




Event History Regression with Pseudo-Observations: Computational Approaches and an Implementation in R

Michael C. Sachs 
Karolinska Institutet

Erin E. Gabriel 
Karolinska Institutet

Abstract

Due to tradition and ease of estimation, the vast majority of clinical and epidemiological papers with time-to-event data report hazard ratios from Cox proportional hazards regression models. Although hazard ratios are well known, they can be difficult to interpret, particularly as causal contrasts, in many settings. Nonparametric or fully parametric estimators allow for the direct estimation of more easily causally interpretable estimands such as the cumulative incidence and restricted mean survival. However, modeling these quantities as functions of covariates is limited to a few categorical covariates with nonparametric estimators, and often requires simulation or numeric integration with parametric estimators. Combining pseudo-observations based on non-parametric estimands with parametric regression on the pseudo-observations allows for the best of these two approaches and has many nice properties. In this paper, we develop a user friendly, easy to understand way of doing event history regression for the cumulative incidence and the restricted mean survival, using the pseudo-observation framework for estimation. The interface uses the well known formulation of a generalized linear model and allows for features including plotting of residuals, the use of sampling weights, and correct variance estimation.

Keywords: survival analysis, competing risks, pseudo-observations, regression, R.

1. Introduction

Approaches to event history modeling with covariates can be designated into three categories: nonparametric, semi-parametric, and fully parametric modeling. Under these three paradigms, the flexibility with which one can incorporate covariate information, as well as the estimands of interest, increases from nonparametric to fully parametric, but so do the

assumptions that are required. Semiparametric Cox regression occupies the middle ground of modeling assumptions, having an unspecified baseline hazard but still allowing for multiple and continuous covariates (Cox 1972).

The vast majority of clinical and epidemiological papers with time-to-event data use hazard ratios as their primary estimand. We believe this is due to two things, tradition, and how easy Cox models are to estimate using standard statistical software. Fully parametric survival models require some understanding of the parameterization to interpret the results. The results of a Cox model however are familiar even to first year medical students, yet hazard ratios are commonly misinterpreted as relative risks (Sutradhar and Austin 2018). Furthermore, as was pointed out by Aalen, Cook, and Røysland (2015); Martinussen, Vansteelandt, and Andersen (2020), hazard ratios are difficult to interpret as causal contrasts in many settings. Nonparametric or fully parametric estimators allow for the direct estimation of cumulative estimands that do not condition on past survival, which are therefore more easily causally interpretable. However, incorporation of covariate information is limited to a few categorical covariates in the former, and interpretation of the coefficients directly is challenging in the latter.

With the term “cumulative estimand” we are referring to quantities that can be expressed as expectations of functionals of random variables that represent times to some event of interest that do not condition on past survival. This is in contrast to the hazard function, which is defined in terms of the probability of failing at a particular time conditional on surviving up to just before that time. Cumulative estimands that are commonly of interest are the probability of surviving beyond a particular time (the survivor function or survival), the probability of failing before a particular time (the cumulative incidence function or risk), and the restricted mean survival. What we call cumulative estimands are sometimes referred to as “marginal” to distinguish them from the hazard which is conditional on past survival. However, since we are interested in regression modeling conditional on covariates, we will use the term “cumulative” for clarity.

In many settings, the outcome of interest may be the time to failure due to one cause (for example, death due to cancer), while the remaining causes (death not due to cancer) would be considered competing causes or competing risks. In the presence of competing risks, cumulative estimands include the probability of failing due to a particular cause before a fixed time (the cause-specific cumulative incidence, also called the subdistribution) and the restricted mean time lost (Zhao *et al.* 2018). In Cox regression, competing risks are often treated as censoring events, but these cumulative estimands are related to the cause-specific hazards of all of the causes, and hazards based on the subdistribution functions are even more difficult to interpret.

Contrasts of cumulative estimands, such as the difference in survival probabilities, have easier causal interpretations, and although there have been several methods suggested in the literature to model the effect of covariates on them, each has its limitations. An overview of fully parametric models is provided by Royston and Parmar (2002), yet these models have similar drawbacks as the Cox model and often require computationally complex post-estimation and standardization to describe covariate effects on cumulative survival quantities. The Fine-Gray model (Fine and Gray 1999) is touted as a model for the cause specific cumulative incidence function, yet the main output from that model are ratios of the hazards defined by the subdistribution functions which lack a useful interpretation in terms of an effect on the cumulative incidence (Austin and Fine 2017). Scheike, Zhang, and Gerds (2008) and Tian, Zhao, and

Wei (2014) developed inverse probability of censoring weighted estimating equation methods for using covariates to predict the cumulative incidence probability and restricted mean event time, respectively. These methods, however, can be statistically inefficient since they omit the censored observations from the estimating equations, and can be difficult to use and model dependent because they require modeling of the censoring distribution.

Pseudo-observations, as introduced by Andersen, Klein, and Rosthøj (2003), can be used to fill this gap. Pseudo-observations are calculated for each individual in a sample based on jackknife values calculated using nonparametric estimators of cumulative estimands. It has been shown that using these pseudo-observations as the outcome (instead of the time and event indicator pair) of regression models provides asymptotically unbiased estimates of the associations of the covariates included in the model on the survival outcome of interest (Graw, Gerds, and Schumacher 2009; Jacobsen and Martinussen 2016; Overgaard, Parner, and Pedersen 2017). The key advantage is that they allow direct parametrization of covariate associations with cumulative survival quantities of interest at a single or multiple times points simultaneously.

Our goal is to provide an user friendly, easy to understand way of doing event history regression for cumulative survival estimands, using the pseudo-observation framework. We do this in our R package **eventglm** (Sachs and Gabriel 2022), using the well known formulation of a generalized linear model (GLM) or generalized estimating equations (GEE) that builds on and leverages the existing infrastructure for such models, specifically in the **stats** package (R Core Team 2021) and the **geepack** package (Halekoh, Højsgaard, and Yan 2006). In this paper we describe our implementation of pseudo-observation based approaches to event history regression in the R package **eventglm**, with the primary functions `cumincglm` and `rmeanglm`, highlighting the interpretation and useful properties of this approach. We use simulated data to evaluate and compare the performance of the different methods to handle covariate dependent censoring, and the different variance estimators in the single time point setting, thus informing the default choices for the methods. Example data analyses are illustrated on two datasets that are included in the package, showing how to use the package and interpret the output. We show how the package is set up so that it can be extended to accommodate new or different methods for pseudo-observation calculation. Finally, we compare the computational performance of our implementation to the existing approaches for calculating pseudo-observations. The package **eventglm** is available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/package=eventglm>.

1.1. Related work

A variety of regression models for time-to-event outcomes can be specified and estimated using different R packages, but our main focus is on regression models for the cumulative incidence and restricted mean survival. The primary infrastructure for the analysis of time-to-event outcomes in R is in the **survival** package (Therneau and Grambsch 2000; Therneau 2022). Regression models for the cumulative incidence function are available in the **timereg** (Scheike and Zhang 2011; Scheike and Martinussen 2006) and **riskRegression** (Gerds and Kattan 2021) packages, which use the direct binomial approach for estimation in addition to the Fine-Gray model for competing risks. The **Cprob** package (Allignol 2018; Allignol, Latouche, Yan, and Fine 2011) implements cumulative incidence regression using temporal process regression or the pseudo-observation approach. For computation of pseudo-observations, the

pseudo (Pohar-Perme and Gerster 2017) and **fastpseudo** (Batten 2015) packages are designed specifically for that task, while the **prodlim** (Gerds 2019) package provides functions to do that for the cumulative incidence function. A **Stata** (StataCorp 2019) package (Overgaard, Andersen, and Parner 2015) and a **SAS** (SAS Institute Inc. 2013) macro (Klein, Gerster, Andersen, Tarima, and Perme 2008) exist for computation of pseudo-observations. With all of these packages that only compute pseudo-observations, it is left to the user to specify regression models, estimate them, and perform inference. This provides a great deal of flexibility, but also is a barrier to entry for less experienced users of statistical methods.

2. Notation and estimands

Let T_i denote the time to event, $\delta_i \in \{1, \dots, d\}$ denote the indicator of the cause of the event for d competing causes, and X_i a vector of covariates for subject $i = 1, \dots, n$. We will use V_i to denote transformations of (T_i, δ_i) whose expectations represent summary statistics of interest. Specifically, we consider the following, where $1\{\cdot\}$ denotes the indicator function that is 1 if the event in brackets is true, and 0 otherwise,

- The cause specific cumulative incidence of cause k at time t^* : $V_i = 1\{T_i < t^*, \delta_i = k\}$ and $E(V_i) = P(T_i < t^*, \delta_i = k)$.
- In the case where $d = 1$, the cumulative incidence (one minus survival) at time t^* : $V_i = 1\{T_i < t^*\}$ and $E(V_i) = P(T_i < t^*)$.
- The expected lifetime lost due to cause k up to time t^* : $V_i = (t^* - \min\{T_i, t^*\})1\{\delta_i = k\}$ and $E(V_i) = \int_0^{t^*} P(T_i < u, \delta_i = k) du$, as was shown in Andersen (2013).
- In the case where $d = 1$, the restricted mean survival up to time t^* : $V_i = \min\{T_i, t^*\}$ and $E(V_i) = \int_0^{t^*} P(T_i > u) du$.

Our main interest is in estimating the parameters of a generalized linear regression model for V_i conditional on covariates X_i :

$$E(V_i | X_i) = g^{-1}\{X_i^\top \beta\}, \quad (1)$$

for some specified link function g .

The interpretation of the coefficients will depend on estimand of interest and the link function, which is specified using the `link` argument of `cumincglm` and `rmeanglm`. For a model of the cumulative incidence using the identity link (the default) $g(x) = x$, and for a single binary covariate, we have

$$P(T_i < t^* | X_i = x_i) = \beta_0 + \beta_1 x_i,$$

so that $\beta_1 = P(T_i < t^* | X_i = 1) - P(T_i < t^* | X_i = 0)$, called the risk difference. This is often of interest in medical studies to summarize the effect of a treatment or exposure. With the log link (`link="log"`), we obtain $\exp(\beta_1) = P(T_i < t^* | X_i = 1)/P(T_i < t^* | X_i = 0)$, the relative risk or risk ratio. If instead our outcome is the restricted mean, for the identity link the β_1 is interpreted as the difference in restricted means comparing $X_i = 1$ to $X_i = 0$. With the log link, we obtain the ratio of restricted means.

If odds ratios are of interest, then the `link = "logit"` option can be used for the cumulative incidence. Another interesting option is the `link = "cloglog"`: $g(x) = \log\{-\log(1-x)\}$, the

complementary log log link for the cumulative incidence implies proportional hazards. Thus models using the complementary log log link applied at various time points can be used to estimate hazard ratios (subdistribution hazard ratios in the competing risks case (Austin and Fine 2017)) and to assess the proportional hazards assumption (Perme and Andersen 2008). Other options for link functions are probit, inverse, μ^{-2} , square root, and users can define custom link functions. It is not immediately clear what the interpretation of the regression coefficients would be in these cases, but they are possible because the specification is based on the `quasi` family. See the `stats::family` help file for more details on the possible link functions.

An important property of the effect measures derived from these models is collapsibility. To see this, consider another binary covariate Z_i and the model

$$g\{\mathbf{P}(T_i < t^* \mid X_i = x_i, Z_i = z_i)\} = \beta_0^* + \beta_1^* x_i + \beta_2^* z_i.$$

If g is the identity or log and if Z_i is independent of X_i , then $\beta_1 = \beta_1^*$; this is not generally true for other link functions (Neuhaus and Jewell 1993). Hazard ratios are not generally collapsible, which is related to the fact that the hazard conditions on past survival (Sjölander, Dahlqvist, and Zetterqvist 2016). Collapsibility is an important property in causal inference. This comes up when adjusting for covariates in randomized controlled trials, and when adjusting for measured confounders in observational studies. For more elaboration on this topic, see Andersen, Syriopoulou, and Parner (2017) and Daniel, Zhang, and Farewell (2021).

2.1. Multiple time points

Andersen *et al.* (2003) described a multivariate version of the model in Equation 1 for the cumulative incidence by considering a finite set of time points t_1, \dots, t_k , and the models of the form

$$g\{\mathbf{P}(T_i < t_l \mid X_i = x_i)\} = (\beta_0 + \beta_l) + \beta_1 x_i, l = 1, \dots, k. \quad (2)$$

In this model, the interpretation of the covariate effect β_1 is similar to the models above, e.g., a risk difference or risk ratio, but assuming that the effect is the same at each of the time points included in the model. In the model in Equation 2, the intercept depends on time, but the covariate effect is assumed to be time-constant. The latter assumption can be relaxed to allow the covariate effect to depend on time as well. In that case, there will be one coefficient for each time point, each one representing the covariate effect on the outcome for that time. Models can also include a mix of time varying and time constant covariate effects. All of these types of models can be estimated using `eventglm`.

3. Estimation

We do not observe T_i and δ_i directly, but rather $Y_i = \min\{C_i, T_i\}$ where C_i is the censoring time, and $\Delta_i \in \{0, 1, \dots, d\}$ where where 0 indicates censoring occurred before any of the events. The collection of observations will be denoted Z_1, \dots, Z_n where $Z_i = (Y_i, \Delta_i, X_i)$, and are assumed to be independent and identically distributed.

If there were no censoring before time t^* , then the V_i are all observed for $i = 1, \dots, n$ and the parameters could be estimated using standard methods. When that is not the case, the model can be estimated using pseudo-observations (Andersen *et al.* 2003), the computational methods for which we will describe in the next subsection. Let P_i denote the pseudo-observation

for subject i which will remain abstract for the moment. When the pseudo-observations are computed in a way that

$$\mathbf{E}(P_i | X_i) = \mathbf{E}(V_i | X_i) + o_p(1) \quad (3)$$

in large samples, this motivates the idea of estimating β in Equation 1 by solving the estimating equations

$$\sum_{i=1}^n \frac{\partial g^{-1}}{\partial \beta} A_i^{-1} \{P_i - g^{-1}(X_i^\top \beta)\} = \sum_{i=1}^n U_i(\beta) = 0, \quad (4)$$

for some specified variance parameter A_i which corresponds to the `variance` function of `stats::family` or the working covariance matrix in the case of GEE. In our implementation, the estimating equations are solved using the `glm.fit` function with the `quasi` family, with `variance = "constant"`, i.e., $A_i = \sigma$. It was suggested in Andersen *et al.* (2003) that when the estimand is the cumulative incidence, efficiency gains could potentially be made by specifying the variance function as `mu(1-mu)`. This, however causes a great deal more numerical instability so it is an area of future consideration. In the multiple time point models, GEE is used with the `glm.fit` estimates used as starting values and working independence covariance. The theoretical justification and precise conditions under which the solution to Equation 4 is consistent and asymptotically normal have been studied in (Graw *et al.* 2009; Jacobsen and Martinussen 2016; Overgaard *et al.* 2017; Overgaard, Parner, and Pedersen 2019).

3.1. Pseudo-observations calculation under independent censoring

Andersen *et al.* (2003) developed the original approach using the leave one observation out jackknife. Let $\theta = \mathbf{E}(V_i)$ denote the cumulative summary statistic of interest but marginal with respect to the covariates (i.e., ignoring the covariates) and $\hat{\theta}$ an estimate of that quantity using all of the observations. The estimator is generally nonparametric, e.g., the Aalen-Johansen estimator (Aalen and Johansen 1978) of the cumulative incidence curve, or the Kaplan-Meier estimator (Kaplan and Meier 1958) of the survivor curve, though recently parametric estimators of the marginal quantities have been suggested (Nygård Johansen, Lundbye-Christensen, and Thorlund Parner 2020; Sabathé, Andersen, Helmer, Gerds, Jacqmin-Gadda, and Joly 2020). Let $\hat{\theta}_{-i}$ denote the jackknife estimate obtained by leaving the i th observation out of the sample and recomputing the estimate. Then the i th jackknife pseudo-observation is

$$P_i = n\hat{\theta} - (n-1)\hat{\theta}_{-i}.$$

When θ is the cumulative incidence and the estimate is based on the Aalen-Johansen estimator, then there are some computational tricks so that the estimator does not need to be rerun n times. This approach is implemented in the `prodlim` package (Gerds 2019) that we made some slight modifications to be more memory saving when there is a large dataset. In the case of the restricted mean, no such tricks are readily implemented and the Aalen-Johansen estimator is computed n times and integrated each time.

When P_i is computed in this way based on a nonparametric estimator, a key condition required for Equation 3 to hold is that $(T_i, X_i, \Delta_i) \perp C_i$. This says that censoring is independent of the event times and of all covariates in the model, called completely independent censoring. In that case, the solution to the estimating equations in Equation 4 yields consistent and asymptotically normal estimates of β .

3.2. Pseudo-observations under covariate dependent censoring

If we instead assume for some subset of covariates \tilde{X}_i that $(T_i, X_i, \Delta_i) \perp C_i \mid \tilde{X}_i$ then we can use different approaches to computing the pseudo-observations that will satisfy Equation 3. When \tilde{X}_i only contains categorical covariates with a finite set of combinations, Andersen and Pohar Perme (2010) suggested computing the jackknife P_i separately for each combination of values in \tilde{X}_i . This is implemented in our package and can be obtained using the `model.censoring = "stratified"` option.

If \tilde{X}_i contains continuous covariates, then we can model the censoring mechanism conditional on those covariates and use an inverse probability of censoring weighted (IPCW) marginal estimator. Modeling the censoring process conditional on covariates and using inverse probability of censoring weighted estimators was first explored in Binder, Gerds, and Andersen (2014). This was further developed in Overgaard *et al.* (2019) who showed that the property in Equation 3 holds for IPCW estimators of the cumulative quantity $E(V_i)$. Specifically, let

$$\tilde{V}_i = \begin{cases} V_i & \text{if } C_i > T_i \\ 0 & \text{otherwise} \end{cases},$$

$I_i = 1\{C_i \geq \min(T_i, t^*)\}$, and let $G(s; \tilde{X}_i) = P(C_i \geq s \mid \tilde{X}_i)$. If the censoring mechanism G can be estimated consistently by \hat{G} , then the property in Equation 3 holds for jackknife pseudo-observations based on the marginal estimator

$$\hat{\theta}^b = n^{-1} \sum_{i=1}^n \frac{\tilde{V}_i I_i}{\hat{G}\{\min(\tilde{T}_i, t^*); \tilde{X}_i\}}.$$

In practice, G is estimated using a regression model for the outcome of the time to censoring, that is, using (\tilde{T}_i, B_i) as the outcome where $B_i = 1$ if $\Delta_i = 0$ and $B_i = 0$ otherwise. In our package, we have implemented estimation of the probability of remaining uncensored at the observed time using a Cox model (Cox 1972) combined with the Breslow estimator of the cumulative hazard (`model.censoring = "coxph"`) or using Aalen's additive hazards model (`model.censoring = "aareg"`) (Aalen 1989).

This weighting approach (`ipcw.method = "binder"`) is the default when `model.censoring` is `"coxph"` or `"aareg"`. An alternative formulation is to use the estimator

$$\hat{\theta}^h = \frac{\sum_{i=1}^n \tilde{V}_i w_i}{\sum_{i=1}^n w_i}, \text{ where } w_i = \frac{I_i}{\hat{G}\{\min(\tilde{T}_i, t^*); \tilde{X}_i\}},$$

which is what would be implied as the solution to the first degree method of moments equations (Hájek 1971). This estimator is available using the option `ipcw.method = "hajek"`. In our simulation study, we find similar performance with the (Binder *et al.* 2014) approach.

The covariates \tilde{X}_i are specified as a one-sided formula in the `formula.censoring` argument. If this argument is `NULL`, the default, then the same covariates as specified on the right side of the main formula are used. The covariates are required to be categorical if the `"stratified"` option is used. Otherwise, the censoring formula is just as flexible as in `glm`, allowing for interactions, transformations, splines, and more.

3.3. Variance estimation

Given calculations of the P_i using one of the methods described above, and a specification of the regression model, including the link function g and function A_i in Equation 4, one can then solve the estimating equations to obtain $\hat{\beta}$ an estimate of β . If one were to make the usual assumptions of a generalized linear model, namely independent and identically distributed observations and correct specification of the mean and variance models, an estimate of the asymptotic variance of $\hat{\beta}$ would be the standard model-based variance estimator. This is available in the package by specifying the option `type = "naive"` in `vcov`, but it is not recommended. Since the P_i are only approximately independent and identically distributed, Andersen *et al.* (2003) suggested instead using the robust sandwich variance estimator. The sandwich variance is the default that we use in the package (`type = "robust"` in `vcov`), by using the implementation available in `vcovHC` function of the **sandwich** package (Zeileis 2004, 2006). In our simulation study, it was the clear superior approach. The option `type = "cluster"` uses the cluster-robust variance estimator of `vcovCL`, also in **sandwich**. With multiple time points, the robust variance estimator as implemented in **geepack** (Halekoh *et al.* 2006) is returned with the option `type = "robust"`.

Jacobsen and Martinussen (2016) argue that the remainder term in the Taylor series expansion used to justify the sandwich variance estimate does not converge to 0 quickly enough, and Overgaard *et al.* (2017) derived a variance estimator that accounts for this remainder for the cumulative incidence. Simulation studies therein showed that the Huber-White variance tends to be conservative and that small gains can be made by using Overgaard's corrected variance estimator for the cumulative incidence outcome. Overgaard *et al.* (2019) further developed similar theory for the inverse probability of censoring weighted estimators. The variance expressions are quite complex and will not be reproduced here, but they are implemented in the package (using `type = "corrected"` in `vcov`). This option is available for the cumulative incidence and survival for a single time point only.

The nonparametric bootstrap can also be used to estimate the variance of $\hat{\beta}$ in which the pseudo-observations are recalculated for each bootstrap subsample (Efron 1992). We do not directly implement a bootstrap method for variance estimation, as existing tools in R can be used for that purpose, which we demonstrate in the example.

4. Main package functions and properties

The primary user-facing functions of the **eventglm** package are `cumincglm` and `rmeanglm`. These are designed to be analogous to the `stats::glm` function, but for the cumulative incidence outcome or restricted mean outcome, respectively. A minimal call to either `cumincglm` or `rmeanglm` requires three arguments: `formula`, `time`, and `data`. The left hand side of the formula must be a call to `Surv`, which specifies a possibly right censored time-to-event outcome, with or without competing risks. Currently only right censoring is supported (not interval censoring nor left truncation), which means there can be only a single time variable provided to `Surv`. The `Surv` function is imported from **survival** and re-exported by **eventglm** for convenience's sake (so users do not have to use `library("survival")` or `survival::Surv`). Without competing risks, the event indicator in `Surv` will normally be 0 for censored and 1 for the event. With competing risks, the event indicator should be a factor whose first level indicates censoring, and the other levels indicating the possible event types. In the competing risks case the `cause` argument is also required, which specifies the failure type of interest as

the factor level either as an integer or character value. In the absence of competing risks, the `cumincglm` has the option to specify `survival = TRUE`, which provides a model for the survival (one minus the cumulative incidence).

The `time` argument may be a vector of times for `cumincglm` and must be a single numeric value for `rmeanglm`, which specifies the time(s) t^* at which it is of interest to model the cumulative incidence, survival, restricted mean, or expected lifetime lost. The times must be less than or equal to the largest observed event time in the sample. By default, when `time` is a vector, the model allows time varying intercepts but it is assumed that all covariate effects are not time varying. We provide the special term `tve()` that can be used in the right side of the formula to indicate that the covariate wrapped inside the term is assumed to be time varying. This is illustrated in the example below.

The `data` argument should be a data frame in which the variables specified in the formula can be found. The `link` argument determines the link function g in our notation, which is identity by default, and any value that is supported by the `stats::quasi` family can be used here.

Covariate dependent censoring can be handled using the argument `model.censoring`, which is "independent" by default, assuming completely independent censoring. Alternatives are "stratified", "coxph", or "aareg", and each of these three options require a specification of the relationship between censoring and covariates in the `formula.censoring` argument. If `formula.censoring` is left unspecified, the right hand side of the main formula is used, otherwise a one sided formula can be specified with the implicit outcome of the censoring time. Only categorical covariates may be specified with the "stratified" option.

Since the modeling framework is based on `glm`, all modeling features such as splines, quadratic terms, interactions, and contrasts that can be used in `glm` can be used in the `eventglm` versions by specifying them as usual on the right side of the relevant formulas. This is true for both the main formula and the formula for censoring. The remaining arguments are passed on to `glm.fit`, and are used in the same way here. A noteworthy argument is `weights`, which can be used to specify prior weights for the observations. These can be used to specify inverse probability of missingness weights, propensity scores weights for causal inference, or sampling weights. We illustrate the use of sampling weights for case-cohort sampling in the data analysis example.

In addition to the standard methods `print` and `summary` that detail the model fit, we provide many post-estimation features in correspondence with 'glm' objects. For example `vcov`, `confint` are used for inference with the argument `type` that determines the type of variance calculation ("robust" by default). Furthermore, `predict`, and `residuals` can be used for prediction of individual values and model checking of various kinds. The residuals in the cumulative incidence model are scaled by default according to the recommendations of [Perme and Andersen \(2008\)](#):

$$\hat{\varepsilon}_i = \frac{\hat{E}(V_i) - \hat{Y}_i}{\sqrt{\hat{Y}_i(1 - \hat{Y}_i)}}.$$

The objects returned by `cumincglm` and `rmeanglm` inherit the classes 'pseudoglm', 'glm', and 'lm', so in addition to the methods we define, many more are available using existing infrastructure. The `y` element of the objects of class 'pseudoglm' returned by these functions contains the pseudo-observations and these can be used for other purposes without recalculating them again, such as estimating relative survival ([Pavlič and Pohar Perme 2019](#)).

5. Data analysis examples

```
R> library("survival")
R> library("eventglm")
```

The **eventglm** package includes two example datasets:

- **colon**: Data from a clinical trial of adjuvant chemotherapy for colon cancer. Levamisole is a low-toxicity compound and 5-FU is a moderately toxic chemotherapy agent. There are only one record per patient that includes the time to death (or censoring). This is redistributed from the **survival** package, with a small modification to include only the death outcome.
- **mgus2**: Observational data from 1341 patients with monoclonal gammopathy of undetermined significance (MGUS). The outcome of interest is the time to plasma cell malignancy (PCM), with death as a competing risk, and censoring at the last month of contact. This dataset is redistributed from the **survival** package with an added competing risks event indicator.

To illustrate the basic concepts, Figure 1 shows the nonparametrically estimated survival quantities under consideration for the two datasets: the Kaplan-Meier survival curves for each treatment group in the colon cancer study, and the Aalen-Johansen estimates of the cumulative incidence for PCM and death. The vertical dotted line indicates the times of interest, with the open circles at the probabilities at that time, and the shaded areas indicate the restricted mean survival (**colon**) and the expected lifetime lost (**mgus2**) up to the times of interest.

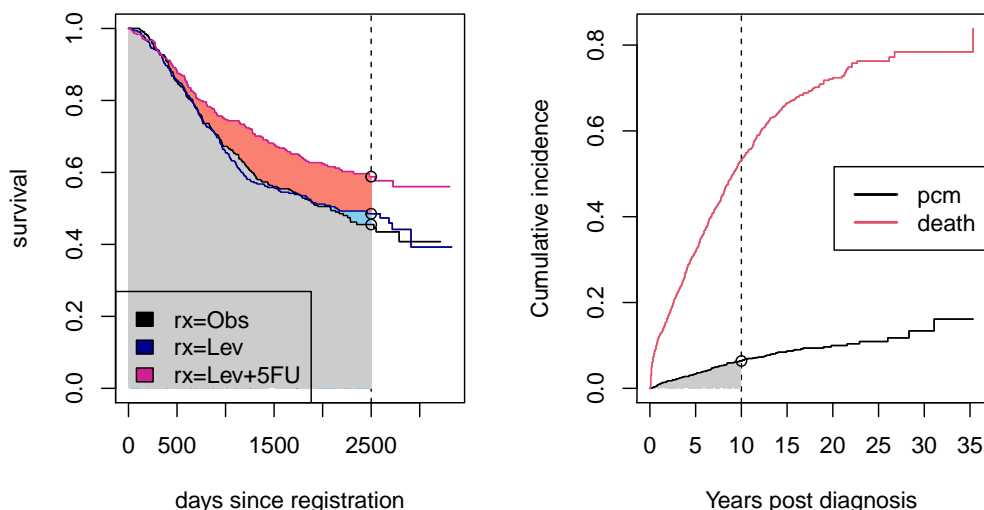


Figure 1: Marginal survival quantities of interest for the colon cancer dataset (left panel) and the MGUS dataset (right panel).

5.1. Overall survival in colon cancer

We can now do inference on the cumulative incidence of death and the restricted mean survival in the colon dataset using the `eventglm` package and the main functions that do the model fitting: `cumincglm` and `rmeanglm`. These functions resemble the `glm` function, with two key differences: the outcome is a call to `Surv`, and there is an argument `time` that specifies the fixed time point at which the cumulative incidence or restricted mean is of interest. First, we fit a regression model for the cumulative incidence, or one minus survival:

```
R> colon.cifit <- cumincglm(Surv(time, status) ~ rx,
+   time = 2500, data = colon)
R> summary(colon.cifit)
```

Call:

```
cumincglm(formula = Surv(time, status) ~ rx, time = 2500, data = colon)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.5875	-0.4902	-0.3467	0.4863	2.1103

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.54345	0.02946	18.449	< 2e-16 ***
rxLev	-0.02907	0.04173	-0.697	0.48596
rxLev+5FU	-0.13176	0.04186	-3.148	0.00165 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasi family taken to be 1)

Null deviance: 253.10 on 928 degrees of freedom
 Residual deviance: 250.15 on 926 degrees of freedom
 AIC: NA

Number of Fisher Scoring iterations: 2

```
R> se.ci <- sqrt(diag(vcov(colon.cifit, type = "robust")))
R> b.ci <- coefficients(colon.cifit)
R> conf.ci <- confint(colon.cifit)
R> round(cbind(b.ci, conf.ci), 2)
```

	b.ci	2.5 %	97.5 %
(Intercept)	0.54	0.49	0.60
rxLev	-0.03	-0.11	0.05
rxLev+5FU	-0.13	-0.21	-0.05

We find that compared to observation alone, the Levamisole alone treatment group has a -0.03 difference in the cumulative incidence of death at 2500 days, with 95% confidence interval

-0.11, 0.05, while the Levamisole plus 5-FU group has a -0.13 difference in the cumulative incidence of death at 2500 days, with 95% confidence interval -0.21, -0.05. This roughly agrees with the Kaplan-Meier estimates from `survfit`:

```
R> colon.smry <- summary(colonsfit, times = 2500, rmean = 2500)
R> cbind(eventglm = b.ci, survfit = c(1 - colon.smry$surv[1],
+   (1 - colon.smry$surv[2:3]) - (1 - rep(colon.smry$surv[1], 2))))
```

	eventglm	survfit
(Intercept)	0.54345139	0.54479221
rxLev	-0.02907499	-0.02990601
rxLev+5FU	-0.13175778	-0.13301654

Unlike `survfit`, it is trivial to perform inference using the `summary` or `confint` methods that we provide for objects of class `'pseudoglm'`. We can fit another model using the log link to obtain estimates of the relative risks comparing the active treatment arms to the observation arm:

```
R> colon.rr <- cumincglm(Surv(time, status) ~ rx, time = 2500,
+   data = colon, link = "log")
R> br.ci <- coefficients(colon.rr)
R> confr.ci <- confint(colon.rr)
R> round(exp(cbind(br.ci, confr.ci)), 2)
```

	br.ci	2.5 %	97.5 %
(Intercept)	0.54	0.49	0.6
rxLev	0.95	0.81	1.1
rxLev+5FU	0.76	0.63	0.9

We find that the estimated probability of death before 2500 days in the Levamisole alone arm is 0.95 times lower compared to observation with 95% confidence interval 0.81, 1.10 and the estimated probability of death before 2500 days in the Levamisole+5FU arm is 0.76 times lower compared to observation with 95% confidence interval 0.63, 0.90.

Now for the restricted mean:

```
R> colon.rmfit <- rmeanglm(Surv(time, status) ~ rx,
+   time = 2500, data = colon)
R> summary(colon.rmfit)
```

Call:

```
rmeanglm(formula = Surv(time, status) ~ rx, time = 2500, data = colon)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1839.4	-903.8	620.9	829.9	848.1

Coefficients:

```

              Estimate Std. Error z value Pr(>|z|)
(Intercept) 1667.403      49.949  33.382 < 2e-16 ***
rxLev        -6.074      71.739  -0.085  0.93253
rxLev+5FU    194.954      70.498   2.765  0.00569 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

(Dispersion parameter for quasi family taken to be 1)

```

Null deviance: 734414066 on 928 degrees of freedom
Residual deviance: 726392934 on 926 degrees of freedom
AIC: NA

```

Number of Fisher Scoring iterations: 2

```

R> se.rm <- sqrt(diag(vcov(colon.rmfit, type = "robust")))
R> b.rm <- coefficients(colon.rmfit)
R> conf.rm <- confint(colon.rmfit)
R> round(cbind(b.rm, conf.rm), 2)

```

```

              b.rm    2.5 %  97.5 %
(Intercept) 1667.40 1569.50 1765.30
rxLev        -6.07 -146.68  134.53
rxLev+5FU    194.95   56.78  333.13

```

We find that compared to observation alone, the Levamisole alone treatment group has a -6.07 difference in the mean time to death up to 2500 days, with 95% confidence interval -146.68, 134.53, while the Levamisole plus 5-FU group has a 194.95 difference in the mean time to death up to 2500 days, with 95% confidence interval 56.78, 333.13. Again, this roughly agrees with the Kaplan-Meier estimates from `survfit`:

```

R> cbind(eventglm = b.rm,
+   survfit = c(colon.smry$table[1, 5],
+   colon.smry$table[2:3, 5] - colon.smry$table[1, 5]))

```

```

              eventglm    survfit
(Intercept) 1667.40308 1666.948078
rxLev        -6.07367  -5.708803
rxLev+5FU    194.95446  195.313754

```

A key advantage of the regression approach is that it gives us the ability to adjust or model other covariates. In this example, since it is a randomized trial, all baseline covariates should be independent of treatment assignment. However, several of these variables are associated with time to death, so they can be used as precision variables. We would expect that adjusting for age, or the number of positive lymph nodes (more than 4) in the above models would reduce the standard error estimates of the treatment effects, without changing the coefficient estimates. Let us find out:

```
R> colon.rm.adj <- rmeanglm(Surv(time, status) ~ rx + age + node4,
+   time = 2500, data = colon)
R> summary(colon.rm.adj)
```

Call:

```
rmeanglm(formula = Surv(time, status) ~ rx + age + node4, time = 2500,
  data = colon)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2035.7	-788.5	443.2	647.6	1385.7

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	2067.576	151.579	13.640	< 2e-16 ***
rxLev	3.421	67.516	0.051	0.95958
rxLev+5FU	185.349	67.365	2.751	0.00593 **
age	-3.735	2.391	-1.562	0.11824
node4	-644.960	64.854	-9.945	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasi family taken to be 1)

Null deviance: 734414066 on 928 degrees of freedom
 Residual deviance: 650039827 on 924 degrees of freedom
 AIC: NA

Number of Fisher Scoring iterations: 2

The estimates of the treatment effects do not notably change and the standard errors are about 5% smaller.

For the cumulative incidence, we can specify the multivariate model by specifying `time` as a vector. Then the output includes the intercept term which corresponds to the smallest time, plus the main effects of each of the times on the intercept.

```
R> colon.mvt <- cumincglm(Surv(time, status) ~ rx + age + node4,
+   time = c(500, 1000, 2500), data = colon)
R> summary(colon.mvt)
```

Call:

```
cumincglm(formula = Surv(time, status) ~ rx + age + node4, time = c(500,
  1000, 2500), data = colon)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.7816	-0.3199	-0.1067	0.3923	1.9567

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.0283668	0.0616262	-0.460	0.64530
factor(pseudo.time)1000	0.1702833	0.0123396	13.800	< 2e-16 ***
factor(pseudo.time)2500	0.3517497	0.0164225	21.419	< 2e-16 ***
rxLev	-0.0095644	0.0282455	-0.339	0.73490
rxLev+5FU	-0.0739035	0.0279946	-2.640	0.00829 **
age	0.0019857	0.0009879	2.010	0.04442 *
node4	0.2767752	0.0277118	9.988	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasi family taken to be 1)

Null deviance: 620.31 on 2786 degrees of freedom
 Residual deviance: 516.91 on 2780 degrees of freedom
 AIC: NA

Number of Fisher Scoring iterations: 2

To allow the covariate effects to vary by time, we can enclose any of the covariates in the right side of the formula in `tve`. For example, we can estimate time varying effects of treatment:

```
R> colon.mvt2 <- cumincglm(Surv(time, status) ~ tve(rx) + age + node4,
+   time = c(500, 1000, 2500), data = colon)
R> colon.mvt2
```

```
Call: cumincglm(formula = Surv(time, status) ~ tve(rx) + age + node4,
  time = c(500, 1000, 2500), data = colon)
```

Model for the identity cumulative incidence at time 500 1000 2500

Coefficients:

(Intercept)	factor(pseudo.time)1000
-0.045210	0.178401
factor(pseudo.time)2500	age
0.394161	0.001986
node4	factor(pseudo.time)500:rxLev
0.276775	-0.008465
factor(pseudo.time)1000:rxLev	factor(pseudo.time)2500:rxLev
0.013173	-0.033402
factor(pseudo.time)500:rxLev+5FU	factor(pseudo.time)1000:rxLev+5FU
-0.023553	-0.070427
factor(pseudo.time)2500:rxLev+5FU	
-0.127730	

Degrees of Freedom: 2786 Total (i.e. Null); 2776 Residual

In the above output, in addition to the intercept term plus the main effects of each of the times on the intercept, for each covariate wrapped in the special term `tve`, there is the interaction between each of the time points and that covariate. Thus, for example, the coefficient labelled `factor(pseudo.time)500:rxLev` is interpreted as the risk difference comparing the Levamisole along group to the Observation group at 500 days, adjusting for the other covariates in a time constant manner. In this model, we observe that the effect on survival, on the risk difference scale, of Levamisole plus 5-FU gets larger in magnitude over time, similar to what we can see in the Kaplan-Meier curves.

5.2. Modeling censoring

By default, we assume that time to censoring is independent of the time to the event, and of all covariates in the model. This is more restrictive than parametric survival models, or Cox regression, which only assumes that censoring time is conditionally independent of event time given the covariates in the model. We provide several options to relax that assumption using the `model.censoring` and `formula.censoring` options. The first is to compute pseudo-observations stratified on a set of categorical covariates, which assumes that the censoring is independent given a set of categorical covariates:

```
R> colon.ci.cen1 <- cumincglm(Surv(time, status) ~ rx + age + node4,
+   time = 2500, data = colon, model.censoring = "stratified",
+   formula.censoring = ~ rx)
```

Next, we can assume that the time to censoring follows a Cox model given a set of covariates. By default, the same covariate formula (right hand side) as the main model is used, but any formula can be specified. We can also use Aalen's additive hazards model instead of a Cox model for the censoring distribution. Then IPCW pseudo-observations are used ([Overgaard et al. 2019](#)). The two different weighting options ("`binder`", the default or "`hajek`") can be specified with the `ipcw.method` option.

```
R> colon.ci.adj <- cumincglm(Surv(time, status) ~ rx + age + node4,
+   time = 2500, data = colon, model.censoring = "independent",
+   formula.censoring = ~ rx + age + node4)
R> colon.ci.cen2 <- cumincglm(Surv(time, status) ~ rx + age + node4,
+   time = 2500, data = colon, model.censoring = "coxph",
+   formula.censoring = ~ rx + age + node4)
R> colon.ci.cen3 <- cumincglm(Surv(time, status) ~ rx + age + node4,
+   time = 2500, data = colon, model.censoring = "aareg",
+   formula.censoring = ~ rx + age + node4)
R> colon.ci.cen2h <- cumincglm(Surv(time, status) ~ rx + age + node4,
+   time = 2500, data = colon, model.censoring = "coxph",
+   formula.censoring = ~ rx + age + node4,
+   ipcw.method = "hajek")
R> colon.ci.cen3h <- cumincglm(Surv(time, status) ~ rx + age + node4,
+   time = 2500, data = colon, model.censoring = "aareg",
+   formula.censoring = ~ rx + age + node4,
+   ipcw.method = "hajek")
R> round(cbind("indep" = coef(colon.ci.adj),
```

```
+ "strat" = coef(colon.ci.cen1),
+ "coxipcw" = coef(colon.ci.cen2),
+ "aalenipcw" = coef(colon.ci.cen3),
+ "coxipcw.hajek" = coef(colon.ci.cen2h),
+ "aalenipcw.hajek" = coef(colon.ci.cen3h)), 3)
```

	indep	strat	coxipcw	aalenipcw	coxipcw.hajek	aalenipcw.hajek
(Intercept)	0.318	0.314	0.535	0.596	0.297	0.317
rxLev	-0.034	-0.035	-0.034	-0.036	-0.031	-0.036
rxLev+5FU	-0.127	-0.128	-0.127	-0.127	-0.110	-0.129
age	0.002	0.002	0.002	0.002	0.003	0.002
node4	0.332	0.334	0.335	0.334	0.330	0.335

The model objects include the estimated weights in the element called `ipcw.weights`. It is recommended to inspect the distribution of these weights in case of issues in estimation that may be caused by extreme values of the estimated weights.

```
R> summary(colon.ci.cen2$ipcw.weights)
```

```
      V1
Min.   :0.2702
1st Qu.:0.4832
Median :0.9094
Mean   :0.7680
3rd Qu.:0.9988
Max.   :1.0000
```

5.3. Competing risks in plasma cell malignancy

The package works very similarly when there are competing risks. The key differences are that the event indicator in `Surv` is a factor with more than 2 levels and that the `cause` option is used to specify the cause of interest. The MGUS dataset has a number of covariates, and the time until progression to PCM, or death. Here the event PCM is of primary interest, with death being a competing event. We can get similar estimates to the marginal Aalen-Johansen estimates for the cumulative incidence of PCM at 10 years and the expected lifetime lost due to PCM up to 10 years with similar commands as above.

```
R> cumincglm(Surv(etime, event) ~ sex,
+   cause = "pcm", time = 120, data = mgus2)
```

```
Call: cumincglm(formula = Surv(etime, event) ~ sex, time = 120,
  cause = "pcm", data = mgus2)
```

Model for the identity cumulative incidence of cause pcm at time 120

Coefficients:

```
(Intercept)      sexM
      0.07383      -0.01857
```

Degrees of Freedom: 1383 Total (i.e. Null); 1382 Residual

```
R> mgfit1 <- rmeanglm(Surv(etime, event) ~ sex,
+   cause = "pcm", time = 120, data = mgus2)
R> mgfit1
R> plot(mgfit1)
```

```
Call: rmeanglm(formula = Surv(etime, event) ~ sex, time = 120,
  cause = "pcm", data = mgus2)
```

Model for the identity restricted mean time lost due to cause pcm at time 120

Coefficients:

```
(Intercept)      sexM
      4.793      -1.293
```

Degrees of Freedom: 1383 Total (i.e. Null); 1382 Residual

Including other covariates in the model is done using the standard formula interface. Inspection of the diagnostic plots in Figure 2 reveals that a more complex model may be appropriate.

```
R> mgfitrmean <- rmeanglm(Surv(etime, event) ~ sex * age + hgb + I(hgb^2),
+   cause = "pcm", time = 120, data = mgus2)
R> summary(mgfitrmean)
```

Call:

```
rmeanglm(formula = Surv(etime, event) ~ sex * age + hgb + I(hgb^2),
  time = 120, cause = "pcm", data = mgus2)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-7.050	-4.749	-3.947	-3.258	112.814

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.21179	19.75007	0.011	0.991
sexM	1.34720	4.31629	0.312	0.755
age	0.06192	0.04760	1.301	0.193
hgb	0.23018	3.28708	0.070	0.944
I(hgb^2)	-0.01679	0.12814	-0.131	0.896
sexM:age	-0.03255	0.06260	-0.520	0.603

(Dispersion parameter for quasi family taken to be 1)

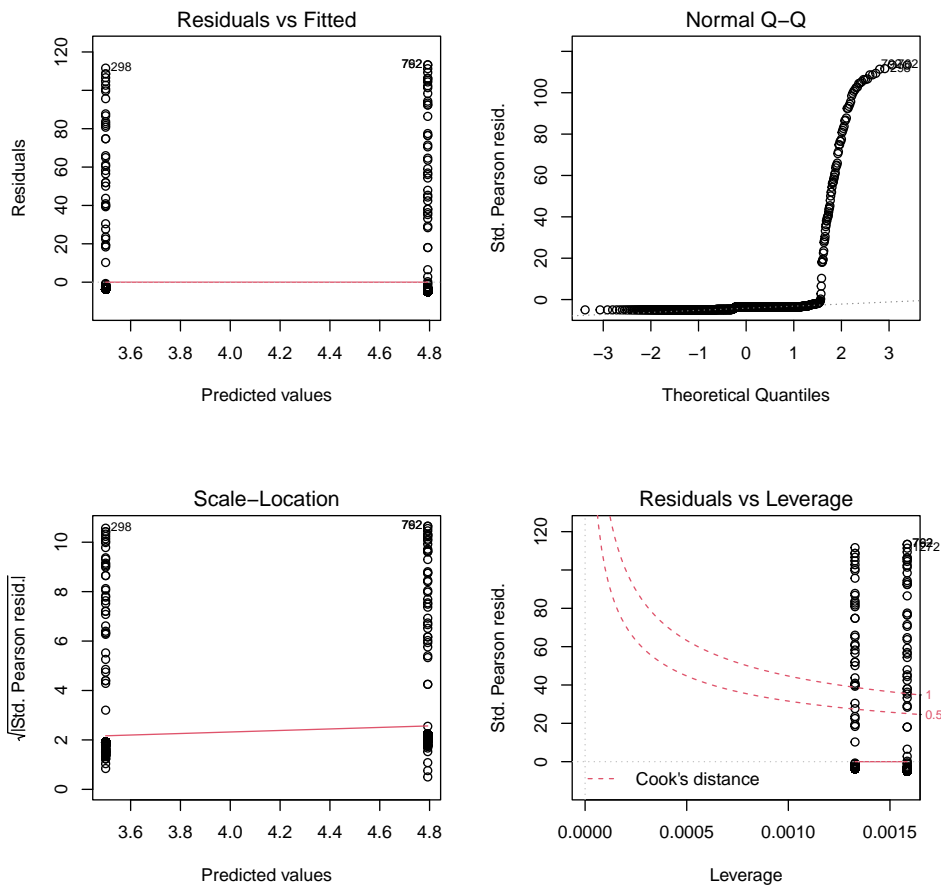


Figure 2: Regression diagnostic plots in the MGUS example.

```

Null deviance: 448531  on 1370  degrees of freedom
Residual deviance: 447259  on 1365  degrees of freedom
(13 observations deleted due to missingness)
AIC: NA

```

```

Number of Fisher Scoring iterations: 2

```

The `vcov` function has several options for calculation of the estimated variance of the estimated regression parameters using the `type` argument. By default, the robust variance estimates are used (`type="robust"`), based on the Huber-White estimator. Other options are `"naive"`, and `"corrected"`, where `corrected` refers to the variance estimators suggested by [Overgaard *et al.* \(2017\)](#) which are based on a second order Von-Mises expansion. We can also use the bootstrap. This recalculates the pseudo-observations every time, but it is still quite fast because of the C code underlying the computation. Let us compare:

```

R> nboot <- 1000
R> bootests <- matrix(numeric(nboot * 4), nrow = nboot, ncol = 4)
R> for(i in 1:nboot) {
+   mgus.b <- mgus2[sample(1:nrow(mgus2), replace = TRUE), ]

```

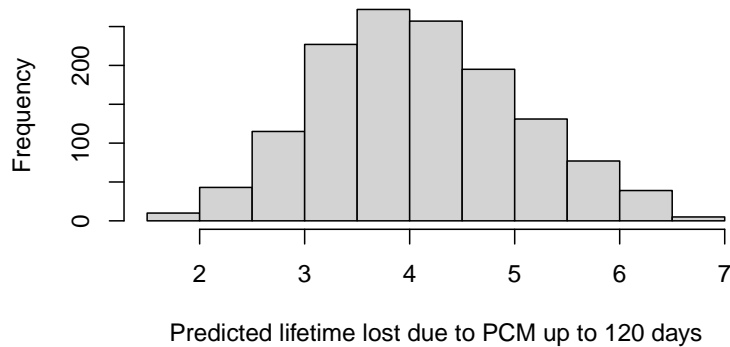


Figure 3: Histogram showing the distribution of predicted lifetimes lost due to PCM up to 120 days in the MGUS example.

```

+   mgfitrmean.b <- rmeanglm(Surv(etime, event) ~ sex + age + hgb,
+     cause = "pcm", time = 120, data = mgus.b)
+   bootests[i,] <- coefficients(mgfitrmean.b)
+ }
R> mgfitrmean2 <- rmeanglm(Surv(etime, event) ~ sex + age + hgb,
+   cause = "pcm", time = 120, data = mgus2)
R> se.boot <- sqrt(diag(cov(bootests)))
R> round(cbind(se.boot = se.boot,
+   se.robust = sqrt(diag(vcov(mgfitrmean2))),
+   se.naive = sqrt(diag(vcov(mgfitrmean2, type = "naive")))), 3)

```

	se.boot	se.robust	se.naive
(Intercept)	3.979	3.874	4.749
sexM	0.994	0.992	1.012
age	0.029	0.029	0.041
hgb	0.250	0.251	0.253

The corrected estimator does not handle ties, and so is not presented for this example.

Predicted restricted means give a possible method to predict individual event times, while the predicted cumulative incidence should be probabilities. Note that with the identity and log links, the predicted cumulative incidences are not guaranteed to be between 0 and 1.

```

R> hist(predict(mgfitrmean, newdata = mgus2),
+   xlab = "Predicted lifetime lost due to PCM up to 120 days",
+   main = "")

```

5.4. Case cohort sampling

Parner, Andersen, and Overgaard (2020) describe how to fit regression models with pseudo-observations that account for case-cohort sampling. The basic idea is weighted estimating

equations, which we can implement easily with the `weights` argument that gets passed to `glm.fit`. First let us create a case-cohort sample of the MGUS dataset by sampling the malignancy events with probability 0.9, and a random subcohort with probability 0.2.

```
R> set.seed(918)
R> subc <- rbinom(nrow(mgus2), size = 1, prob = 0.2)
R> samp.ind <- subc + (1 - subc) * (mgus2$event == "pcm") *
+   rbinom(nrow(mgus2), size = 1, prob = 0.9)
R> mgus2.cc <- mgus2[as.logical(samp.ind), ]
R> mgus2.cc$samp.wt <- 1 / ifelse(mgus2.cc$event == "pcm",
+   0.2 + 0.8 * 0.9, 0.2)
```

Now, the weighted regression model should give similar results as the unweighted one in the full sample:

```
R> mgfit.cc <- rmeanglm(Surv(etime, event) ~ I(age - 65) + sex + hgb,
+   cause = "pcm", time = 120, data = mgus2.cc,
+   weights = samp.wt)
R> mgfit.full <- rmeanglm(Surv(etime, event) ~ I(age - 65) + sex + hgb,
+   cause = "pcm", time = 120, data = mgus2)
R> mdf <- data.frame(casecohort = summary(mgfit.cc)$coefficients[,1:2],
+   fullsamp = summary(mgfit.full)$coefficients[, 1:2])
R> colnames(mdf) <- c("case cohort Est", "SE", "full Est", "SE")
R> round(mdf, 3)
```

	case cohort Est	SE	full Est	SE
(Intercept)	6.343	3.803	6.886	3.431
I(age - 65)	0.044	0.037	0.045	0.029
sexM	-1.053	1.117	-0.990	0.992
hgb	-0.147	0.277	-0.188	0.251

6. Numerical studies

6.1. Simulation study of statistical properties

We conducted a simulation study with the goal of determining which methods should be used as the defaults in our package. The key criteria are validity, as measured by type I error rates, bias, and confidence interval coverage, robustness to misspecification of the censoring mechanism, and statistical efficiency. Detailed descriptions of the simulation setup and results are in the Appendix, and code available in the replication script.

According to our simulation study, the stratified option works quite well even when the censoring model is misspecified, and the Aalen additive model tends to work better than the Cox model. Even when the censoring models are misspecified, either by omitting covariates or incorrectly assuming proportional hazards, some form of adjustment for covariate dependent censoring is an improvement over assuming completely independent censoring. There were no

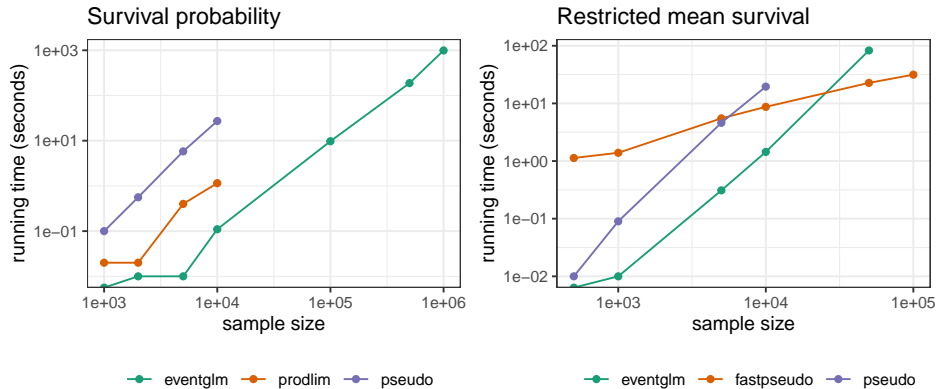


Figure 4: Running time comparison of the calculation of pseudo-observations for the survival probability at a fixed time (left panel), and the restricted mean survival (right panel).

clear differences in terms of bias comparing the Binder versus Hajek weighting approaches. The robust variance estimator (the sandwich variance as implemented in the **sandwich** package) is the clearly superior approach for inference, with minimal bias and approximately correct confidence interval coverage in all cases. Overgaard’s corrected variance estimator has only marginal benefits over the robust estimator in a few cases.

6.2. Speed and memory comparison

Pseudo-observations can also be computed using the packages **pseudo**, **prodlim** (survival and cumulative incidence only), and **fastpseudo** (restricted mean only). The **pseudo** package does the job, but is not optimized for speed or memory usage. The **prodlim** approach is optimized in C code, but cannot handle large datasets because it stores the jackknife values for every observed event time. We have optimized this code further in **eventglm** so that it only stores the jackknife values for the time of interest, thus it can be used for much larger datasets. We compare the speed of computing the pseudo-observations for the survival curve at a fixed time in Figure 4. These timing calculations were done on a laptop with an 9th generation Intel core i7 processor, and 8gb of RAM. Neither **pseudo** nor **prodlim** are able to handle a dataset with 100,000 observations, while **eventglm** can go at least an order of magnitude larger and in a reasonable amount of time.

The **fastpseudo** package uses only base R to efficiently compute pseudo-observations for the restricted mean survival, but does not handle competing risks. Upon inspection of the code during testing, it is clear that it also can only handle integer observation times, which is something that is not clearly documented in the package. Since the restricted mean does require computing the survival curve at all times less than the time of interest, the default method in **eventglm** has the same limitations as **prodlim**. However, the IPCW method only requires fitting a regression model for the time to censoring once, and then simply computing means, and thus can be applied to much larger datasets. The stratified option can also be used to improve computational efficiency as computing several sets of pseudo-observations on partitions of the data can be faster and a better use of memory than doing it once for a large dataset.

7. Extending eventglm

As of version 1.1.0, the argument `model.censoring` of `cumincglm` and `rmeanglm` refers to a function. This function is the workhorse that does the computation of the pseudo-observations that are later used in the generalized linear model. A number of computation methods are built in as “modules” that are contained in the source file called “pseudo-modules.R”. As an example, consider the independent module:

```
R> eventglm::pseudo_independent
function(formula, time, cause = 1, data,
  type = c("cuminc", "survival", "rmean"),
  formula.censoring = NULL, ipcw.method = NULL) {
  margformula <- update.formula(formula, . ~ 1)
  mr <- model.response(model.frame(margformula, data = data))
  stopifnot(attr(mr, "type") %in% c("right", "mright"))
  marginal.estimate <- survival::survfit(margformula, data = data)
  if(type == "cuminc") {
    POi <- get_pseudo_cuminc(marginal.estimate, time, cause, mr)
  } else if(type == "survival") {
    if(marginal.estimate$type != "right") {
      stop("Survival estimand not available for outcome with
        censoring type", marginal.estimate$type)
    }
    POi <- 1 - get_pseudo_cuminc(marginal.estimate, time, cause, mr)
  } else if(type == "rmean") {
    POi <- get_pseudo_rmean(marginal.estimate, time, cause, mr)
  }
  POi
}
<bytecode: 0x0000017d922fa750>
<environment: namespace:eventglm>
```

This function, and any pseudo-observation module, must take the same named arguments (though they do not all have to be used), and return a vector of pseudo-observations. Users can specify their own functions directly, or by name. Our built in modules all have the prefix `pseudo_`, and so if a name is given rather than a function, we search for functions with this prefix first, and if not found, without the prefix.

7.1. Example: Parametric pseudo-observations

Let us see how to define a custom function for computation of pseudo-observations. In this first example, we will fit a parametric Weibull survival model with `survreg` marginally and do jackknife leave-one-out estimates. This may be useful if there is interval censoring, for example.

```
R> pseudo_parametric <- function(formula, time, cause = 1, data,
+   type = c("cuminc", "survival", "rmean"),
+   formula.censoring = NULL, ipcw.method = NULL) {
```

```

+   margformula <- update.formula(formula, . ~ 1)
+   mr <- model.response(model.frame(margformula, data = data))
+   marginal.estimate <- survival::survreg(margformula, data = data,
+     dist = "weibull")
+   theta <- pweibull(time, shape = 1 / marginal.estimate$scale,
+     scale = exp(marginal.estimate$coefficients[1]))
+   theta.i <- sapply(1:nrow(data), function(i) {
+     me <- survival::survreg(margformula, data = data[-i, ],
+       dist = "weibull")
+     pweibull(time, shape = 1 / me$scale,
+       scale = exp(me$coefficients[1]))
+   })
+   POi <- theta + (nrow(data) - 1) * (theta - theta.i)
+   POi
+ }

```

Now let us try it out by passing it to the `cumincglm` function and compare to the default independence estimator:

```

R> fitpara <- cumincglm(Surv(time, status) ~ rx + sex + age, time = 2500,
+   model.censoring = pseudo_parametric,
+   data = colon)
R> fitdef <- cumincglm(Surv(time, status) ~ rx + sex + age, time = 2500,
+   model.censoring = "independent",
+   data = colon)
R> sapply(list(parametric = fitpara, default = fitdef),
+   coefficients)

```

	parametric	default
(Intercept)	0.5473823439	0.489105540
rxLev	-0.0216382248	-0.029287251
rxLev+5FU	-0.1488141565	-0.132651617
sex	0.0008128962	-0.010226326
age	0.0004232579	0.001004726

You can also refer to the function with a string, omitting the "pseudo_" prefix, if you wish, e.g.,

```

R> fitpara <- cumincglm(Surv(time, status) ~ rx + sex + age, time = 2500,
+   model.censoring = "parametric",
+   data = colon)

```

7.2. Example 2: Infinitesimal jackknife

When the `survival` package version 3.0 was released, it became possible to get the influence function values returned from `survfit` estimation functions. These efficient influence functions are used in the variance calculations, and they are related to pseudo-observations.

More information is available in the “Pseudo-values” vignette of **survival**, which is under development at the time of writing. We can use this feature to create a custom function for infinitesimal jackknife pseudo-observations:

```
R> pseudo_infjack <- function(formula, time, cause = 1, data,
+   type = c("cuminc", "survival", "rmean"),
+   formula.censoring = NULL, ipcw.method = NULL) {
+   marge <- survival::survfit(update.formula(formula, . ~ 1),
+     data = data, influence = TRUE)
+   tdex <- sapply(time, function(x) max(which(marge$time <= x)))
+   pstate <- marge$surv[tdex]
+   POi <- matrix(pstate, nrow = marge$n,
+     ncol = length(time), byrow = TRUE) +
+     (marge$n) * (marge$influence.surv[, tdex + 1])
+   POi
+ }
```

Note that this computes pseudo-observations for survival, rather than the cumulative incidence, so to compare we can use the `survival = TRUE` option. Now we try it out

```
R> fitinf <- cumincglm(Surv(time, status) ~ rx + sex + age, time = 2500,
+   model.censoring = "infjack",
+   data = colon)
R> fitdefsurv <- cumincglm(Surv(time, status) ~ rx + sex + age,
+   time = 2500, survival = TRUE, data = colon)
R> sapply(list(infjack = fitinf, default = fitdefsurv),
+   coefficients)
```

	infjack	default
(Intercept)	0.510826426	0.510894460
rxLev	0.029260880	0.029287251
rxLev+5FU	0.132636051	0.132651617
sex	0.010256818	0.010226326
age	-0.001003621	-0.001004726

8. Conclusion

Using the pseudo-observation approach, in comparison to Cox regression or fully parametric regression, can directly parametrize associations of interest between covariates and cumulative summaries of survival. This provides valid inference under similar assumptions as the Cox model, but easier interpretation of resulting coefficients, particularly when one is interested in causal effects. Given the advantages of the pseudo-observation approach, it is not surprising that there has been a great deal of development of statistical methods surrounding the estimation and inference based on them. However, we believe the barrier to this approach becoming as common as Cox regression is the lack of easy implementation. Our package enables the

use of these methods with a user-friendly interface that will be familiar to even a beginning R user but, by leveraging existing infrastructure, allows for the flexibility and options advanced R users are expecting. For example, the objects returned by `cumincglm` and `rmeanglm` inherit from `'pseudoglm'`, `'glm'`, and `'lm'`, so in addition to the methods we define, many more are available using existing infrastructure available in such packages as `stats`, `broom`, `splines`, and many more (R Core Team 2021; Robinson, Hayes, and Couch 2022). Our hope is that pseudo-observation based survival regression will become as common as Cox models.

8.1. Future work

A GEE approach that allows for borrowing information across multiple time points was actually suggested initially (Andersen *et al.* 2003), although it was found that the robust standard errors used in GEE are not exactly correct Overgaard *et al.* (2017). Our future goal is to implement the standard error calculations that correctly account for both the correlated data and the pseudo-observation calculation when there are multiple time points. The key challenge is to design the interface with the appropriate balance of usability, understanding and flexibility.

In addition to new features such as goodness of fit statistics based on cumulative residuals as described by Pavlič, Martinussen, and Andersen (2019), we also plan to extend to additional estimands like unrestricted lifetime and the probability of being in state in more general multi-state model settings, allowing for left-truncation or delayed entry. To this end, the modular approach we described in Section 7 allows the user to specify their own `model.censoring` function that takes as input the design matrix and outputs the vector of pseudo-observations that are used in the subsequent models. This allows further development and implementation of new methods in this area, such as the use of the infinitesimal jackknife and the use of flexible parametric models for interval censoring. This opens up a lot of possibilities for future extensions of `eventglm`, and will also make it easier to maintain.

Acknowledgments

This work was supported by the Swedish Research Council, grant numbers 2017-01898 and 2019-00227. The authors are thankful for the discussions with Mark Clements, Paul Lambert, and Arvid Sjölander, who helped improve the paper.

References

- Aalen OO (1989). “A Linear Regression Model for the Analysis of Life Times.” *Statistics in Medicine*, **8**(8), 907–925. doi:10.1002/sim.4780080803.
- Aalen OO, Cook RJ, Røysland K (2015). “Does Cox Analysis of a Randomized Survival Study Yield a Causal Treatment Effect?” *Lifetime Data Analysis*, **21**(4), 579–593. doi:10.1007/s10985-015-9335-y.
- Aalen OO, Johansen S (1978). “An Empirical Transition Matrix for Non-Homogeneous Markov Chains Based on Censored Observations.” *Scandinavian Journal of Statistics*, pp. 141–150.

- Allignol A (2018). **Cprob**: *The Conditional Probability Function of a Competing Event*. R package version 1.4.1, URL <https://CRAN.R-project.org/package=Cprob>.
- Allignol A, Latouche A, Yan J, Fine JP (2011). “A Regression Model for the Conditional Probability of a Competing Event: Application to Monoclonal Gammopathy of Unknown Significance.” *Journal of the Royal Statistical Society C*, **60**(1), 135–142. doi:10.1111/j.1467-9876.2010.00729.x.
- Andersen PK (2013). “Decomposition of Number of Life Years Lost According to Causes of Death.” *Statistics in Medicine*, **32**(30), 5278–5285. doi:10.1002/sim.5903.
- Andersen PK, Klein JP, Rosthøj S (2003). “Generalised Linear Models for Correlated Pseudo-Observations, with Applications to Multi-State Models.” *Biometrika*, **90**(1), 15–27. doi:10.1093/biomet/90.1.15.
- Andersen PK, Pohar Perme M (2010). “Pseudo-Observations in Survival Analysis.” *Statistical Methods in Medical Research*, **19**(1), 71–99. ISSN 0962-2802, 1477-0334. doi:10.1177/0962280209105020.
- Andersen PK, Syriopoulou E, Parner ET (2017). “Causal Inference in Survival Analysis Using Pseudo-Observations.” *Statistics in Medicine*, **36**(17), 2669–2681. doi:10.1002/sim.7297.
- Austin PC, Fine JP (2017). “Practical Recommendations for Reporting Fine-Gray Model Analyses for Competing Risk Data.” *Statistics in Medicine*, **36**(27), 4391–4400. doi:10.1002/sim.7501.
- Batten D (2015). **fastpseudo**: *Fast Pseudo Observations*. R package version 0.1, URL <https://CRAN.R-project.org/package=fastpseudo>.
- Beyersmann J, Latouche A, Buchholz A, Schumacher M (2009). “Simulating Competing Risks Data in Survival Analysis.” *Statistics in Medicine*, **28**(6), 956–971. doi:10.1002/sim.3516.
- Binder N, Gerds TA, Andersen PK (2014). “Pseudo-Observations for Competing Risks with Covariate Dependent Censoring.” *Lifetime Data Analysis*, **20**(2), 303–315. doi:10.1007/s10985-013-9247-7.
- Cox DR (1972). “Regression Models and Life-Tables.” *Journal of the Royal Statistical Society B*, **34**(2), 187–202. doi:10.1111/j.2517-6161.1972.tb00899.x.
- Daniel R, Zhang J, Farewell D (2021). “Making Apples from Oranges: Comparing Non-collapsible Effect Estimators and Their Standard Errors After Adjustment for Different Covariate Sets.” *Biometrical Journal*, **63**(3), 528–557. doi:10.1002/bimj.201900297.
- Efron B (1992). “Bootstrap Methods: Another Look at the Jackknife.” In *Breakthroughs in Statistics*, pp. 569–593. Springer-Verlag.
- 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. doi:10.1080/01621459.1999.10474144.
- Gerds TA (2019). **prodlim**: *Product-Limit Estimation for Censored Event History Analysis*. R package version 2019.11.13, URL <https://CRAN.R-project.org/package=prodlim>.

- Gerds TA, Kattan MW (2021). *Medical Risk Prediction Models: With Ties to Machine Learning*. Chapman & Hall/CRC. doi:10.1201/9781138384484.
- Graw F, Gerds TA, Schumacher M (2009). “On Pseudo-Values for Regression Analysis in Competing Risks Models.” *Lifetime Data Analysis*, **15**(2), 241–255. doi:10.1007/s10985-008-9107-z.
- Hájek J (1971). “Comment on a Paper by D. Basu.” In GVS D (ed.), *Foundations of Statistical Inference*, p. 242. Holt, Rinehart, Winston, Toronto.
- Halekoh U, Højsgaard S, Yan J (2006). “The R Package **geepack** for Generalized Estimating Equations.” *Journal of Statistical Software*, **15**(2), 1–11. doi:10.18637/jss.v015.i02.
- Jacobsen M, Martinussen T (2016). “A Note on the Large Sample Properties of Estimators Based on Generalized Linear Models for Correlated Pseudo-Observations.” *Scandinavian Journal of Statistics*, **43**(3), 845–862. doi:10.1111/sjos.12212.
- Kaplan EL, Meier P (1958). “Nonparametric Estimation from Incomplete Observations.” *Journal of the American Statistical Association*, **53**(282), 457–481. doi:10.1080/01621459.1958.10501452.
- Klein JP, Gerster M, Andersen PK, Tarima S, Perme MP (2008). “SAS and R Functions to Compute Pseudo-Values for Censored Data Regression.” *Computer Methods and Programs in Biomedicine*, **89**(3), 289–300. doi:10.1016/j.cmpb.2007.11.017.
- Martinussen T, Vansteelandt S, Andersen PK (2020). “Subtleties in the Interpretation of Hazard Contrasts.” *Lifetime Data Analysis*, pp. 1–23. doi:10.1007/s10985-020-09501-5.
- Neuhaus JM, Jewell NP (1993). “A Geometric Approach to Assess Bias Due to Omitted Covariates in Generalized Linear Models.” *Biometrika*, **80**(4), 807–815. doi:10.1093/biomet/80.4.807.
- Nygård Johansen M, Lundbye-Christensen S, Thorlund Parner E (2020). “Regression Models Using Parametric Pseudo-Observations.” *Statistics in Medicine*. doi:10.21203/rs.3.rs-78804/v1.
- Overgaard M, Andersen PK, Parner ET (2015). “Regression Analysis of Censored Data Using Pseudo-Observations: An Update.” *The Stata Journal*, **15**(3), 809–821. doi:10.1177/1536867x1501500313.
- Overgaard M, Parner ET, Pedersen J (2017). “Asymptotic Theory of Generalized Estimating Equations Based on Jack-Knife Pseudo-Observations.” *The Annals of Statistics*, **45**(5), 1988–2015. doi:10.1214/16-aos1516.
- Overgaard M, Parner ET, Pedersen J (2019). “Pseudo-Observations Under Covariate-Dependent Censoring.” *Journal of Statistical Planning and Inference*, **202**, 112–122. doi:10.1016/j.jspi.2019.02.003.
- Parner ET, Andersen PK, Overgaard M (2020). “Cumulative Risk Regression in Case-Cohort Studies Using Pseudo-Observations.” *Lifetime Data Analysis*, pp. 1–20. doi:10.1007/s10985-020-09492-3.

- Pavlič K, Martinussen T, Andersen PK (2019). “Goodness of Fit Tests for Estimating Equations Based on Pseudo-Observations.” *Lifetime Data Analysis*, **25**(2), 189–205. doi: [10.1007/s10985-018-9427-6](https://doi.org/10.1007/s10985-018-9427-6).
- Pavlič K, Pohar Perme M (2019). “Using Pseudo-Observations for Estimation in Relative Survival.” *Biostatistics*, **20**(3), 384–399. doi:[10.1093/biostatistics/kxy008](https://doi.org/10.1093/biostatistics/kxy008).
- Perme MP, Andersen PK (2008). “Checking Hazard Regression Models Using Pseudo-Observations.” *Statistics in Medicine*, **27**(25), 5309–5328. doi:[10.1002/sim.3401](https://doi.org/10.1002/sim.3401).
- Pohar-Perme M, Gerster M (2017). **pseudo**: *Computes Pseudo-Observations for Modeling*. R package version 1.4.3, URL <https://CRAN.R-project.org/package=pseudo>.
- R Core Team (2021). *R: A Language and Environment For Statistical Computing*. R Foundation for Statistical Computing, Vienna. URL <https://www.R-project.org/>.
- Robinson D, Hayes A, Couch S (2022). **broom**: *Convert Statistical Objects into Tidy Tibbles*. R package version 0.8.0, URL <https://CRAN.R-project.org/package=broom>.
- Royston P, Parmar MKB (2002). “Flexible Parametric Proportional-Hazards and Proportional-Odds Models for Censored Survival Data, with Application to Prognostic Modelling and Estimation of Treatment Effects.” *Statistics in Medicine*, **21**(15), 2175–2197. doi:[10.1002/sim.1203](https://doi.org/10.1002/sim.1203).
- Sabathé C, Andersen PK, Helmer C, Gerds TA, Jacqmin-Gadda H, Joly P (2020). “Regression Analysis in an Illness-Death Model with Interval-Censored Data: A Pseudo-Value Approach.” *Statistical Methods in Medical Research*, **29**(3), 752–764. doi: [10.1177/0962280219842271](https://doi.org/10.1177/0962280219842271).
- Sachs MC, Gabriel EE (2022). **eventglm**: *Regression Models for Event History Outcomes*. R package version 1.2.2, URL <https://CRAN.R-project.org/package=eventglm>.
- SAS Institute Inc (2013). *The SAS System, Version 9.4*. SAS Institute Inc., Cary. URL <http://www.sas.com/>.
- Scheike TH, Martinussen T (2006). *Dynamic Regression Models for Survival Data*. Springer-Verlag.
- Scheike TH, Zhang MJ (2011). “Analyzing Competing Risk Data Using the R **timereg** Package.” *Journal of Statistical Software*, **38**(2), 1–15. doi:[10.18637/jss.v038.i02](https://doi.org/10.18637/jss.v038.i02).
- Scheike TH, Zhang MJ, Gerds TA (2008). “Predicting Cumulative Incidence Probability by Direct Binomial Regression.” *Biometrika*, **95**(1), 205–220. doi:[10.1093/biomet/asm096](https://doi.org/10.1093/biomet/asm096).
- Sjölander A, Dahlqvist E, Zetterqvist J (2016). “A Note on the Noncollapsibility of Rate Differences and Rate Ratios.” *Epidemiology*, **27**(3), 356–359. doi:[10.1097/ede.0000000000000433](https://doi.org/10.1097/ede.0000000000000433).
- StataCorp (2019). *Stata Statistical Software: Release 16*. StataCorp LLC, College Station. URL <http://www.stata.com/>.

- Sutradhar R, Austin PC (2018). “Relative Rates Not Relative Risks: Addressing a Widespread Misinterpretation of Hazard Ratios.” *Annals of Epidemiology*, **28**(1), 54–57. doi:10.1016/j.annepidem.2017.10.014.
- Therneau TM (2022). *A Package for Survival Analysis in R*. R package version 3.3-1, URL <https://CRAN.R-project.org/package=survival>.
- Therneau TM, Grambsch PM (2000). *Modeling Survival Data: Extending the Cox Model*. Springer-Verlag, New York. ISBN 0-387-98784-3.
- Tian L, Zhao L, Wei LJ (2014). “Predicting the Restricted Mean Event Time with the Subject’s Baseline Covariates in Survival Analysis.” *Biostatistics*, **15**(2), 222–233. doi:10.1093/biostatistics/kxt050.
- Zeileis A (2004). “Econometric Computing with HC and HAC Covariance Matrix Estimators.” *Journal of Statistical Software*, **11**(10), 1–17. doi:10.18637/jss.v011.i10.
- Zeileis A (2006). “Object-Oriented Computation of Sandwich Estimators.” *Journal of Statistical Software*, **16**(9), 1–16. doi:10.18637/jss.v016.i09.
- Zhao L, Tian L, Claggett B, Pfeffer M, Kim DH, Solomon S, Wei L (2018). “Estimating Treatment Effect With Clinical Interpretation From a Comparative Clinical Trial With an End Point Subject to Competing Risks.” *JAMA Cardiology*, **3**(4), 357–358. doi:10.1001/jamacardio.2018.0127.

A. Simulation study

A.1. Data generation

We generated datasets with competing risks according to [Beyersmann, Latouche, Buchholz, and Schumacher \(2009\)](#) as follows: We first generated a binary covariate Z as Bernoulli with probability 0.3, a normal random variable with mean 4 and standard deviation 4, and a log normal random variable with parameters 0 and 1: X_1, X_2 . Then $\mathbf{Q} = (1, Z, X_1, X_2)$. We used a proportional hazards Weibull distribution to generate the time data for $k = 1, 2$, with a hazard of: $h_k(t \mid \mathbf{Q}) = \gamma_k * (1/e^{(\mathbf{Q}^\top \zeta_k)})^{\gamma_k} * t^{\gamma_k - 1}$ and a cumulative hazard given by: $H_k(t \mid \mathbf{Q}) = (1/e^{(\mathbf{Q}^\top \zeta_k)})^{\gamma_k} * (t)^{\gamma_k}$, where \mathbf{Q} is the vector of all covariates of interest in this order $(1, Z, X_1, X_2)$, which then correspond to the cause specific vector of coefficients $\zeta_k = (\zeta_0, \zeta_z, \zeta_{x1}, \zeta_{x2})$. The overall survivor function of the times to any of the events is then given by: $P(T_{ov} > t \mid \mathbf{Q}) = \exp(-\sum_k H_k(t \mid \mathbf{Q}))$.

We create overall survival times (times to any event) by inverting the CDF, one less the survivor, using the probability integral transform to obtain overall survival times, T_{ov} . We then determine which of the event types a time belongs to by randomly generating from a Bernoulli with probability $h_m(T_{ov} \mid \mathbf{Q}) / (h_m(T_{ov} \mid \mathbf{Q}) + h_{m'}(T_{ov} \mid \mathbf{Q}))$ and assigning event type 1 if 1 and 2 if 0. We then generate Weibull censoring times using the `rweibull` parameterization with shape parameter equal to $e^{\mathbf{Q}^\top \alpha}$ and scale parameter γ_c . The intercept (i.e., first element) of α determines the amount of censoring, and whether the remaining coefficients are non-zero determines whether the censoring depends on covariates. When $\gamma_c = 1$, the censoring times follow a proportional hazards model, and thus the Cox model for the censoring times is correctly specified.

We consider a binary covariate of main interest (with probability 0.3), and two continuous covariates, one normally distributed with mean 4 and variance 1, and the other log normally distributed with parameters 0 and 1. Intercept and shape parameters were determined so that the proportion of observations having the event of interest was approximately equally probable as the competing event before the time of interest and for varying amounts of censoring. We consider 3 different effects of covariates on the outcome of interest. In the null scenario, there is no association of any covariates with the event times. We additionally consider moderate and large effect sizes in combination with small effects of the continuous covariates. We allow for any (or none) of the covariates to be associated with censoring. Specific parameter values are given in the supplementary materials and as a companion R package (`sachsmc/pseudoglm` on GitHub) for running the simulation studies. Within each scenario, we consider different sample sizes, censoring rates, and strength of covariate effects on the censoring time.

For each scenario and simulation replicate, we fit regression models with the cause of interest at a fixed time modeled as a function of the binary covariate of interest, adjusted for the two continuous covariates. We did this for the cumulative incidence and the restricted mean at a fixed time for the identity and log link functions and compared the estimated coefficient for the binary covariate to the true coefficient. All of the available model estimation options were run and compared in the simulation study. We report a subset of the findings that are representative of the main conclusions, using a sample size of 500 observations, with 1000 simulation replicates.

The true values of the coefficients were determined by generated a very large sample of

covariates \mathbf{Q} , then calculating the corresponding true values of the cumulative incidence or restricted mean life time lost, and finally regressing those true values against the covariates using the link function. Samples large enough to achieve a precision of $1e-4$ on the coefficient values were used. Code for reproducing the simulation study is available in the reproducibility materials as an R script.

A.2. Results

Under the null setting, where none of the covariates are associated with the outcome, we find that all of the methods are approximately unbiased and preserve the nominal type I error rate (data not shown). This holds regardless of whether or how strongly associated the covariates are with censoring (similar to what was found in the simulations study of Binder *et al.* (2014)). The more interesting results are where we find when and how the standard pseudo-observation method and the stratified method break down due to dependent censoring. In what follows, we present settings with samples of size 500 and the identity link. The patterns of relative performance of the methods for other sample sizes and link functions are similar.

In Table 1, we show the bias of the coefficient estimates for the cumulative incidence as a proportion of the true coefficient value and empirical standard deviation over the 1000 simulation replicates in a subset of the scenarios and with a subset of the methods. The `beta.cens` column shows the values of the three coefficients (binary, continuous 1, continuous 2) representing the strengths of the associations between the covariates and censoring. When there is a large amount of censoring (80%), the independent approach shows a large amount of bias. When the censoring depends only on the binary covariate (0.1, 0, 0), the stratified approach effectively removes that bias. When the censoring depends on all three covariates, the stratified approach is misspecified and thus biased, but the weighting methods (`ipcw.aalen` and `ipcw.coxph`) effectively decrease that bias. The true censoring model does not follow the proportional hazards model, and thus the `ipcw.coxph` approach is misspecified and is apparently less efficient but more effective at reducing bias as compared to the `ipcw.aalen` approach. Similar trends were observed with the restricted mean models.

Drilling down into the scenario with a large amount of covariate dependent censoring, we compare the different inverse probability of censoring weighting approaches in Table 2. The second column shows whether the censoring model follows proportional hazards or not, and the weighting column shows the weighting method used. All weighting methods exhibit similarly small amounts of bias, with no clear patterns emerging regarding degrees of bias or relative efficiency. It seems that the Binder approach to weighting combined with either Aalen’s additive hazards model or the Cox proportional hazards model would work well in many plausible scenarios.

Turning now to the variance estimation, Table 3 shows the bias of the standard deviation estimation relative to the empirical standard deviation over the replicates, along with the 95% confidence interval coverage using the different variance estimates. The robust variance estimator (the sandwich variance as implemented in the `sandwich` package) is the clear winner here, with minimal bias and approximately correct coverage in all cases. The corrected variance estimator has marginal benefits over the robust variance estimator in some settings, that is, smaller variance but still correct coverage.

Coeff.	Cens. rate	beta.cens	Independent	Stratified	ipcw.aalen	ipcw.coxph
moderate	0.50	(0.1, 0, 0)	0.054 (0.069)	0.019 (0.066)	0.021 (0.067)	0.019 (0.070)
moderate	0.50	(0.1, 0.1, 0.05)	0.057 (0.069)	0.022 (0.066)	0.026 (0.068)	0.009 (0.072)
moderate	0.80	(0.1, 0, 0)	0.099 (0.154)	-0.007 (0.135)	-0.007 (0.150)	-0.057 (0.163)
moderate	0.80	(0.1, 0.1, 0.05)	0.144 (0.164)	0.034 (0.145)	0.032 (0.191)	-0.055 (0.220)
large	0.50	(0.1, 0, 0)	0.034 (0.047)	0.000 (0.042)	0.003 (0.042)	0.001 (0.045)
large	0.50	(0.1, 0.1, 0.05)	0.059 (0.048)	0.022 (0.044)	0.008 (0.045)	0.002 (0.048)
large	0.80	(0.1, 0, 0)	0.079 (0.099)	-0.005 (0.086)	0.017 (0.095)	0.008 (0.116)
large	0.80	(0.1, 0.1, 0.05)	0.122 (0.104)	0.031 (0.090)	0.006 (0.122)	-0.016 (0.151)

Table 1: Bias and empirical standard deviation of the coefficient estimate under different censoring scenarios and using different estimation methods.

Coeff.	Cens. model	Cens. rate	Weighting	ipcw.aalen	ipcw.coxph
moderate	PH	0.50	Binder	0.019 (0.065)	0.012 (0.066)
moderate	PH	0.50	Hajek	0.020 (0.065)	0.003 (0.066)
moderate	PH	0.80	Binder	-0.005 (0.119)	-0.004 (0.123)
moderate	PH	0.80	Hajek	0.016 (0.124)	0.027 (0.131)
moderate	nonPH	0.50	Binder	0.009 (0.066)	0.005 (0.067)
moderate	nonPH	0.50	Hajek	0.026 (0.068)	0.009 (0.072)
moderate	nonPH	0.80	Binder	0.020 (0.156)	0.042 (0.164)
moderate	nonPH	0.80	Hajek	0.032 (0.191)	-0.055 (0.220)
large	PH	0.50	Binder	0.001 (0.043)	0.001 (0.044)
large	PH	0.50	Hajek	0.004 (0.044)	0.001 (0.045)
large	PH	0.80	Binder	0.010 (0.076)	0.009 (0.077)
large	PH	0.80	Hajek	0.014 (0.079)	0.009 (0.089)
large	nonPH	0.50	Binder	0.004 (0.044)	0.002 (0.045)
large	nonPH	0.50	Hajek	0.008 (0.045)	0.002 (0.048)
large	nonPH	0.80	Binder	-0.025 (0.095)	-0.030 (0.105)
large	nonPH	0.80	Hajek	0.006 (0.122)	-0.016 (0.151)

Table 2: Bias and empirical standard deviation of the coefficient estimate under different censoring scenarios and using different estimation methods.

Scenario	Cens. model	Cens. rate	Corrected	Naive	Robust
Cumulative incidence					
null	PH	0.50	-0.04 (0.94)	-0.04 (0.94)	-0.03 (0.94)
null	PH	0.80	-0.04 (0.93)	-0.04 (0.94)	-0.03 (0.93)
null	nonPH	0.50	-0.00 (0.94)	0.00 (0.95)	0.01 (0.94)
null	nonPH	0.80	0.02 (0.95)	0.02 (0.96)	0.03 (0.96)
moderate	PH	0.50	0.02 (0.95)	0.01 (0.95)	0.05 (0.96)
moderate	PH	0.80	-0.04 (0.94)	-0.03 (0.94)	0.03 (0.95)
moderate	nonPH	0.50	-0.02 (0.94)	-0.04 (0.94)	0.02 (0.95)
moderate	nonPH	0.80	-0.08 (0.93)	-0.04 (0.95)	0.08 (0.96)
large	PH	0.50	-0.01 (0.93)	0.25 (0.98)	0.12 (0.96)
large	PH	0.80	-0.06 (0.91)	0.21 (0.98)	0.14 (0.97)
large	nonPH	0.50	0.02 (0.94)	0.28 (0.98)	0.12 (0.97)
large	nonPH	0.80	-0.07 (0.90)	0.19 (0.97)	0.10 (0.97)
Restricted mean					
null	PH	0.50	—	-0.04 (0.94)	-0.05 (0.93)
null	PH	0.80	—	-0.04 (0.94)	-0.04 (0.94)
null	nonPH	0.50	—	-0.01 (0.96)	-0.01 (0.94)
null	nonPH	0.80	—	0.04 (0.97)	0.03 (0.93)
moderate	PH	0.50	—	0.06 (0.97)	0.02 (0.96)
moderate	PH	0.80	—	0.03 (0.95)	0.02 (0.95)
moderate	nonPH	0.50	—	0.04 (0.95)	0.00 (0.94)
moderate	nonPH	0.80	—	0.03 (0.95)	0.04 (0.95)
large	PH	0.50	—	0.32 (0.99)	0.05 (0.96)
large	PH	0.80	—	0.29 (0.99)	0.09 (0.96)
large	nonPH	0.50	—	0.32 (0.99)	0.03 (0.95)
large	nonPH	0.80	—	0.25 (0.98)	0.05 (0.94)

Table 3: Proportional bias of the estimated standard error relative to the empirical standard deviation and 95% confidence interval coverage.

Affiliation:

Michael C. Sachs
 Department of Medical Epidemiology and Biostatistics
 Karolinska Institutet
 11334 Stockholm, Sweden
 E-mail: michael.sachs@ki.se
 URL: <http://sachsmc.github.io/>

Journal of Statistical Software

published by the Foundation for Open Access Statistics

April 2022, Volume 102, Issue 9

doi:10.18637/jss.v102.i09

<https://www.jstatsoft.org/>

<https://www.foastat.org/>

Submitted: 2020-12-01

Accepted: 2021-07-14