✓	✓

Table 1: Comparison of various R packages for survival analysis. **Competing Risks**: whether an implementation for competing risks is present. **Allows Non PH**: includes models for non-proportional hazards. **Penalized Regression**: allows for a penalty term on the regression coefficients when estimating hazards (e.g. lasso or ridge). **Splines**: allows a flexible fit on time through the use of splines. **Parametric**: implementation for parametric models. **Semi-parametric**: implementation for semi-parametric models. **Interval/left censoring**: models for interval and left-censoring. If this is not selected, the package only handles right-censoring. **Risk estimates**: estimation of survival curve and cumulative incidence is available.

Implementation details

The functions in the **casebase** package can be divided into two categories: 1) exploratory data analysis, in the form of population-time plots; and 2) parametric modeling of the hazard function. We strove for compatibility with both `data.frames` and `data.tables`; this can be seen in the coding choices we made and the unit tests we wrote.

Population-time plots

Population-time plots are a descriptive visualization of incidence density, where the study base is represented by area and events by points within the area. The case-base sampling approach described above can be visualized in the form of a population time plot. These plots are informative graphical displays of survival data and should be one of the first steps in an exploratory data analysis. The `popTime` function and `plot` method facilitate this task:

1. The `casebase::popTime` function takes as input the original dataset along with the column names corresponding to the timescale, the event status and an exposure group of interest (optional). This will create an object of class `popTime`.

2. The corresponding `plot` method for the object created in Step 1 can be called to create the population time plot with several options for customizing the aesthetics.

By splitting these tasks, we give flexibility to the user. While the method call in Step 2 allows further customization by using the **ggplot2** (**?**) family of functions, users may choose the graphics system of their choice to create population-time plots from the object created in Step 1.

To illustrate these functions, we will use data from the European Randomized Study of Prostate Cancer Screening (ERSPC) (**?**) which was extracted using the approach described in Liu *et al.* (**?**). This dataset is available through the **casebase** package. It contains the individual observations for 159,893 men from seven European countries, who were between the ages of 55 and 69 years when recruited for the trial.

We first create the necessary dataset for producing the population time plot using the `popTime` function. In this example, we stratify the plot by treatment group. The resulting object inherits from class `popTime` and stores the exposure variable as an attribute:

```
pt_object <- casebase::popTime(ERSPC, time = "Follow.Up.Time",
                               event = "DeadOfPrCa", exposure = "ScrArm")
inherits(pt_object, "popTime")

#> [1] TRUE

attr(pt_object, "exposure")

#> [1] "ScrArm"
```

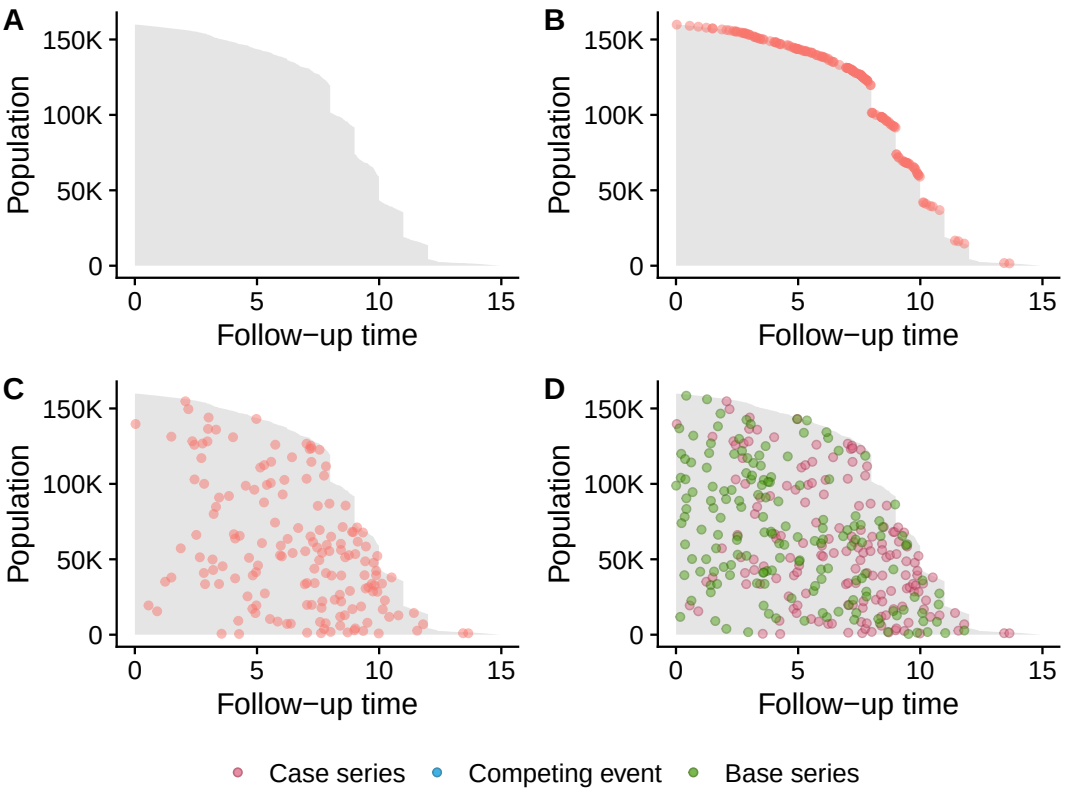

Figure 1: Population time plot for the ERSPC dataset. **A**: The gray area can be thought of as N = 159,893 infinitely thin horizontal rectangles ordered by length of follow-up. **B**: The red points correspond to when death has occurred for any one of those infinitely thin rectangles. **C**: To improve visibility, these red points are randomly redistributed along their respective x-coordinates, providing a visualization of incidence density. More events are observed at later follow-up times, motivating the use of non-constant hazard models. **D**: The base series, a representative sample of the entire grey area, is represented by the green points.

We then pass this object to the corresponding `plot` method:

```
plot(pt_object, add.base.series = TRUE)
```

Figure 1 depicts the process of creating a population-time plot. It is built sequentially by first adding a layer for the area representing the population time in gray (Figure 1 A), with subjects having the least amount of observation time plotted at the top of the y-axis. We immediately notice a distinctive *stepwise shape* in the population time area. This is due to the randomization of the Finnish cohorts which were carried out on January 1 of each of year from 1996 to 1999. Coupled with the uniform December 31 2006 censoring date, this led to large numbers of men with exactly 11, 10, 9 or 8 years of follow-up. Tracked backwards in time (i.e. from right to left), the population-time plot shows the recruitment pattern from its beginning in 1991, and the January 1 entries in successive years. Tracked forwards in time (i.e. from left to right), the plot for the first three years shows attrition due entirely to death (mainly from other causes). Since the Swedish and Belgian centres were the last to complete recruitment in December 2003, the minimum potential follow-up is three years. Tracked

further forwards in time (i.e. after year 3) the attrition is a combination of deaths and staggered entries. As we can see, population-time plots summarise a wealth of information about the study into a simple graph.

Next, layers for the case series and base series are added. The y-axis location of each case moment is sampled at random vertically on the plot to avoid having all points along the upper edge of the gray area (Figure 1 B). By randomly distributing the cases, we can get a sense of the incidence density. In Figure 1 C, we see that more events are observed at later follow-up times. Therefore, a constant hazard model would not be appropriate in this instance as it would overestimate the incidence earlier on in time, and underestimate it later on. Finally, the base series is sampled uniformly from the study base (Figure 1 D). The reader should refer to the package vignettes for more examples and a detailed description of how to modify the aesthetics of a population-time plot.

Parametric modeling

The parametric modeling step was separated into three parts:

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

By separating the sampling and estimation functions, we allow the possibility of users implementing more complex sampling scheme (as described in Saarela (**?**)), or more complex study designs (e.g. time-varying exposure).

The sampling scheme selected for `sampleCaseBase` was described in Hanley & Miettinen (**?**): 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.

Next, 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 1 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 the elastic-net (**?**) penalty (cf. Case Study 3). This functionality is provided through the **glmnet** package (**?**).
- `gam`: This option provides support for *Generalized Additive Models* via the **mgcv** package (**?**).

In the case of multiple competing 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

$$S(t) = \exp\left(-\int_0^t \lambda(u; X)du\right). \tag{6}$$

The integral is computed using either the numerical or Monte-Carlo integration. The risk function (or

cumulative distribution function) is then defined as

$$F(t) = 1 - S(t). \tag{7}$$

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:

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

By integrating these subdensities, we obtain the event-specific CI functions:

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

Again, 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.

Finally, the output from `absoluteRisk` can be passed to a method `confint` to compute confidence bands around the survival function. These bands are computed using parametric bootstrap, and therefore are only valid when `family = "glm"` as it relies on the asymptotic normality of the estimator. Currently, this is only available for the single-event setting.

Illustration of package

In this section, we illustrate the main functions of the **casebase** package through three case studies. Each one showcases a different type of analysis. First, we show how to model time flexibly as well as non-proportional hazards. Then we perform a competing-risk analysis and compare our results with the Cox model and the Fine-Gray model. The third case study illustrates how to perform variable selection in high-dimensional datasets.

Case study 1—Modeling different functions of time and non-proportional hazards

For our first case study, we return to the ERSPC study and investigate the differences in risk between the control and screening arms. Previous re-analyses of these data suggest that the 20% reduction in prostate cancer death due to screening was an underestimate (**?**). The estimated 20% (from a proportional hazards model) did not account for the delay between screening and the time the effect is expected to be observed. As a result, the null effects in years 1–7 masked the substantial reductions that began to appear from year 8 onward. This motivates the use of a time-dependent hazard ratio which can easily be fit with the **casebase** package by including an interaction term with time in the model. We fit a flexible hazard by using a smooth function of time modeled with a penalised cubic spline basis with 2 degrees of freedom. The model is fit using `fitSmoothHazard` with the familiar formula interface:

```
library(survival) # for the pspline function
fit <- fitSmoothHazard(DeadOfPrCa ~ pspline(Follow.Up.Time, df = 2) * ScrArm,
                       data = ERSPC, ratio = 100)
```

The output object from `fitSmoothHazard` inherits from the `singleEventCB` and `glm` classes. For this reason, we can leverage the `summary` method for `glm` objects to output a familiar summary of the results:

```
summary(fit)
```

```
#> Fitting smooth hazards with case-base sampling
#>
#> Sample size: 159893
#> Number of events: 540
#> Number of base moments: 54000
#> ----
```

```
#>
#> Call:
#> fitSmoothHazard(formula = DeadOfPrCa ~ pspline(Follow.Up.Time,
#>     df = 2) * ScrArm, data = ERSPC, ratio = 100)
#>
#> Deviance Residuals:
#>    Min      1Q  Median      3Q     Max
#> -0.477  -0.164  -0.140  -0.084   3.882
#>
#> Coefficients:
#>                                                    Estimate Std. Error
#> (Intercept)                                         -12.613      9.756
#> pspline(Follow.Up.Time, df = 2)1                      1.320     10.705
#> pspline(Follow.Up.Time, df = 2)2                      5.190      9.526
#> pspline(Follow.Up.Time, df = 2)3                      4.481      9.866
#> pspline(Follow.Up.Time, df = 2)4                      6.060      9.690
#> pspline(Follow.Up.Time, df = 2)5                      5.231      9.851
#> pspline(Follow.Up.Time, df = 2)6                      9.176      9.749
#> pspline(Follow.Up.Time, df = 2)7                     -0.434     19.572
#> ScrArmScreening group                                 7.900     13.050
#> pspline(Follow.Up.Time, df = 2)1:ScrArmScreening group   -7.962     14.507
#> pspline(Follow.Up.Time, df = 2)2:ScrArmScreening group   -8.351     12.690
#> pspline(Follow.Up.Time, df = 2)3:ScrArmScreening group   -7.787     13.224
#> pspline(Follow.Up.Time, df = 2)4:ScrArmScreening group   -8.098     12.942
#> pspline(Follow.Up.Time, df = 2)5:ScrArmScreening group   -9.389     13.223
#> pspline(Follow.Up.Time, df = 2)6:ScrArmScreening group   -7.385     13.114
#> pspline(Follow.Up.Time, df = 2)7:ScrArmScreening group  -17.380     30.806
#>                                                    z value Pr(>|z|)
#> (Intercept)                                          -1.29     0.20
#> pspline(Follow.Up.Time, df = 2)1                      0.12     0.90
#> pspline(Follow.Up.Time, df = 2)2                      0.54     0.59
#> pspline(Follow.Up.Time, df = 2)3                      0.45     0.65
#> pspline(Follow.Up.Time, df = 2)4                      0.63     0.53
#> pspline(Follow.Up.Time, df = 2)5                      0.53     0.60
#> pspline(Follow.Up.Time, df = 2)6                      0.94     0.35
#> pspline(Follow.Up.Time, df = 2)7                     -0.02     0.98
#> ScrArmScreening group                                 0.61     0.54
#> pspline(Follow.Up.Time, df = 2)1:ScrArmScreening group   -0.55     0.58
#> pspline(Follow.Up.Time, df = 2)2:ScrArmScreening group   -0.66     0.51
#> pspline(Follow.Up.Time, df = 2)3:ScrArmScreening group   -0.59     0.56
#> pspline(Follow.Up.Time, df = 2)4:ScrArmScreening group   -0.63     0.53
#> pspline(Follow.Up.Time, df = 2)5:ScrArmScreening group   -0.71     0.48
#> pspline(Follow.Up.Time, df = 2)6:ScrArmScreening group   -0.56     0.57
#> pspline(Follow.Up.Time, df = 2)7:ScrArmScreening group   -0.56     0.57
#>
#> (Dispersion parameter for binomial family taken to be 1)
#>
#>     Null deviance: 6059.0  on 54539  degrees of freedom
#> Residual deviance: 5772.7  on 54524  degrees of freedom
#> AIC: 5805
#>
#> Number of Fisher Scoring iterations: 9
```

As noted in the Theoretical Details section, the usual asymptotic results hold for likelihood ratio tests built using case-base sampling models. Therefore, we can easily test the significance of the spline model:

```
#> Analysis of Deviance Table
#>
#> Model: binomial, link: logit
#>
#> Response: DeadOfPrCa
#>
#> Terms added sequentially (first to last)
#>
```

```
#>
#>                                       Df Deviance Resid. Df Resid. Dev
#> NULL                                                  54539       6059
#> pspline(Follow.Up.Time, df = 2)        7    269.1     54532       5790
#> ScrArm                                 1      8.0     54531       5782
#> pspline(Follow.Up.Time, df = 2):ScrArm 7      9.1     54524       5773
#>                                       Pr(>Chi)
#> NULL
#> pspline(Follow.Up.Time, df = 2)          <2e-16 ***
#> ScrArm                                   0.0046 **
#> pspline(Follow.Up.Time, df = 2):ScrArm   0.2454
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis is rejected in favor of the spline model. Similarly, to compare different models (e.g. time modeled linearly), we could compute Akaike's Information Criterion (AIC) for each model and compare them.

Time-dependent hazard ratios

Note that we did not have to specify any cut points, as would be the case with the `survSplit` function in the **survival** package. In Figure 2, we have the estimated hazard ratio and 95% confidence interval for screening vs. control group as a function of time using the `plot` method for objects of class `singleEventCB`:

```
new_time <- seq(1, 12, by  = 0.1)
new_data <- data.frame(ScrArm = factor("Control group",
                                       levels = c("Control group","Screening group")),
                       Follow.Up.Time = new_time)
plot(fit, type = "hr", newdata = new_data,
     var = "ScrArm", xvar = "Follow.Up.Time", ci = F)
```

The plot shows that the effect of screening only becomes statistically apparent by year 7 and later. The 25-60% reductions seen in years 8-12 of the study suggests a much higher reduction in prostate cancer due to screening than the single overall 20% reported in the original article.

Bibliography

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

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

Jesse Islam
McGill University
1020 Pine Avenue West Montreal, QC, Canada H3A 1A2

jesse.islam@mail.mcgill.ca

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

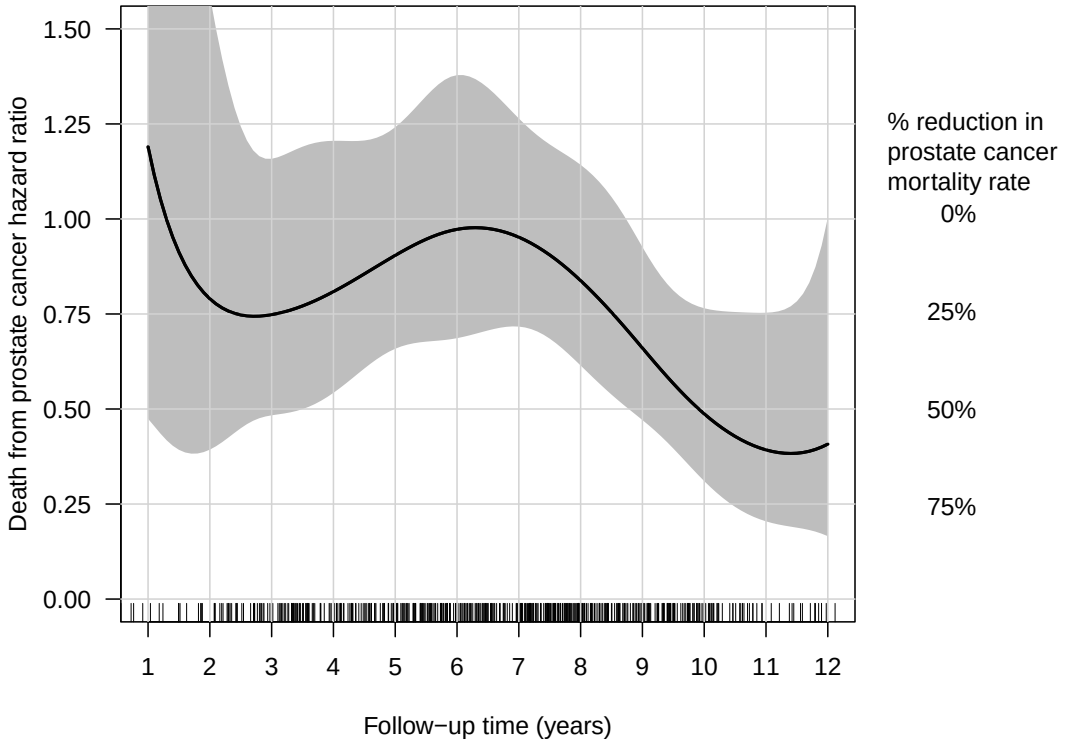

Figure 2: Estimated hazard ratio and 95% confidence interval for screening vs. control group as a function of time in the ERSPC dataset. Hazard ratios are estimated from fitting a parametric hazard model as a function of the interaction between a cubic B-spline basis of follow-up time and treatment arm. 95% confidence intervals are calculated using the delta method. The plot shows that the effect of screening only begins to become statistically apparent by year 7. The 25-60% reductions seen in years 8-12 of the study suggests a much higher reduction in prostate cancer due to screening than the single overall 20% reported in the original article.

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