penPHcure: Variable Selection in Proportional Hazards Cure Model with Time-Varying Covariates

by Alessandro Beretta and Cédric Heuchenne

Abstract We describe the **penPHcure** R package, which implements the semiparametric proportional-hazards (PH) cure model of Sy and Taylor (2000) extended to time-varying covariates and the variable selection technique based on its SCAD-penalized likelihood proposed by Beretta and Heuchenne (2019a). In survival analysis, cure models are a useful tool when a fraction of the population is likely to be immune from the event of interest. They can separate the effects of certain factors on the probability of being susceptible and on the time until the occurrence of the event. Moreover, the **penPHcure** package allows the user to simulate data from a PH cure model, where the event-times are generated on a continuous scale from a piecewise exponential distribution conditional on time-varying covariates, with a method similar to Hendry (2014). We present the results of a simulation study to assess the finite sample performance of the methodology and illustrate the functionalities of the **penPHcure** package using criminal recidivism data.

1 Introduction

In contrast to other statistical methods, survival analysis models are designed to model the time to an event of interest (e.g., death or occurrence of a disease in medical studies). A typical feature of time-to-event data is the presence of right censoring, an incomplete information problem that arises when a subject is lost to follow-up or does not experience the event before the end of the study. In these cases, it is unknown whether the subject will eventually experience the event and when it will occur, given that it can occur. The most common assumption of standard survival analysis models is that the whole population will sooner or later experience the event of interest. However, in practice, this may not be the case because a fraction of the population may be immune (i.e., not susceptible) to this event. *Cure models*, also known as *split population duration models* or *limited-failure population models*, were developed to handle this kind of situation. They allow us to investigate the effects of some covariates (e.g., type of treatment, stage of the tumor, sex, or age) on the probability to be susceptible to the event of interest (i.e., incidence), and on the survival time conditional on being susceptible (i.e., latency).

Originally, cure models were introduced in the medical literature by Boag (1949) and Berkson and Gage (1952), but they have been used in several other disciplines during the years. In reliability engineering, Meeker (1987) investigates the failure of solid-state electronic components (e.g., integrated circuits). In social science, Schmidt and Witte (1989) investigate the timing of return to prison for a sample of prison releases, and they use it to make predictions of whether or not individuals return to prison. In finance, Cole and Gunther (1995) analyze the determinants of commercial bank failures in the United States; in credit scoring, Tong et al. (2012) predict defaults on a portfolio of UK personal loans. In political science, Svolik (2008) studies the likelihood that a democracy consolidates and the timing of authoritarian reversals in democracies that are not consolidated. In marketing, Polo et al. (2011) investigate the drivers of customer retention in a liberalizing market, using data for a sample of 650 consumers in the Spanish mobile phone industry. In the literature, several variants of cure models have been proposed (see Amico and Van Keilegom (2018) for a comprehensive survey), which belong to two main families: mixture cure models and promotion time cure models.

In this article, we present the **penPHcure** package (Beretta and Heuchenne, 2019b), which implements the semiparametric proportional-hazards (PH) mixture cure model of Sy and Taylor (2000) extended to time-varying covariates, where the incidence and latency distributions are modeled by a logistic regression and a Cox's PH model (Cox, 1972), respectively. The **penPHcure** package contains two main functions: `penPHcure`, to estimate the regression coefficients, their confidence intervals using the basic/percentile bootstrap method, and to perform variable selection using the SCAD-penalized likelihood technique proposed by Beretta and Heuchenne (2019a); and `penPHcure.simulate` to simulate data from a PH cure model, where the event-times are generated on a continuous scale from a piecewise exponential distribution conditional on time-varying covariates, using a method similar to the one described in Hendry (2014).

At the time of writing this article, we are unaware of other R packages for estimation of semiparametric PH mixture cure models with time-varying covariates and, above all, that enable the user to perform variable selection. In the context of cure models for right-censored data, available

R packages include: the **flexsurvcure** package (Amdahl, 2019) for estimation of parametric mixture and non-mixture cure models with time-invariant covariates using time-to-event distributions from the **flexsurv** package (Jackson, 2016); the **nltm** package (Garibotti et al., 2019) for estimation of the semiparametric PH cure model with time-invariant covariates of Tsodikov et al. (2003), as well as other nonlinear transformation models for analyzing survival data using the method of Tsodikov (2003); and the **smcure** package (Cai et al., 2012) for estimation of the semiparametric PH cure model and the accelerated failure time cure model with time-invariant covariates and the **spduration** package (Beger et al., 2018) that implements a parametric cure model with time-varying covariates using Weibull and Log-Logistic latency distributions. Compared to `spduration`, the **penPHcure** package has some advantages: the latency distribution is modeled by a more flexible semiparametric Cox's PH model; the response variable and the time to the event of interest are continuous; and, above all, it allows the user to simultaneously select variables and estimate their parameters using a variable selection technique based on SCAD penalties.

The remainder of this article is structured as follows:

- *Methodology*. We present the PH cure model with time-varying covariates implemented in the `penPHcure` function when the argument `pen.type` is set equal to `"none"` (default);
  - *Variable selection*. We present the variable selection technique based on SCAD penalties implemented in the `penPHcure` function when `pen.type=="SCAD"`;
  - *Data generation*. We describe the algorithm implemented in the `penPHcure.simulate` function, which generates data from a PH cure model with time-varying covariates.
- *Simulation study*. We analyze the finite sample performance of the PH cure model estimates and its variable selection technique implemented in the `penPHcure` function.
- *An application to Criminal Recidivism data*. We provide an example of practical use of the `penPHcure` function to analyze a real data set.

2 Methodology

Let Y be a Bernoulli random variable indicating whether an individual is susceptible ($Y = 1$) or immune ($Y = 0$) to the event of interest with probability $p = P(Y = 1)$. Let T be the time to event, defined only when $Y = 1$. Assuming that a fraction of the population is immune to the event of interest, the marginal survival function of T is defined as

$$S(t) = (1 - p) + pS(t|Y = 1),$$

where p is the incidence (i.e., probability of being susceptible) and $S(t|Y = 1)$ is the latency (i.e., survival function conditional on being susceptible).

The incidence is modeled by a logistic regression model:

$$p(\mathbf{x}) = P(Y = 1|\mathbf{x}) = \frac{\exp(\mathbf{x}'\mathbf{b})}{1 + \exp(\mathbf{x}'\mathbf{b})},$$

where $\mathbf{x}$ is a vector of time-fixed covariates (including the intercept) and $\mathbf{b}$ a vector of unknown coefficients. Whereas the latency is modeled by a Cox's PH model:

$$S(t|Y = 1, \bar{\mathbf{z}}(t)) = \exp\left(-\int_0^t h_0(u)e^{\mathbf{z}'(u)\boldsymbol{\beta}}du\right),$$

where $\mathbf{z}(t)$ is a vector of time-varying covariates (we denote by $\bar{\mathbf{z}}(t)$ the full history of the covariates up to time t), $\boldsymbol{\beta}$ is a vector of unknown coefficients, and $h_0(t)$ is an arbitrary baseline conditional hazard function.

Let $\mathbf{O} = \{(t_i, \delta_i, \bar{\mathbf{z}}_i(t_i), \mathbf{x}_i); i = 1, ..., n\}$ denote the observed data, where t_i is the event/censoring time and δ_i is the censoring indicator, which takes value 1 if t_i is uncensored and 0 otherwise. Since we know that $y_i = 1$ when $\delta_i = 1$, but y_i is unobserved when $\delta_i = 0$, we can estimate the unknown parameters $\boldsymbol{\theta} = (\mathbf{b}, \boldsymbol{\beta}, h_0)$ using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). The complete-data likelihood can be written as

$$L_C(\mathbf{b}, \boldsymbol{\beta}, h_0) = \underbrace{\prod_{i=1}^{n} p(\mathbf{x}_i)^{y_i} [1 - p(\mathbf{x}_i)]^{(1-y_i)}}_{L_1(\mathbf{b})} \times \underbrace{\prod_{i=1}^{n} \left[h_0(t_i)e^{\mathbf{z}_i'(t_i)\boldsymbol{\beta}}\right]^{\delta_i y_i} \left[e^{-\int_0^{t_i} h_0(u)e^{\mathbf{z}_i'(u)\boldsymbol{\beta}}\, du}\right]^{y_i}}_{L_2(\boldsymbol{\beta}, h_0)}, \quad (1)$$

i.e., the product between the incidence component L_1 depending on a set of time-fixed covariates $\mathbf{x}_i$, and the latency component L_2 depending on a set of time-varying covariates $\mathbf{z}_i'(t)$.

Given some starting values $\boldsymbol{\theta}^{(0)}$, the m-th iteration of the EM algorithm consists of two steps:

E step. Compute the expectation of the complete-data likelihood with respect to the conditional distribution of the y_i's given the current parameter estimates $\hat{\boldsymbol{\theta}}^{(m-1)}$ and the observed data $\mathbf{O}$. This expectation is obtained by replacing the y_i's in (1) by their expectation

$$\pi_i^{(m)} = E\left[Y_i \left| \hat{\boldsymbol{\theta}}^{(m-1)}, \mathbf{O}\right.\right] = \delta_i + (1-\delta_i)\frac{p(\mathbf{x}_i)S(t|Y=1,\bar{\mathbf{z}}_i(t))}{1-p(\mathbf{x}_i)+p(\mathbf{x}_i)S(t|Y=1,\bar{\mathbf{z}}_i(t))}.$$

Note that we removed the dependence of the theoretical functions on the estimated parameters to simplify the notation.

M step. Maximize the expected complete-data likelihood with respect to $\mathbf{b}$, $\boldsymbol{\beta}$, and the function h_0. Given $\boldsymbol{\pi}^{(m)} = \{\pi_1^{(m)}, ..., \pi_n^{(m)}\}$, the incidence component L_1 in (1) is maximized using the Newton-Raphson method as in the classical logistic regression model. Whereas the latency component L_2 in (1) is maximized using a profile likelihood approach. The latter involves two steps: (i) the baseline conditional hazard function is estimated nonparametrically by

$$\hat{h}_0(t) = \frac{1}{\left(t_{(j)} - t_{(j-1)}\right)\sum_{i\in R_j}\pi_i^{(m)}e^{\mathbf{z}_i'(t_{(j)})\boldsymbol{\beta}}}, \quad \text{for } t \in (t_{(j-1)}, t_{(j)}], \tag{2}$$

where $t_{(1)} \leq ... \leq t_{(k)}$ are the k ordered event times and R_j is the risk set at $t_{(j)}^-$ (i.e., the set of all individuals who did not experience the event of interest and have not been censored just prior to time $t_{(j)}$); and then (ii) the function h_0 in L_2 is replaced by its estimator given in (2) to obtain the following partial likelihood, which does not depend on the function h_0 anymore,

$$\tilde{L}_2\left(\boldsymbol{\beta}\left|\boldsymbol{\pi}_i^{(m)}\right.\right) = \prod_{j=1}^{k}\frac{e^{\mathbf{z}_i(t_{(j)})\boldsymbol{\beta}}}{\sum_{i\in R_j}\pi_i^{(m)}e^{\mathbf{z}_i'(t_{(j)})\boldsymbol{\beta}}}. \tag{3}$$

Finally, the latency component is estimated by maximizing (3) with respect to $\boldsymbol{\beta}$. In case of tied event-times, (2) and (3) can be rewritten using the Breslow (1974) or Efron (1977) approximation as in the standard Cox's PH model.

The EM algorithm terminates whenever $\left\|\hat{\mathbf{b}}^{(m)} - \hat{\mathbf{b}}^{(m-1)}\right\|_2 < \epsilon$ and $\left\|\hat{\boldsymbol{\beta}}^{(m)} - \hat{\boldsymbol{\beta}}^{(m-1)}\right\|_2 < \epsilon$, where ϵ is a tolerance threshold (by default 10^{-6}).

Variable selection

When the number of available covariates is large, fitting all possible subsets to find the most relevant covariates would be too time consuming. Beretta and Heuchenne (2019a) proposed a regularization method based on the maximization of a penalized version of the complete-data log-likelihood

$$\ell_C^p(\boldsymbol{\theta};\lambda_1,\lambda_2) = \underbrace{\ell_1(\mathbf{b}) - n\sum_{j=2}^{q_1+1}p_{\lambda_1}\left(|b_j|\right)}_{\ell_1^p(\mathbf{b};\lambda_1)} + \underbrace{\ell_2(\boldsymbol{\beta},\mathbf{h}_0) - n\sum_{l=1}^{q_2}p_{\lambda_2}(|\beta_l|)}_{\ell_2^p(\boldsymbol{\beta},\mathbf{h}_0;\lambda_2)},$$

where ℓ denotes a log-likelihood and $p_\lambda(.)$ a SCAD penalty function, which role is to shrink the small coefficients toward zero. We assume that the q_1 and q_2 covariates in the incidence and latency component, respectively, have been standardized, such that the coefficients in $\mathbf{b}$ and $\boldsymbol{\beta}$ are on the same scale. The Smoothly Clipped Absolute Deviation (SCAD) penalty function (Fan and Li, 2001) is given by

$$p_\lambda(|\beta_j|) = \begin{cases} \lambda|\beta_j|, & \text{if } |\beta_j| \leq \lambda \\ \frac{(a^2-1)\lambda^2-(|\beta_j|-a\lambda)^2}{2(a-1)}, & \text{if } \lambda < |\beta_j| \leq a\lambda \\ \frac{(a+1)\lambda^2}{2}, & \text{if } |\beta_j| > a\lambda \end{cases},$$

for some $a > 2$ and $\lambda > 0$. As explained by Fan and Li (2001, 2002) in the context of linear regression, generalized linear models, Cox's PH model, and frailty models, the SCAD estimator has three desirable properties: unbiasedness (do not penalize to much large coefficients), sparsity, and continuity. Moreover, with a proper choice of the tuning parameters (λ, a), it also possesses what is known as the "oracle property", meaning that the SCAD estimator is asymptotically equivalent to the oracle

estimator (i.e., an estimator with only the relevant variables with nonzero coefficients).

For fixed values of the tuning parameters $(\lambda_1, \lambda_2, a_1, a_2)$, estimates of $\boldsymbol{\theta}$ can be obtained using an EM algorithm close to the one described in the previous section. The only difference lies in the M-step. Given that the penalized log-likelihoods $\ell_1^P(\mathbf{b}; \lambda_1)$ and $\tilde{\ell}_2^P(\boldsymbol{\beta}; \lambda_2)$, where $\tilde{\ell}_2^P(\boldsymbol{\beta}; \lambda_2)$ is the logarithm of (3), are non-concave and non-differentiable at the origin, $\mathbf{b}$ and $\boldsymbol{\beta}$ are now estimated using the MM algorithm of Hunter and Li (2005) based on a perturbed Local Quadratic Approximation (LQA) of the penalty function. By default, when the argument `pen.type` of the function `penPHcure` is set equal to `"SCAD"`, the initial values for $\mathbf{b}$ and $\boldsymbol{\beta}$ are vectors with all elements equal to zero. Otherwise, the user can specify other values (e.g., the estimated coefficients of the model with all covariates) using the argument `SV`.

Regarding the choice of the tuning parameters, following the suggestion of Fan and Li (2001), we keep $a_1 = a_2 = 3.7$, and given a set of possible values for (λ_1, λ_2), we select the ones that minimize the following Akaike (AIC) or Bayesian (BIC) Information Criteria:

$$\text{AIC}(\lambda_1, \lambda_2) = -2\ell\left(\hat{\boldsymbol{\theta}}_{\lambda_1,\lambda_2}\right) + 2\nu; \text{ and}$$

$$\text{BIC}(\lambda_1, \lambda_2) = -2\ell\left(\hat{\boldsymbol{\theta}}_{\lambda_1,\lambda_2}\right) + \ln(n)\nu,$$

where $\ell\left(\hat{\boldsymbol{\theta}}_{\lambda_1,\lambda_2}\right)$ is the observed data log-likelihood evaluated at the penalized MLE $\hat{\boldsymbol{\theta}}_{\lambda_1,\lambda_2}$ and ν is the number of nonzero coefficients, identified as the number of coefficients with an absolute value greater than a given threshold (by default 10^{-6}).

Data generation

Let $\mathbf{S} = \{s_1, s_2, ..., s_J\}$ be a partition of the time scale forming $J+1$ intervals $(0, s_1]$, $(s_1, s_2]$, ..., $(s_{J-1}, s_J]$, $(s_J, \infty]$. Define a vector of time-varying covariates piecewise constant in each interval: $\mathbf{z}(t) = \mathbf{z}_j$, for $t \in (s_{j-1}, s_j]$. Consider a transformation g, such that $g(0) = 0$, $g(t)$ is strictly increasing for $t > 0$, and $g^{-1}(t)$ is differentiable. In the implementation of the `penPHcure.simulate` function, we use $g(t) = t^{1/\gamma}$, where the parameter γ can be specified by the user via the argument `gamma`, which by default is equal to 1. According to Hendry (2014), if we generate a random variable V as a piecewise exponential distribution with density function given by

$$f_V(t) = \prod_{l=1}^{j-1} \exp\{-\lambda_l[g^{-1}(s_l) - g^{-1}(s_{l-1})]\}\lambda_j \exp\{-\lambda_j[t - g^{-1}(s_{j-1})]\}, \text{ for } t \in (g^{-1}(s_{j-1}), g^{-1}(s_j)],$$

where $\lambda_j = \exp(\mathbf{z}_j'\boldsymbol{\beta})$ is the constant hazard in the interval $(g^{-1}(s_{j-1}), g^{-1}(s_j)]$, then $g(V)$ follows a Cox's PH model with time-varying covariates with a baseline hazard function given by $h_0(t) = \frac{d}{dt}[g^{-1}(t)]$. This method is part of the algorithm implemented in the `penPHcure.simulate` function to simulate data from a PH cure model with time-varying covariates (see Table 1 for a detailed description).

Require: N, sample size; $\mathbf{S}$, partition of the time scale; $g(t)$, variable transformation; $\mathbf{b}$, incidence coefficients; $\boldsymbol{\beta}$, latency coefficients.

for $i = 1, ..., N$ **do**

1. Generate a vector $\mathbf{x}_i$ from an arbitrary distribution;
2. Generate y_i from a Bernoulli distribution with probability $p(\mathbf{x}_i)$;
3. Generate $\mathbf{z}_i = \{\mathbf{z}_{i,1}, \mathbf{z}_{i,2}, ...\mathbf{z}_{i,J}\}$ from an arbitrary distribution;
4. Generate v_i from a piecewise exponential distribution with density $f_V(t)$;
5. Compute $w_i = g(v_i)$;
6. Generate c_i from an arbitrary distribution;

if $y_i = 0$ **or** $w_i > c_i$ **then**

$t_i = c_i$;
$\delta_i = 0$;

else

$t_i = w_i$;
$\delta_i = 1$;

end if

end for

return $\{(t_i, \delta_i, \mathbf{z}_i, \mathbf{x}_i); i = 1, ..., N\}$

Table 1: Data generation algorithm: PH cure model with time-varying covariates

3 Simulation study

In this section, we present the results of a simulation study conducted to assess the finite sample performance of the PH cure model estimation and its variable selection technique implemented in the `penPHcure` function. The event-times follow a Cox's PH model with baseline hazard function $h_0(t) = 3t^2$ and 8 time-varying covariates. These covariates are constant within $J = 30$ equally-spaced intervals $(0, s_1]$, $(s_1, s_2]$, ..., $(s_{J-1}, s_J]$, where $s_1 = 0.2$ and $s_J = 6$. They follow a multivariate normal distribution $\mathbf{z}_{i,j} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, where $\Sigma_{p,q} = 0.5^{|p-q|}$, for $p, q = 1, ..., 8$. The censoring times follow an exponential distribution truncated above 6 and with parameter λ_C. The cure indicators are generated from a logistic regression model with 8 time-fixed covariates that follow a multivariate normal distribution $\mathbf{x}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, where $\Sigma_{p,q} = 0.5^{|p-q|}$, for $p, q = 1, ..., 8$. Finally, the regression coefficients vectors are set equal to $\boldsymbol{\beta}_0 = (-0.7, 0, 1, 0, -0.5, 0.75, 0, 0)'$ and $\mathbf{b}_0 = (b_0, 1.5, 0, -0.75, 0, -1.5, 0, 0.75, 0)'$.

We consider 6 simulation settings, with different levels of censoring and proportions of non-susceptible individuals (expressed as a fraction of the sample size), depending on different values of b_0 and λ_C (see Table 2). For each of these settings, we generated 500 replications using the `penPHcure.simulate` function for different sample sizes $N = \{250, 500, 1000\}$. Then, for all simulated datasets, we use the `penPHcure` function to (i) fit a standard PH cure model with all covariates (FULL), (ii) fit a standard PH cure model with the covariates associated to the non-zero coefficients only (ORACLE), and (iii) to perform variable selection using the regularization method with SCAD penalties and tuning parameters chosen according to the BIC criterion. The possible values of the tuning parameters (λ_1, λ_2) are obtained with the function `exp(seq(-6,-1,length.out = 10))`, whereas (a_1, a_2) are kept equal to 3.7. Furthermore, we use the `coxph` function in the **survival** package (Therneau, 2015) to fit the classical Cox's PH model with the covariates associated to the non-zero coefficients only (COX).

Censoring	Cure	λ_C	b_0
Low (40%)	High (30%)	0.02	1.45
Low (40%)	Medium (20%)	0.3	2.35
Medium (60%)	High (45%)	0.35	0.35
Medium (60%)	Medium (30%)	0.75	1.45
High (80%)	High (60%)	0.95	-0.7
High (80%)	Medium (40%)	1.55	0.7

Table 2: Simulation settings (censoring and cure are expressed as fractions of all individuals).

The performance is measured in terms of Mean Estimation Error (MEE) and average number of correct and incorrect zeros identified by the variable selection technique (SCAD). In particular, the estimation error for the incidence component is computed as $E\left[(\hat{p}(\mathbf{x}) - p_0(\mathbf{x}))^2\right]$, where $\hat{p}(\mathbf{x})$ and $p_0(\mathbf{x})$ are the estimated and true probabilities of being susceptible. Whereas the estimation error for the latency component is computed as $E\left[(\hat{S}(T|Y=1) - S_0(T|Y=1))^2\right]$, where $\hat{S}(T|Y=1)$ and $S_0(T|Y=1)$ are the estimated and true survival functions conditional on being susceptible.

In Figures 1 and 2, we provide the MEEs for the incidence and latency components, respectively, while in Figure 3, we provide the average number of correct and incorrect zeros. From those figures, we can see that the PH cure model estimation and its variable selection technique implemented in the `penPHcure` function perform reasonably well. For an increase of the sample size or a decrease of the level of censoring, the MEE decreases, and the number of correct (resp. incorrect) zeros converges to 4 (resp. 0). The MEEs of the ORACLE model are always the lowest ones, but we notice that the ones of the SCAD method tend towards them as the sample size increases. It is important to note that, for a fixed level of censoring, we observe higher MEEs in the case of a lower fraction of cured individuals. The worst results are obtained in situations of high censoring and low cure rates, but it is enough to increase the sample size to obtain better results. This is evidence of the fact that a cure model should always be applied to data with a sufficient number of non-susceptible individuals. Last but not the least important, we note that the use of the classical Cox's PH model (COX) leads to very high errors. This was expected since the model is wrongly specified as it ignores the existence of cured subjects.

Finally, in Table 3, we also present the coverage probabilities of the estimated 95% confidence intervals for the ORACLE model using the basic and percentile bootstrap methods with 500 resamples. In most cases, the basic bootstrap method outperforms the percentile bootstrap method, especially for the smallest sample sizes, with coverage probabilities closer to the 95% nominal level.

The R code used to obtain the results in Figures 1 to 3 (resp. Table 3) are provided in Section 2 (resp. Section 3) of the supplementary material ('beretta-heuchenne.R'). Moreover, in the file 'beretta-heuchenne-suppl.pdf', we provide a table with all the results contained in Figures 1 to 3.

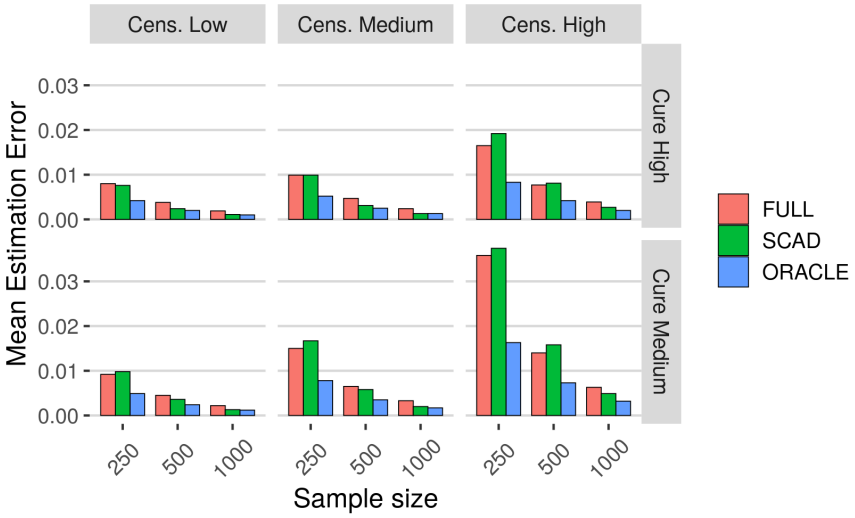

Figure 1: Results of the simulations: mean estimation errors (incidence component).

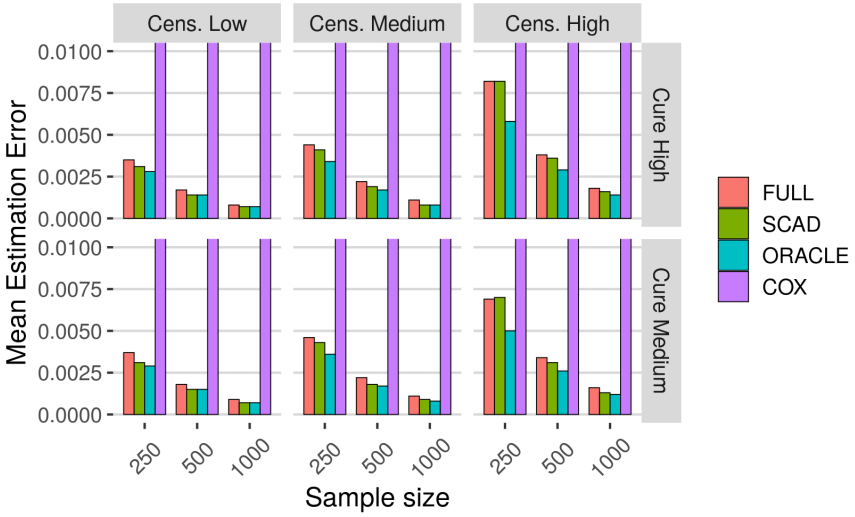

Figure 2: Results of the simulations: mean estimation errors (latency component).

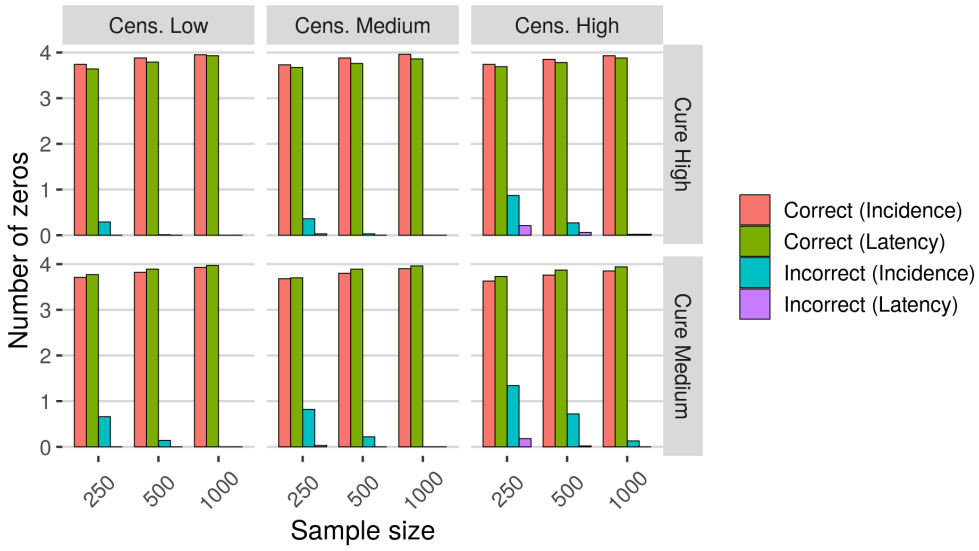

Figure 3: Results of the simulations: average number of correct/incorrect zeros identified by the variable selection technique (SCAD).

cens	cure	N	Method	b_0	b_1	b_2	b_3	b_4	β_1	β_2	β_3	β_4
0.40	0.30	250	basic	0.97	0.97	0.98	0.97	0.97	0.96	0.97	0.96	0.94
			perc	0.9	0.91	0.94	0.91	0.94	0.93	0.94	0.95	0.94
		500	basic	0.97	0.95	0.95	0.94	0.96	0.96	0.96	0.96	0.96
			perc	0.91	0.91	0.94	0.92	0.94	0.94	0.91	0.95	0.95
		1000	basic	0.95	0.96	0.95	0.97	0.97	0.93	0.95	0.95	0.96
			perc	0.94	0.94	0.93	0.95	0.97	0.93	0.95	0.95	0.95
0.40	0.20	250	basic	0.95	0.96	0.99	0.96	0.97	0.96	0.97	0.97	0.96
			perc	0.89	0.9	0.94	0.91	0.93	0.94	0.96	0.96	0.96
		500	basic	0.98	0.97	0.97	0.97	0.98	0.95	0.96	0.96	0.96
			perc	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.93	0.93
		1000	basic	0.97	0.96	0.96	0.95	0.95	0.94	0.93	0.96	0.95
			perc	0.95	0.94	0.94	0.93	0.94	0.94	0.94	0.96	0.95
0.60	0.45	250	basic	0.98	0.95	0.96	0.96	0.97	0.98	0.98	0.97	0.97
			perc	0.95	0.91	0.93	0.93	0.94	0.93	0.94	0.94	0.94
		500	basic	0.98	0.95	0.94	0.97	0.96	0.94	0.94	0.96	0.95
			perc	0.96	0.93	0.94	0.92	0.93	0.94	0.92	0.95	0.95
		1000	basic	0.95	0.96	0.96	0.95	0.95	0.95	0.94	0.93	0.95
			perc	0.95	0.94	0.94	0.93	0.94	0.94	0.92	0.94	0.93
0.60	0.30	250	basic	0.96	0.94	0.98	0.94	0.99	0.97	0.96	0.96	0.96
			perc	0.89	0.9	0.91	0.89	0.93	0.94	0.94	0.95	0.95
		500	basic	0.96	0.97	0.98	0.95	0.97	0.96	0.96	0.95	0.95
			perc	0.94	0.91	0.94	0.93	0.93	0.95	0.96	0.95	0.95
		1000	basic	0.96	0.96	0.96	0.95	0.96	0.96	0.95	0.95	0.96
			perc	0.95	0.94	0.94	0.92	0.94	0.94	0.94	0.94	0.95
0.80	0.60	250	basic	0.99	0.97	0.98	0.97	0.97	0.97	0.96	0.97	0.97
			perc	0.94	0.91	0.92	0.91	0.92	0.94	0.92	0.96	0.94
		500	basic	0.98	0.95	0.97	0.95	0.97	0.97	0.96	0.98	0.98
			perc	0.94	0.93	0.95	0.93	0.94	0.95	0.94	0.95	0.96
		1000	basic	0.96	0.95	0.96	0.94	0.96	0.96	0.95	0.94	0.96
			perc	0.93	0.95	0.94	0.93	0.95	0.95	0.94	0.93	0.93
0.80	0.40	250	basic	0.98	0.95	0.99	0.98	1	0.97	0.96	0.99	0.99
			perc	0.92	0.88	0.91	0.91	0.93	0.95	0.93	0.97	0.95
		500	basic	0.97	0.95	0.97	0.96	0.98	0.96	0.96	0.97	0.96
			perc	0.94	0.9	0.93	0.91	0.93	0.94	0.94	0.96	0.94
		1000	basic	0.96	0.96	0.97	0.95	0.95	0.98	0.95	0.93	0.96
			perc	0.96	0.95	0.95	0.94	0.93	0.94	0.94	0.93	0.93

Table 3: Results of the simulations: coverage probabilities.

4 An application to Criminal Recidivism data

In this section, we illustrate the use of the **penPHcure** R package using a Criminal Recidivism dataset, which contains a sample of 432 inmates released from Maryland state prisons and followed for one year after release (Rossi et al., 1980). The aim of this study was to investigate the relationship between the time to first arrest after release and some covariates observed during the follow-up period. In particular, to study the effect of providing financial aid at the moment of release. The original source of the data is the `Rossi` dataset in the **RcmdrPlugin.survival** package (Fox and Carvalho, 2012), which has been converted into a counting process format and included in the **penPHcure** package.

Let us load and illustrate the dataset:

```
> library(penPHcure)
> data("cpRossi", package = "penPHcure")
> head(cpRossi)
  id tstart tstop arrest fin age  race wexp mar paro prio educ emp
1  1      0    20    yes  no  27 black   no  no  yes    3    3  no
2  2      0     9     no  no  18 black   no  no  yes    8    4  no
3  2      9    14     no  no  18 black   no  no  yes    8    4 yes
4  2     14    17    yes  no  18 black   no  no  yes    8    4  no
5  3      0    16     no  no  19 other  yes  no  yes   13    3  no
6  3     16    17     no  no  19 other  yes  no  yes   13    3 yes
> str(cpRossi)
'data.frame': 1405 obs. of  13 variables:
 $ id    : int  1 2 2 2 3 3 3 4 4 4 ...
```

```
 $ tstart: int  0 0 9 14 0 16 17 0 4 21 ...
 $ tstop : int  20 9 14 17 16 17 25 4 21 31 ...
 $ arrest: Factor w/ 2 levels "no","yes": 2 1 1 2 1 1 2 1 1 1 ...
 $ fin   : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 2 2 ...
 $ age   : int  27 18 18 18 19 19 19 23 23 23 ...
 $ race  : Factor w/ 2 levels "black","other": 1 1 1 1 2 2 2 1 1 1 ...
 $ wexp  : Factor w/ 2 levels "no","yes": 1 1 1 1 2 2 2 2 2 2 ...
 $ mar   : Factor w/ 2 levels "yes","no": 2 2 2 2 2 2 2 1 1 1 ...
 $ paro  : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ prio  : int  3 8 8 8 13 13 13 1 1 1 ...
 $ educ  : Factor w/ 3 levels "3","4","5": 1 2 2 2 1 1 1 3 3 3 ...
 $ emp   : Factor w/ 2 levels "no","yes": 1 1 2 1 1 2 1 1 2 1 ...
```

The object cpRossi is a data.frame in counting process format with 1405 observations for 432 individuals on 13 variables. The id variable provides the unique identification number for every individual in the study. The variables tstart and tstop denote the time interval of the observation (measured in weeks). The variable arrest denotes whether the individual has been arrested during the 1-year follow-up period. The remaining explanatory variables are described hereafter.

- fin. Financial aid received after release: yes or no;
- age. Age in years at the time of release;
- race. Race of the individual: black or other;
- wexp. Full-time work experience before incarceration: yes or no;
- mar. Married at the time of release: yes or no;
- paro. Released on parole: yes or no;
- prio. Number of convictions prior to incarceration;
- educ. Level of education: <=9th degree ("3"), 10th or 11th degree ("4"), or >=12th degree ("5");
- emp. Working full time during the observed time interval: yes or no. This is the only variable which is varying over time (e.g., the individual with id = 2 did not work full time during the first 9 weeks after release, then he did for 5 weeks, and, finally, he has been arrested after 3 weeks without working full time).

Using the penPHcure function, by default, we can fit the standard PH cure model. First, we use a formula object with the response on the left of the tilde operator and the explanatory variables to be included in the latency component on the right. The response is a survival object returned by the Surv(tstart,tstop,arrest) function. Then, using the argument cureform, we specify the explanatory variables to be included in the incidence component. By default, these covariates are set equal to the last observation, but in this case, we set the argument which.X = "mean" to compute the time-weighted average over the full history. Finally, setting the argument inference = TRUE, we conduct inference about the parameter estimates via bootstrapping (by default, 100 bootstrap resamples). The user can increase/decrease the number of bootstrap resamples with the argument nboot.

```
> set.seed(123) # for reproducibility
> fit <- penPHcure(Surv(tstart,tstop,arrest)~fin+age+race+wexp+mar+paro+prio+educ+emp,
+                  cureform = ~fin+age+race+wexp+mar+paro+prio+educ+emp,
+                  data = cpRossi,which.X = "mean",inference = TRUE)

Initializing PH cure model with time-varying covariates...

Number of individuals: 432
Censoring proportion: 0.736
Tied failure times: TRUE
Number of unique failure times: 49
Number of covariates in the survival component: 10
Number of covariates in the cure component: 10

Checking starting values...

Fitting standard PH cure model with time-varying covariates... Please wait...

Performing inference via bootstrapping... Please wait ...
  |======================================================================| 100%
DONE!
```

This call to the penPHcure function returned an object of class "PHcure", and we can print a summary of the results using the summary method. By default, confidence intervals are computed using the basic bootstrap method (the alternative is percentile bootstrap) and a confidence level of 95%. In order to control these features, the user can provide the arguments conf.int and conf.int.level, respectively.

```
> summary(fit)

-----------------------------------------------------
+++   PH cure model with time-varying covariates   +++
-----------------------------------------------------
Sample size: 432
Censoring proportion: 0.7361111
Number of unique event times: 49
Tied failure times: TRUE

log-likelihood:  -643.65

-----------------------------------------------------
+++     Cure (incidence) coefficient estimates     +++
+++          and 95% confidence intervals *        +++
-----------------------------------------------------
             Estimate       2.5%     97.5%
(Intercept)  1.136709 -34.041743  9.769052
finyes      -0.455199  -2.299870 12.188817
age         -0.067715  -0.382429  0.413001
raceother   -0.100950  -2.988104 35.336024
wexpyes      0.251663  -3.257339  2.193371
marno        0.261947 -15.041406 35.102574
paroyes     -0.041289  -3.156395  1.659637
prio         0.068443  -0.285553  0.237089
educ4       -0.570782  -2.614311  2.532353
educ5       -1.163257 -39.473732 34.360612
empyes      -0.860659  -3.299354  1.216299

-----------------------------------------------------
+++    Survival (latency) coefficient estimates    +++
+++          and 95% confidence intervals *        +++
-----------------------------------------------------
           Estimate      2.5%     97.5%
finyes     0.062630 -1.427436  1.446067
age        0.046192 -0.067209  0.176043
raceother -0.759985 -2.770654  0.720247
wexpyes   -0.552549 -1.672866  0.576657
marno      0.123655 -2.327914  1.600195
paroyes    0.040388 -0.816177  1.110058
prio       0.048407 -0.107942  0.195763
educ4      0.588156 -0.545885  1.881803
educ5      0.838098 -2.527512  5.118107
empyes    -1.431782 -1.980471 -0.781978

-----------------------------------------------------
* Confidence intervals computed by the basic
  bootstrap method, with 100 replications.
-----------------------------------------------------
```

As you can see, only one covariate (emp) in the latency component is statistically significant (the 95% confidence interval does not include zero). The negative sign of the estimated coefficient implies that the individuals working full time after release have a lower risk of being rearrested (among the individuals susceptible to be rearrested). The lack of significance of the other covariates might be explained by the small sample size, the high level of censoring (only 114 out of 432 individuals have been rearrested), or by potential confounding factors.

We now perform variable selection with the proposed SCAD-penalized likelihood method to check whether other covariates may be relevant to explain incidence and latency. First, we specify the possible values of the tuning parameters (using the argument pen.tuneGrid) and set the starting

values equal to the coefficient estimates from the unpenalized model (using the argument SV). Then, we still use the penPHcure function, but we now include the argument pen.type = "SCAD".

```
> pen.tuneGrid <- list(CURE = list(lambda = seq(0.01,0.12,by=0.01),
+                                  a = 3.7),
+                      SURV = list(lambda = seq(0.01,0.12,by=0.01),
+                                  a = 3.7))
> SV <- list(b=fit$b,beta=fit$beta)
> tuneSCAD <- penPHcure(Surv(tstart,tstop,arrest)~fin+age+race+wexp+mar+paro+prio+educ+emp,
+                       cureform = ~fin+age+race+wexp+mar+paro+prio+educ+emp,
+                       data = cpRossi,which.X = "mean",pen.type = "SCAD",
+                       pen.tuneGrid = pen.tuneGrid,SV = SV)

Initializing PH cure model with time-varying covariates...

Number of individuals: 432
Censoring proportion: 0.736
Tied failure times: TRUE
Number of unique failure times: 49
Number of covariates in the survival component: 10
Number of covariates in the cure component: 10

Checking starting values...

Tuning SCAD-penalized PH cure model with time-varying covariates... Please wait...

  iter     aCURE     aSURV  lambdaCURE  lambdaSURV           AIC           BIC      df
     1      3.70      3.70        0.01        0.01     1319.1625     1384.2573      16
     2      3.70      3.70        0.01        0.02     1319.1625     1384.2573      16
     3      3.70      3.70        0.01        0.03     1316.0665     1360.8192      11
     4      3.70      3.70        0.01        0.04     1318.0458     1358.7300      10
     5      3.70      3.70        0.01        0.05     1318.0457     1358.7300      10

   ...       ...   (omitted rows)  ...         ...     (omitted rows)      ...     ...

   140      3.70      3.70        0.12        0.08     1325.5349     1333.6718       2
   141      3.70      3.70        0.12        0.09     1325.5349     1333.6718       2
   142      3.70      3.70        0.12        0.10     1325.5349     1333.6718       2
   143      3.70      3.70        0.12        0.11     1325.5349     1333.6718       2
   144      3.70      3.70        0.12        0.12     1325.5349     1333.6718       2

DONE!
```

This time, the call to the penPHcure function returned an object of class "penPHcure". We can print a summary of the results using the summary method, and, by default, the fitted model with the lowest BIC criterion is returned.

```
> summary(tuneSCAD)

-----------------------------------------------------
+++   PH cure model with time-varying covariates   +++
+++             [ Variable selection ]             +++
-----------------------------------------------------
Sample size:  432
Censoring proportion:  0.7361111
Number of unique event times: 49
Tied failure times:  TRUE
Penalty type:  SCAD
Selection criterion:  BIC

-----------------------------------------------------
+++               Tuning parameters                +++
-----------------------------------------------------
 Cure (incidence) --- lambda:  0.09
                          a:  3.7
```

```
 Survival (latency) - lambda:  0.05
                           a:  3.7

 BIC =  1329.481

-------------------------------------------------------
+++                Cure (incidence)                +++
+++     [ Coefficients of selected covariates ]    +++
-------------------------------------------------------
             Estimate
(Intercept)  1.776907
age         -0.076498

-------------------------------------------------------
+++              Survival (latency)                +++
+++     [ Coefficients of selected covariates ]    +++
-------------------------------------------------------
        Estimate
prio    0.101202
empyes -1.537286
```

The Bayesian Information Criterion is minimized for $\lambda_1 = 0.09$ and $\lambda_1 = 0.05$. In this case, the covariate age is selected in the incidence component. The negative sign of the estimated coefficient implies that younger individuals are more susceptible to be rearrested. The covariates `prio` and `emp` are selected in the latency component. The positive sign of the estimated coefficient (`prio`) implies that a higher number of convictions prior to incarceration increases the risk of being rearrested (among the individuals susceptible to be rearrested).

Let us now have a look at the fitted model with the lowest AIC criterion:

```
> summary(tuneSCAD,crit.type = "AIC")

-------------------------------------------------------
+++   PH cure model with time-varying covariates   +++
+++             [ Variable selection ]             +++
-------------------------------------------------------
Sample size:  432
Censoring proportion:  0.7361111
Number of unique event times: 49
Tied failure times:  TRUE
Penalty type:  SCAD
Selection criterion:  AIC

-------------------------------------------------------
+++               Tuning parameters                +++
-------------------------------------------------------
 Cure (incidence) --- lambda:  0.06
                           a:  3.7

 Survival (latency) - lambda:  0.03
                           a:  3.7

 AIC =  1310.79

-------------------------------------------------------
+++                Cure (incidence)                +++
+++     [ Coefficients of selected covariates ]    +++
-------------------------------------------------------
             Estimate
(Intercept)  1.829260
finyes      -0.585638
age         -0.067130
educ5       -0.887636

-------------------------------------------------------
```

```
+++              Survival (latency)                +++
+++     [ Coefficients of selected covariates ]    +++
-------------------------------------------------------
           Estimate
raceother -0.586626
prio       0.103746
empyes    -1.552737
```

The Akaike Information Criterion is minimized for $\lambda_1 = 0.06$ and $\lambda_1 = 0.03$. As expected the AIC criterion selected a less penalized and more complex model. In the incidence component, also the covariates `fin` and `educ` have been selected. The negative signs imply that individuals who received financial aid or with a high level of education (>=12th degree) are less susceptible to be rearrested. In the latency component, also the covariate `race` has been selected. The negative coefficient implies that individuals of a race other than black have a lower risk of being rearrested (among the individuals susceptible to be rearrested).

5 Conclusion

In survival analysis studies, it may be the case that a fraction of the population is likely to be not susceptible to the event of interest. In this article, we presented the **penPHcure** R package, which implements the semiparametric proportional-hazards (PH) cure model of Sy and Taylor (2000) extended to time-varying covariates. This model can measure the effects of some covariates on the probability of being susceptible and on the time until the occurrence of the event. The **penPHcure** package is composed of two main functions: `penPHcure`, to estimate the regression coefficients, their confidence intervals using the basic/percentile bootstrap method and to perform variable selection using the SCAD-penalized likelihood technique proposed by Beretta and Heuchenne (2019a); and `penPHcure.simulate` to simulate data from a PH cure model with time-dependent covariates. We first explained the methodology behind these functions and presented the results of a simulation study to assess its finite-sample performance. Then, we illustrated the use of the `penPHcure` function through an example based on the Criminal Recidivism dataset.

6 Availability

The latest release and a development version of the **penPHcure** package are respectively avilable on CRAN and at https://github.com/a-beretta/penPHcure.

Bibliography

J. Amdahl. *flexsurvcure: Flexible Parametric Cure Models*, 2019. URL https://CRAN.R-project.org/package=flexsurvcure. R package version 1.0.0. [p54]

M. Amico and I. Van Keilegom. Cure models in survival analysis. *Annual Review of Statistics and Its Application*, 5(1):311–342, 2018. URL https://doi.org/10.1146/annurev-statistics-031017-100101. [p53]

A. Beger, D. Chiba, D. W. Hill, Jr., N. W. Metternich, S. Minhas, and M. D. Ward. *spduration: Split-Population Duration (Cure) Regression*, 2018. URL https://CRAN.R-project.org/package=spduration. R package version 0.17.1. [p54]

A. Beretta and C. Heuchenne. Variable selection in proportional hazards cure model with time-varying covariates, application to us bank failures. *Journal of Applied Statistics*, 46(9):1529–1549, 2019a. URL https://doi.org/10.1080/02664763.2018.1554627. [p53, 55, 64]

A. Beretta and C. Heuchenne. *penPHcure: Variable Selection in PH Cure Model with Time-Varying Covariates*, 2019b. URL https://CRAN.R-project.org/package=penPHcure. R package version 1.0.2. [p53]

J. Berkson and R. P. Gage. Survival curve for cancer patients following treatment. *Journal of the American Statistical Association*, 47(259):501–515, 1952. URL https://doi.org/10.2307/2281318. [p53]

J. W. Boag. Maximum likelihood estimates of the proportion of patients cured by cancer therapy. *Journal of the Royal Statistical Society: Series B (Methodological)*, 11(1):15–44, 1949. URL https://doi.org/10.1111/j.2517-6161.1949.tb00020.x. [p53]

N. Breslow. Covariance analysis of censored survival data. *Biometrics*, 30(1):89–99, 1974. URL https://doi.org/10.2307/2529620. [p55]

C. Cai, Y. Zou, Y. Peng, and J. Zhang. *smcure: Fit Semiparametric Mixture Cure Models*, 2012. URL https://CRAN.R-project.org/package=smcure. R package version 2.0. [p54]

R. A. Cole and J. W. Gunther. Separating the likelihood and timing of bank failure. *Journal of Banking & Finance*, 19(6):1073 – 1089, 1995. URL https://doi.org/10.1016/0378-4266(95)98952-M. [p53]

D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972. URL https://doi.org/10.2307/2985181. [p53]

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. URL https://doi.org/10.1111/j.2517-6161.1977.tb01600.x. [p54]

B. Efron. The efficiency of cox's likelihood function for censored data. *Journal of the American Statistical Association*, 72(359):557–565, 1977. URL https://doi.org/10.1080/01621459.1977.10480613. [p55]

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001. URL https://doi.org/10.1198/016214501753382273. [p55, 56]

J. Fan and R. Li. Variable selection for cox's proportional hazards model and frailty model. *Annals of Statistics*, 30(1):74–99, 2002. URL https://doi.org/10.1214/aos/1015362185. [p55]

J. Fox and M. Carvalho. The rcmdrplugin.survival package: Extending the r commander interface to survival analysis. *Journal of Statistical Software, Articles*, 49(7):1–32, 2012. URL https://doi.org/10.18637/jss.v049.i07. [p59]

G. Garibotti, A. Tsodikov, and M. Clements. *nltm: Non-Linear Transformation Models*, 2019. URL https://CRAN.R-project.org/package=nltm. R package version 1.4.2. [p54]

D. Hendry. Data generation for the cox proportional hazards model with time-dependent covariates: a method for medical researchers. *Statistics in medicine*, 33:436–454, 2014. URL https://doi.org/10.1002/sim.5945. [p53, 56]

D. R. Hunter and R. Li. Variable selection using mm algorithms. *The Annals of Statistics*, 33(4):1617–1642, 08 2005. URL https://doi.org/10.1214/009053605000000200. [p56]

C. Jackson. flexsurv: A platform for parametric survival modeling in R. *Journal of Statistical Software*, 70(8):1–33, 2016. URL https://doi.org/10.18637/jss.v070.i08. [p54]

W. Q. Meeker. Limited failure population life tests: Application to integrated circuit reliability. *Technometrics*, 29(1):51–65, 1987. URL https://doi.org/10.1080/00401706.1987.10488183. [p53]

Y. Polo, F. J. Sese, and P. C. Verhoef. The effect of pricing and advertising on customer retention in a liberalizing market. *Journal of Interactive Marketing*, 25(4):201 – 214, 2011. URL https://doi.org/10.1016/j.intmar.2011.02.002. [p53]

P. H. Rossi, R. A. Berk, and K. J. Lenihan. 2 - historical background of the transitional aid research project experiments. In P. H. Rossi, R. A. Berk, and K. J. Lenihan, editors, *Money, Work, and Crime*, pages 21 – 46. Academic Press, 1980. URL https://doi.org/10.1016/B978-0-12-598240-5.50009-0. [p59]

P. Schmidt and A. D. Witte. Predicting criminal recidivism using 'split population' survival time models. *Journal of Econometrics*, 40(1):141 – 159, 1989. URL https://doi.org/10.1016/0304-4076(89)90034-1. [p53]

M. Svolik. Authoritarian reversals and democratic consolidation. *American Political Science Review*, 102 (2):153–168, 2008. URL https://doi.org/10.1017/S0003055408080143. [p53]

J. P. Sy and J. M. G. Taylor. Estimation in a cox proportional hazards cure model. *Biometrics*, 56(1): 227–236, 2000. URL https://doi.org/10.1111/j.0006-341X.2000.00227.x. [p53, 64]

T. M. Therneau. *A Package for Survival Analysis in S*, 2015. URL https://CRAN.R-project.org/package=survival. version 2.38. [p57]

E. N. Tong, C. Mues, and L. C. Thomas. Mixture cure models in credit scoring: If and when borrowers default. *European Journal of Operational Research*, 218(1):132–139, 2012. URL https://doi.org/10.1016/j.ejor.2011.10.007. [p53]

A. Tsodikov. Semiparametric models: a generalized self-consistency approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3):759–774, 2003. URL https://doi.org/10.1111/1467-9868.00414. [p54]

A. D. Tsodikov, J. G. Ibrahim, and A. Y. Yakovlev. Estimating cure rates from survival data: An alternative to two-component mixture models. *Journal of the American Statistical Association*, 98(464): 1063–1078, 2003. URL https://doi.org/10.1198/016214503000000001007. [p54]

Alessandro Beretta
Centre for Quantitative Methods and Operations Management (QuantOM)
HEC Liège
Rue Louvrex, 14 - 4000 Liège
Belgium
a.beretta@uliege.be

Cédric Heuchenne
Centre for Quantitative Methods and Operations Management (QuantOM)
HEC Liège
Rue Louvrex, 14 - 4000 Liège
Belgium
c.heuchenne@uliege.be