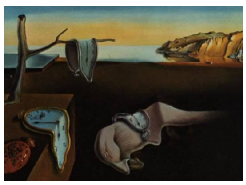
View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Journal of Statistical Software



Journal of Statistical Software
August 2013, Volume 54, Issue 11. http://www.jstatsoft.org/

survPresmooth: An R Package for Presmoothed Estimation in Survival Analysis

Ignacio López-de-Ullibarri
Universidade da Coruña

M. Amalia Jácome
Universidade da Coruña

Abstract

The **survPresmooth** package for R implements nonparametric presmoothed estimators of the main functions studied in survival analysis (survival, density, hazard and cumulative hazard functions). Presmoothed versions of the classical nonparametric estimators have been shown to increase efficiency if the presmoothing bandwidth is suitably chosen. The **survPresmooth** package provides plug-in and bootstrap bandwidth selectors, also allowing the possibility of using fixed bandwidths.

Keywords: nonparametric estimation, presmoothing, R.

1. Introduction

Survival analysis is oriented to the study of the random time (lifetime, failure time) T from an initial point to the occurrence of some event of interest. An important goal is to estimate the functions that characterize the distribution of T (in the following, assumed to be absolutely continuous): (a) the distribution function, $F(t) = P(T \leq t)$ or, equivalently, the survival function, $S(t) = 1 - F(t)$, (b) the density function, $f(t) = F'(t)$, c) the hazard function, $\lambda(t) = \lim_{\Delta t \to 0^+} P(t \leq T < t + \Delta t | T \geq t)/\Delta t = f(t)/S(t)$ and d) the cumulative hazard function, $\Lambda(t) = \int_0^t \lambda(v)dv$, for $t > 0$. The handling of incomplete observations is one of the major problems one has to face in the analysis of lifetimes. Typically, the true lifetimes are incompletely observed due to censoring. In the right censoring (RC) model, the lifetime T can be observed only if its value is smaller than that of an independent censoring variable C. Thus, based on a random sample (T_i, C_i), $i = 1, \ldots, n$, the actual information for the ith observation is conveyed by the pair (Z_i, δ_i), where $Z_i = \min(T_i, C_i)$ is the observed time and $\delta_i = \mathbf{1}_{\{T_i < C_i\}}$ indicates whether the observation is censored ($\delta_i = 0$) or not ($\delta_i = 1$).

Posed in statistical terms, the problem is how to estimate the different functionals of the lifetime T using the observed (Z, δ). Classical nonparametric estimators in the presence of

right censoring are well established in the literature. The Kaplan-Meier (KM) estimator of the survival function (Kaplan and Meier 1958), the kernel estimator of the density with KM weights (Földes, Rejtö, and Winter 1981), the kernel estimator of the hazard function by Tanner and Wong (1983) and the Nelson-Aalen (NA) estimator of the cumulative hazard function (Nelson 1972; Aalen 1978) are a representative selection of this type of estimators. A general account of these estimators can be found in standard texts on survival analysis (see e.g., Klein and Moeschberger 2003).

To motivate the presmoothing procedures, note that the KM and NA estimators are step functions with jumps located only at the uncensored observations. Therefore, when many data are censored, the KM and NA estimators have only a few jumps with increasing sizes and the accuracy of the estimation might not be acceptable. Heavily censored data sets are becoming more frequent, since developments lead to increasing lifetimes, and if the testing time is not enlarged (and it usually can not be enlarged), an increase in lifetimes leads to increasing censoring. In such a situation, more efficient competitors for the classical estimators are essential. The presmoothed estimators are a good alternative, since they are computed by giving mass to all the data, including the censored observations. Central to the idea behind presmoothing is the function $p(t) = P(\delta = 1|Z = t)$, i.e., the conditional probability that the observation at time t is not censored. The function p depends on the observable variables (Z, δ), and for this reason, it can be easily estimated. Another important feature of p is that functionals of the incomplete lifetimes T can be expressed in terms of $p(t)$ and functions of the observed (Z, δ). For example, for the cumulative hazard rate we have

$$\Lambda(t) = \int_0^t \frac{p(u)dH(u)}{1 - H(u^-)},$$

where H denotes the distribution function of Z. The classical NA estimator of Λ is obtained by replacing H with its empirical estimator H_n and the value of $p(Z_i)$ by the corresponding indicator of non-censoring δ_i, giving rise to a step function with jumps only at the uncensored data:

$$\widehat{\Lambda}_n^{NA}(t) = \frac{1}{n} \sum_{i:Z_i \leq t} \frac{\delta_i}{1 - H_n(Z_i) + 1/n}. \tag{1}$$

The straightforward idea on which the presmoothed estimators are based is to consider a smoother estimator of $p(Z_i)$ rather than δ_i. This has important implications:

(a) The new estimators are computed by giving mass to each observation regardless of whether it is censored or not. Thus, more information on the local behavior of the lifetime distribution is provided. The accuracy of the estimation is then increased, above all for heavily censored data.

(b) Using the smooth estimator of p, the available information can be extrapolated to better describe the tail behavior.

Since δ is a dichotomic variable, p can also be written as a regression function $p(t) = E(\delta|Z = t)$. Thus, p can be estimated parametrically (e.g., using a logistic fit) or nonparametrically, for example, using the Nadaraya-Watson (NW) kernel estimator (Nadaraya 1964; Watson 1964)

with bandwidth b_1:

$$\widehat{p}_{b_1}(t) = \frac{\sum_{i=1}^{n} K_{b_1}(t - Z_i)\delta_i}{\sum_{i=1}^{n} K_{b_1}(t - Z_i)}, \tag{2}$$

where K is a kernel function and $K_b(t) = b^{-1}K(t/b)$ denotes the rescaled kernel. Typically K is a symmetric density function compactly supported, without loss of generality, in the interval $[-1, 1]$.

Estimation of S and Λ with a logistic fit of p has been studied by Dikta (1998, 2000, 2001). It is shown in Dikta (1998) that, when the parametric model assumed for p is correct, this semiparametric estimator of S is at least as efficient as the KM estimator in terms of the asymptotic variance. As a drawback, there is a clear risk of a miss-specification of the parametric model for p.

The presmoothed approach is based on the NW estimator of p, and has been extensively studied in the literature in the estimation of S and Λ (Cao, López-de-Ullibarri, Janssen, and Veraverbeke 2005), the density f (Cao and Jácome 2004; Jácome and Cao 2007; Jácome, Gijbels, and Cao 2008), the hazard rate λ (Cao and López-de-Ullibarri 2007), and also the quantile function (Jácome and Cao 2008) (for an illustration of the use of nonparametric regression estimators other than the NW smoother, see Jácome *et al.* 2008). Nonparametric kernel regression, as the NW estimator, does not requires preliminary specification of a parametric family. In contrast, a bandwidth b_1 must be chosen for the computation of $\widehat{p}_{b_1}(t)$. Note that when the bandwidth is very small then $\widehat{p}_{b_1}(Z_i) \simeq \delta_i$, and the presmoothed estimators reduce to the classical ones.

The beneficial effect of presmoothing depends, as expected, on the choice of the presmoothing bandwidth b_1. When the asymptotically optimal bandwidth is used, the presmoothed estimators have smaller asymptotic variance and, therefore, a better performance in terms of mean squared error (MSE). This improvement is of second order in the estimation of S and Λ (Cao *et al.* 2005), but may be of first order for the density function (Cao and Jácome 2004). The simulation studies confirm this gain in efficiency under moderate sample sizes. Moreover, they also show that the presmoothed estimators are better than the classical ones, not only for the optimal value of the bandwidth but for quite wide ranges of values of b_1. A comparison of the semiparametric and presmoothed estimators of S has been carried out under left truncation and right censored (LTRC) data by Jácome and Iglesias-Pérez (2008), where the nice behavior of both estimators, with respect to the classical one, is shown in a simulation study. Specifically, the presmoothed estimator has a better performance than the classical estimator in the complete interval of computation, and than the semiparametric estimator for inner points, while the improvement vanishes in the boundary of the interval. In summary, this good performance suggests that presmoothing is a competitive method that may outperforms the classical estimators.

The **survPresmooth** package (López-de-Ullibarri and Jácome 2013) provides an implementation in R (R Core Team 2013) of the presmoothed estimators of the functions S, f, λ and Λ in the RC model, including methods for bandwidth selection and correction of possible boundary effects.

Our main purpose on writing this paper was twofold: (a) to introduce the **survPresmooth** package to R users, providing at the same time a review of presmoothing techniques; and (b) to

show the performance of presmoothed estimators both in the analysis of a real dataset and in simulated scenarios. The presmoothed estimators implemented in the package are reviewed in Section 2. The two following sections deal with additional technical aspects of presmoothing, like bandwidth-parameter selection (Section 3) or boundary-effect correction (Section 4). In Section 5, after describing the package functions, the implemented presmoothed estimation procedures are applied to a real dataset and their performance is shown by means of a simulation study. Some concluding remarks are given in Section 6.

2. Presmoothed estimators

Survival and distribution functions

The presmoothed estimator of the survival function S (Jácome and Cao 2007) is

$$\widehat{S}^P_{b_1}(t) = \prod_{i:Z_i \leq t} \left(1 - \frac{\widehat{p}_{b_1}(Z_i)}{n(1 - H_n(Z_i) + 1/n)}\right).$$

It can be derived from the KM estimator,

$$\widehat{S}^{KM}_n(t) = \prod_{i:Z_i \leq t} \left(1 - \frac{\delta_i}{n(1 - H_n(Z_i) + 1/n)}\right),$$

just by replacing δ_i with the value at point Z_i of the NW estimate of p in Equation 2. An obvious presmoothed estimator of the distribution function F is $\widehat{F}^P_{b_1} = 1 - \widehat{S}^P_{b_1}$.

The estimator $\widehat{S}^P_{b_1}$ is a decreasing step function, with jumps at the observed (censored or uncensored) times. In this aspect it differs from $\widehat{S}^{KM}_n$, whose jumps are restricted to the uncensored times. Two further properties relating the presmoothed estimator with its classical counterpart should be mentioned. Firstly, when $b_1 \downarrow 0$, then $\widehat{S}^P_{b_1}$ coincides in the limit with $\widehat{S}^{KM}_n$. Secondly, when there is no censoring, $\widehat{S}^P_{b_1}$ reduces to the empirical estimator of S.

Density function

If F is estimated by a step function $\widehat{F}$, the density $f = F'$ can be estimated by smoothing the increments of $\widehat{F}$. This is the idea behind the most popular nonparametric estimator of f, Parzen-Rosenblatt's (PR) kernel density estimator (Parzen 1962; Rosenblatt 1956):

$$\widehat{f}_{b_2}(t) = \int_0^\infty K_{b_2}(t - u) d\widehat{F}(u) \tag{3}$$

where $b_2 \equiv b_{2n} \downarrow 0$ is the smoothing parameter and K a kernel function.

If, for example, $\widehat{F} \equiv \widehat{F}^{KM}_n = 1 - \widehat{S}^{KM}_n$, simple calculations show that the estimator in Equation 3 takes the form

$$\widehat{f}^{KM}_{b_2}(t) = \sum_{i=1}^{n} K_{b_2}(t - Z_{(i)}) W^{KM}_{(i)},$$

where $Z_{(i)}$ denotes the ith ordered observation and the weights $W^{KM}_{(i)}$ are defined as $W^{KM}_{(i)} = \widehat{F}^{KM}_n(Z_{(i)}) - \widehat{F}^{KM}_n(Z_{(i-1)})$. This is the density estimator proposed by Földes *et al.* (1981).

Note that without censoring, $W_{(i)}^{KM} = 1/n$ and $Z_i = T_i$ for $i = 1, \ldots, n$. Then, the well-known kernel estimator for uncensored data, $\widehat{f}_{b_2}(t) = \sum_{i=1}^{n} K_{b_2}(t - T_i)/n$, is recovered.

In a similar way, if $\widehat{F}_{b_1}^P$ is used to estimate F, a presmoothed estimator of the density function is obtained:

$$\widehat{f}_{b_1,b_2}^P(t) = \int_0^\infty K_{b_2}(t-u)d\widehat{F}_{b_1}^P(u) = \sum_{i=1}^{n} K_{b_2}(t - Z_{(i)})W_{(i),b_1}^P, \tag{4}$$

where $W_{(i),b_1}^P = \widehat{F}_{b_1}^P(Z_{(i)}) - \widehat{F}_{b_1}^P(Z_{(i-1)})$. This estimator depends on two parameters: the presmoothing bandwidth b_1, needed to compute $\widehat{p}_{b_1}$, and a smoothing bandwidth b_2. Key properties of $\widehat{f}_{b_1,b_2}^P$, such as its asymptotic normality and an almost sure asymptotic representation, are proved in Cao and Jácome (2004), Jácome and Cao (2007) and Jácome *et al.* (2008).

Hazard function and cumulative hazard function

There is a rich literature on nonparametric hazard function estimation. Here we restrict ourselves to the estimator proposed by Tanner and Wong (1983) for right-censored data. Noting that $\lambda = \Lambda'$ the Tanner-Wong estimator (TW), very similar to the independent proposals by Ramlau-Hansen (1983) and Yandell (1983), is obtained by smoothing the increments of the NA estimator in Equation 1:

$$\widehat{\lambda}_{b_2}(t) = \int_0^\infty K_{b_2}(t-u)d\widehat{\Lambda}_n^{NA}(u) = \frac{1}{n}\sum_{i=1}^{n} \frac{K_{b_2}(t - Z_i)\delta_i}{1 - H_n(Z_i) + 1/n}.$$

As was pointed out in Section 1, the presmoothed NA estimator of the cumulative hazard function results from substituting δ_i with $\widehat{p}_{b_1}(Z_i)$, and is defined by:

$$\widehat{\Lambda}_{b_1}^P(t) = \frac{1}{n}\sum_{i:Z_i \le t} \frac{\widehat{p}_{b_1}(Z_i)}{1 - H_n(Z_i) + 1/n}.$$

An asymptotic representation and asymptotic distributional properties of $\widehat{\Lambda}_{b_1}^P$ can be found in Cao *et al.* (2005). Some evidence of the beneficial effect of presmoothing is also provided in that reference.

Following the same ideas leading to Equation 4 in the density case, a presmoothed version of the Tanner-Wong estimator of λ (Cao and López-de-Ullibarri 2007) can be obtained:

$$\widehat{\lambda}_{b_1,b_2}^P(t) = \int_0^\infty K_{b_2}(t-u)d\widehat{\Lambda}_{b_1}^P(u) = \frac{1}{n}\sum_{i=1}^{n} \frac{K_{b_2}(t - Z_i)\widehat{p}_{b_1}(Z_i)}{1 - H_n(Z_i) + 1/n}. \tag{5}$$

Like the presmoothed density estimator, $\widehat{\lambda}_{b_1,b_2}^P$ also depends on two parameters, b_1 and the smoothing bandwidth b_2.

3. Bandwidth selection

The new estimators depend on the presmoothing bandwidth b_1, needed to compute the NW estimator $\widehat{p}_{b_1}$. In the case of f and λ, their presmoothed estimators, as the classical counterparts, also depend on a second smoothing bandwidth b_2, which controls the degree of kernel

smoothing. If b_2 is very small, the resulting estimator is too rough and contains spurious features. On the contrary, if b_2 is too large, oversmoothed estimates are obtained, where important features of the underlying structure of f and λ may have been smoothed away.

In general terms, let us denote by φ the target function (i.e., S, Λ, f or λ) and by $\mathbf{b}$ the (scalar or vectorial) bandwidth ($\mathbf{b} = b_1$ for S or Λ and $\mathbf{b} = (b_1, b_2)$ for f or λ). A way of choosing $\mathbf{b}$ is as the minimizer of some error measure, usually the mean integrated squared error (MISE):

$$MISE_{\varphi}(\mathbf{b}) = E\left[ISE_{\varphi}(\mathbf{b})\right] = E\left[\int_0^{\infty} \left(\widehat{\varphi}_{\mathbf{b}}^{P}(t) - \varphi(t)\right)^2 \omega(t)dt\right], \tag{6}$$

where ω is a nonnegative weight function, introduced to allow elimination of boundary effects (Gasser and Müller 1979). In our implementation ω is an indicator function with user-defined support.

Since the MISE depends on the unknown function φ, the optimal bandwidth $\mathbf{b}$ is in practice obtained by minimizing an approximation of the MISE. Different bandwidth selectors are obtained depending on the way the MISE is approximated. The **survPresmooth** package provides plug-in and bootstrap bandwidth selectors (allowing also the possibility of using fixed bandwidths). Both methodologies are competitive in the sense that neither of them can be claimed to be the best procedure in all cases.

When b_1 is close to zero no significant presmoothing is carried out. The **survPresmooth** package makes possible, by fixing the bandwidth $b_1 = 0$, to compute all the classical estimators, and for f and λ also select automatically the smoothing bandwidth for the kernel estimation. In this sense, the usefulness of the package is clear.

3.1. Plug-in bandwidth selector

The complicated structure of the presmoothed estimators makes the MISE in Equation 6 difficult to handle. However, $\widehat{\varphi}_{\mathbf{b}}^{P}$ can be decomposed as a sum of independent and identically distributed (i.i.d.) variables plus a negligible term of lower order (see Cao *et al.* 2005; Cao and López-de-Ullibarri 2007; Jácome and Cao 2007). Replacing $\widehat{\varphi}_{\mathbf{b}}^{P}$ in Equation 6 with this i.i.d. representation yields a more tractable approximation of the MISE, which will be called AMISE. The plug-in methodology consists in replacing the unknown quantities in that AMISE with estimates of them and finding the bandwidth $\mathbf{b}$ minimizing that approximation.

Both for $\varphi = S$ and Λ, the AMISE bandwidth is:

$$b_{1,\varphi}^{AMISE} = \left(\frac{e_K Q}{2n d_K^2 A}\right)^{1/3}, \tag{7}$$

where $e_K = \int_{-1}^{1} uK(u) \int_{-1}^{u} K(t)dtdu$, $d_K = \int_{-1}^{1} t^2 K(t)dt$ and A and Q are defined by:

$$\begin{aligned} Q &= \int_0^{\infty} q(t)\omega(t)dt \quad \text{with} \quad q(t) = \frac{p(t)(1-p(t))h(t)}{(1-H(t))^2}, \\ A &= \int_0^{\infty} \alpha^2(t)\omega(t)dt \quad \text{with} \quad \alpha(t) = \int_0^{t} \frac{p''(u)h(u)/2 + p'(u)h'(u)}{1-H(u)}du, \end{aligned}$$

and $h = H'$ is the density of Z. The plug-in bandwidth selector of b_1 results from replacing in Equation 7 the constants Q and A with estimates of them (obtained by replacing H, h, h',

p, p', and p'' with their corresponding estimators). In our implementation, we use for H the empirical estimator, while kernel-type estimators are used for p (NW estimator) and h (PR estimator) with pilot bandwidths g_1 and g_2 respectively:

$$\widehat{p}_{g_1}(t) = \frac{\widehat{\psi}_{g_1}(t)}{\widehat{h}_{g_1}(t)},$$

with $\widehat{\psi}_{g_1}(t) = \frac{1}{n}\sum_{i=1}^{n} K_{g_1}(t - Z_i)\delta_i$ and $\widehat{h}_{g_2}(t) = \frac{1}{n}\sum_{i=1}^{n} K_{g_2}(t - Z_i)$.
For h', p' and p'', the derivatives of h and p are estimated by the derivatives of the same order of the corresponding kernel estimator with pilot bandwidth g_2:

$$\begin{aligned}
\widehat{h}^{(k)}_{g_2}(t) &= \frac{1}{n}\sum_{i=1}^{n} K^{(k)}_{g_2}(t - Z_i) \\
\widehat{p}'_{g_2}(t) &= \frac{\widehat{\psi}'_{g_2}(t)\,\widehat{h}_{g_2}(t) - \widehat{\psi}_{g_2}(t)\,\widehat{h}'_{g_2}(t)}{\widehat{h}^2_{g_2}(t)}, \\
\widehat{p}''_{g_2}(t) &= \frac{\widehat{\psi}''_{g_2}(t)\,\widehat{h}^2_{g_2}(t) - \widehat{\psi}_{g_2}(t)\,\widehat{h}''_{g_2}(t)\,\widehat{h}_{g_2}(t) - 2\widehat{\psi}'_{g_2}(t)\,\widehat{h}'_{g_2}(t)\,\widehat{h}_{g_2}(t) + 2\widehat{\psi}_{g_2}(t)\,\widehat{h}'_{g_2}(t)^2}{\widehat{h}^3_{g_2}(t)},
\end{aligned}$$

where $\widehat{\psi}^{(k)}_{g_2}(t) = \frac{1}{n}\sum_{i=1}^{n} K^{(k)}_{g_2}(t - Z_i)\,\delta_i$ and $K^{(k)}_{g_2}(t) = \frac{1}{g_2^{k+1}} K^{(k)}\left(\frac{t}{g_2}\right)$. The choice of g_1 and g_2 will be addressed in Section 3.3.

Turning to f and λ, the AMISE depends on two bandwidths, $\mathbf{b} = (b_1, b_2)$. Following Jácome and Cao (2007) for f and Cao and López-de-Ullibarri (2007) for λ, the AMISE is

$$AMISE_{\varphi}(\mathbf{b}) = \frac{1}{4} d_K^2 c_1^{\varphi}\left(\frac{b_1}{b_2}\right) b_2^4 + \frac{1}{n b_2} c_2^{\varphi}\left(\frac{b_1}{b_2}\right) \tag{8}$$

where c_1^{φ} and c_2^{φ} have different expressions for $\varphi = f$ and $\varphi = \lambda$:

$$\begin{aligned}
c_1^f(x) &= \int_0^{\infty} \left\{ f''(t) + 2x^2 \left((1 - F(t))\,\alpha(t) \right)' \right\}^2 \omega(t)\, dt, \\
c_2^f(x) &= \int_0^{\infty} p(t) h(t) \left(\frac{1 - F(t)}{1 - H(t)} \right)^2 \left\{ p(t)\, c_K + (1 - p(t))\, A_K(x) \right\} \omega(t)\, dt,
\end{aligned}$$

and

$$\begin{aligned}
c_1^{\lambda}(x) &= \int_0^{\infty} \left\{ (\lambda_H(t) p(t))'' + x^2 \left(\lambda_H(t) p''(t) + 2(\lambda'_H(t) - \lambda^2_H(t)) p'(t) \right) \right\}^2 \omega(t) dt, \\
c_2^{\lambda}(x) &= \int_0^{\infty} \frac{\lambda_H(t) p(t)}{1 - H(t)} \left\{ p(t) c_K + (1 - p(t)) \frac{A_K(1/x)}{x} \right\} \omega(t) dt,
\end{aligned}$$

where $\lambda_H = h/(1 - H)$ is the hazard rate of Z, $c_K = \int_{-1}^{1} K^2(t) dt$ and

$$A_K(x) = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} K(t) K(u) K(v) K(t + x(u - v)) dt du dv.$$

The AMISE bandwidths are obtained by minimizing the function in Equation 8:

$$\left(b_{1,\varphi}^{AMISE}, b_{2,\varphi}^{AMISE}\right) = \underset{(b_1, b_2) \in \mathbb{R}^+ \times \mathbb{R}^+}{\operatorname{argmin}} \; AMISE_{\varphi}(b_1, b_2).$$

It can be shown that without presmoothing (i.e., $b_1 = 0$) then $A_K(0) = \lim_{x\to\infty} x^{-1}A_K(1/x) = c_K$. As a consequence, $AMISE_\varphi$ reduces to that of the classical estimators of f and λ, and the minimization in b_2 of $AMISE_\varphi(0, b_2)$ gives the well-known plug-in bandwidth for the classical kernel estimates of f and λ (see Sánchez-Sellero, González-Manteiga, and Cao 1999).

Again, the plug-in bandwidth selector for $\mathbf{b} = (b_1, b_2)$ requires some estimates of the functions H, p, p', p'', h, h', h'', F and f'' (the last two only for $\varphi = f$) to be plugged-in into the terms c_1^φ and c_2^φ of Equation 8 and proceeds by numerically minimizing the resulting estimate of $AMISE_\varphi$. As before, our implementation makes use of the empirical estimator for H, the NW estimator and derivatives with pilot bandwidth $\widetilde{b}_1$ for p, p' and p'', and the PR estimator and derivatives with pilot bandwidth $\widetilde{b}_3$ for h, h' and h''. When $\varphi = f$, we estimate F and f using the presmoothed estimators with bandwidths $\mathbf{b} = \widetilde{b}_1$ and $\mathbf{b} = \left(\widetilde{b}_1, \widetilde{b}_2\right)$ respectively. Section 3.3 below explains the procedure we follow to choose the needed pilot bandwidths $\widetilde{b}_1, \widetilde{b}_2$ and $\widetilde{b}_3$.

3.2. Bootstrap bandwidth selector

The bootstrap bandwidth selector for $\mathbf{b}$ is obtained by minimizing a bootstrap estimate of the MISE in Equation 6 according to the following algorithm:

1. Generate B bootstrap resamples $\{Z_i^*, \delta_i^*\}_{i=1}^n$ from the original data $\{Z_i, \delta_i\}_{i=1}^n$. The resampling method must be adapted to the censored data context. Here we use the procedure called 'presmoothed simple' in Jácome *et al.* (2008), which, in general, exhibits a good practical performance:

   (a) Draw $\{Z_i^*\}_{i=1}^n$ by sampling randomly with replacement from $\{Z_i\}_{i=1}^n$.

   (b) Draw $\{\delta_i^*\}_{i=1}^n$ from the conditional Bernoulli distribution with parameter $\widehat{p}_{\widetilde{b}_1}(Z_i^*)$. Here, $\widehat{p}_{\widetilde{b}_1}(\cdot)$ is the NW estimator of p computed with the pilot bandwidth $\widetilde{b}_1$ (see Section 3.3 for pilot bandwidth selection).

2. For the jth bootstrap resample ($j = 1, \ldots, B$), compute $\widehat{\varphi}_{\mathbf{b}_l}^{P*(j)}$, the presmoothed estimator with bandwidth $\mathbf{b}_l$, $l = 1, 2, \ldots, L$, in a grid of L bandwidths.

3. With the original sample $\{Z_i, \delta_i\}_{i=1}^n$ compute the presmoothed estimator $\widehat{\varphi}_{\widetilde{\mathbf{b}}}^P$ using the pilot bandwidth $\widetilde{\mathbf{b}}$ (see Section 3.3 for pilot bandwidth selection).

4. Obtain the Monte Carlo approximation of the bootstrap version of *MISE* for each bandwidth $\mathbf{b}_l$, $l = 1, 2, \ldots, L$:

$$MISE_\varphi^*(\mathbf{b}_l) \simeq \frac{1}{B}\sum_{j=1}^{B}\int_0^\infty \left(\widehat{\varphi}_{\mathbf{b}_l}^{P*(j)}(t) - \widehat{\varphi}_{\widetilde{\mathbf{b}}}^P(t)\right)^2 \omega(t)\, dt. \tag{9}$$

5. The bootstrap bandwidth, $\mathbf{b}_\varphi^*$, is the minimizer of $MISE_\varphi^*$ over the grid of bandwidths:

$$\mathbf{b}_\varphi^* = \underset{\mathbf{b}\in\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_L\}}{\operatorname{argmin}} MISE_\varphi^*(\mathbf{b}).$$

3.3. Selection of the pilot bandwidths

As discussed above, both the bootstrap and plug-in methods require the preliminary computation of some pilot bandwidths.

Plug-in bandwidth

When the estimand φ is S or Λ, the plug-in bandwidth selector of $\mathbf{b} = b_1$ is obtained by replacing in Equation 7 the constants Q and A with the following estimates:

$$\widehat{Q}_{g_1} = \frac{1}{n}\sum_{i=1}^{n}\frac{\widehat{p}_{g_1}(Z_i)(1-\widehat{p}_{g_1}(Z_i))\omega(Z_i)}{(1-H_n(Z_i)+1/n)^2},$$

$$\widehat{A}_{g_2} = \int_0^\infty \widehat{\alpha}^2_{g_2}(v)\omega(v)dv \quad \text{with} \quad \widehat{\alpha}_{g_2}(t) = \int_0^t \frac{\frac{1}{2}\widehat{p}''_{g_2}(u)\widehat{h}_{g_2}(u) + \widehat{p}'_{g_2}(u)\widehat{h}'_{g_2}(u)}{1-H_n(u)+1/n}du.$$

Theorems 7 and 8 of Cao *et al.* (2005) give expressions for the optimal pilot bandwidths g_1 and g_2, in the sense of minimizing the asymptotic MSE of $\widehat{Q}_{g_1}$ and $\widehat{A}_{g_2}$. These bandwidths depend on some unknown functions: p, H and their first four derivatives. At this stage, we estimate g_1 and g_2 parametrically by fitting a logistic regression model for p and assuming a Weibull model for H.

In the case of $\varphi = f, \lambda$, we choose the pilot bandwidths $\tilde{b}_1, \tilde{b}_2$ and $\tilde{b}_3$ following the procedure adopted by Jácome (2005). Specifically, the first pilot bandwidth $\tilde{b}_1$, used for the NW estimates of p and its derivatives, is obtained by cross-validation (see Stone 1974). When $\varphi = f$, we use for F and f'' the corresponding presmoothed estimators with bandwidths $\mathbf{b} = b_1$ and $\mathbf{b} = \left(\widetilde{b}_1, \widetilde{b}_2\right)$ respectively, where:

$$\tilde{b}_2 = \left(\frac{c_{K''}\sum_{i=1}^{n}\left(\frac{1-\widehat{F}_n^{KM}(Z_i)}{1-H_n(Z_i)+1/n}\right)^2\delta_i\omega(Z_i)}{nd_K\int_0^\infty \widehat{f}'''(t)^2\omega(t)dt} \right)^{1/7}, \tag{10}$$

with $c_{K''} = \int_{-1}^{1} K''(t)^2dt$. This expression for the bandwidth $\tilde{b}_2$ is an estimate of the optimal bandwidth for estimating the curvature $\int_0^\infty f''(t)^2\omega(t)dt$ under censoring (see Sánchez-Sellero *et al.* 1999). The estimation of f''' in Equation 10 is not an easy matter. We use a parametric, but flexible, procedure, which fits a mixture of three Weibull distributions by maximum likelihood.

Finally, to compute the PR estimates of h and its derivatives, we use another pilot bandwidth $\tilde{b}_3$, which is essentially equivalent to $\tilde{b}_2$ in a setting without censoring:

$$\tilde{b}_3 = \left(\frac{c_{K''}}{nd_K\int_0^\infty \widehat{h}'''(t)^2\omega(t)dt} \right)^{1/7}. \tag{11}$$

The estimation of h''' in Equation 11 is carried out in a similar way to that of f''' in Equation 10.

Bootstrap bandwidth

If the estimands are S or Λ, one pilot bandwidth $\tilde{b}_1$ is required to compute the NW estimator $\widehat{p}_{\tilde{b}_1}$ and the presmoothed estimator in steps 1 and 3 of the algorithm described in Section 3.2. On the other hand, when the estimands are f or λ a second bandwidth, $\tilde{b}_2$ is required for computing $\widehat{\varphi}^P_{\tilde{\mathbf{b}}}$ in step 3 of the algorithm mentioned above.

In our implementation, $\tilde{b}_1$ is obtained by the same cross-validation procedure used in the plug-in bandwidth case. For $\tilde{b}_2$, we take:

$$\tilde{b}_2 = \left(\frac{c_K \sum_{i=1}^{n} \left(\frac{1 - \widehat{F}_n^{KM}(Z_i)}{1 - H_n(Z_i) + 1/n} \right)^2 \delta_i \omega(Z_i)}{n d_K^2 \int_0^\infty \widehat{f}''(t)^2 \omega(t) dt} \right)^{1/5}, \tag{12}$$

where f'' is estimated by the same method described for f''' in Equation 10. The bandwidth in Equation 12 corresponds to that proposed by Sánchez-Sellero *et al.* (1999) for density estimation under right censoring, and its use when $\varphi = f$ has been advocated by Jácome *et al.* (2008). Even if the use of $\tilde{b}_2$ in the case $\varphi = \lambda$ is not supported on rigorous theoretical grounds, here we use it after considering both the close relationship between the two settings and the satisfactory empirical evidence we have gathered (see Section 5.3). With simpler alternatives, like the pilot bandwidth suggested in Müller and Wang (1994) (i.e., $r/(8n_u^{0.2})$, with r a right endpoint of the support of λ and n_u the number of uncensored observations), we have observed worse results.

4. Correcting the boundary effect

When the support of $\varphi = f$ or λ has finite endpoints, both classical and presmoothed kernel estimators $\widehat{\varphi}$ may be inconsistent. Let b_2 be the smoothing bandwidth. For $0 \leq t = cb_2 < b_2$, with $c \in [0, 1)$, we have

$$E[\widehat{\varphi}_{b_2}(t)] = \varphi(t) \int_{-1}^{c} K(x)dx + o(1), \text{with} \int_{-1}^{c} K(x)dx \neq 1.$$

A similar phenomenon occurs at the right finite endpoint, say r. There is an extensive literature on how to correct this boundary effect. Among the great variety of methods available, we have chosen the boundary kernel method described in Gasser, Müller, and Mammitzsch (1985) for the density function, in Müller and Wang (1994) for the hazard rate, the latter being implemented in the R package **muhaz** (Hess and Gentleman 2010). The idea is that the presmoothed kernel estimators (4) and (5) remain invariable at the 'interior', where boundary effects do not occur, while near the endpoints the kernel K is substituted for K_t, a kernel depending on the point t, $0 \leq t < b_2$ or $r - b_2 < t \leq r$, where the estimate is to be computed. Explicit formulas for the most used boundary kernels are given in Table 1 in Müller and Wang (1994). Boundary kernels may take negative values, which leads to negative density and hazard rate estimates near endpoints. To correct this deficiency, the negative estimates are usually truncated to zero.

In our implementation the selected bandwidth **b** is the same independently of whether the boundary effect is corrected or not. This is justified by the fact that **b** is a global bandwidth chosen as the minimizer of the $MISE_\varphi$ in Equation 6, where the weight function ω discards the boundary points.

5. The survPresmooth package

This section contains a brief description of the package functionality. This is followed by the results of the analysis of a real dataset and a simulation study, both of them carried out with the package.

5.1. General description

The main function of the **survPresmooth** package is `presmooth`. This function computes the presmoothed estimates of S, Λ, f or λ, as defined in Section 2. The precise function which will be estimated when `presmooth` is called is specified through the `estimand` argument. The reader should refer to Table 1 for details on the correct way of passing values to this or other arguments of `presmooth`. For every estimand, the plug-in or bootstrap bandwidths described in Section 3 can be computed. The bandwidth selection method used is specified by the value of the `bw.selec` argument. Besides, the estimation can also be carried out with an arbitrarily chosen bandwidth, whose value must then be passed to the `fixed.bw` argument. In this case, when the presmoothing bandwidth is set to zero, one gets classical, non-presmoothed estimates. In fact, the function provides an alternative way of getting non-presmoothed estimates, through the `presmoothing` argument (see also Table 1 and the next subsection). Although the default estimates computed by `presmooth` are not corrected for possible boundary effects, in the case of f and λ estimation the `bound` argument makes it possible to apply the technique for boundary effect correction discussed in Section 4 at one or both endpoints.

The additional arguments of `presmooth` are also listed and briefly described in Table 1. Their role covers a variety of aspects like data input (`times`, `status` and `dataset` arguments), choice of kernel function (`kernel` argument) and specification of some grids of bandwidths (`grid.bw.pil` and `grid.bw` arguments), characteristics of the output (`x.est` argument) or control parameters (`control` argument).

The standard way of passing values to the `control` argument is by assigning to it the output of a call to the secondary function `control.presmooth`. This function's arguments are related to a series of factors controlling details of the computation of the presmoothed estimators. One of them is the weight function ω, which, as commented in Section 3, is an indicator function in our implementation. The endpoints of the support of ω are specified via the `q.weight` argument of `control.presmooth`. Another influential factor in bootstrap bandwidth selection is the number B of bootstrap resamples taken to compute the MISE in Equation 9 on a grid of bandwidths (incidentally, the grid itself may be set with the argument `grid.bw` of `presmooth`). The value of B is set with the `n.boot` argument of `control.presmooth`. Also, the MISE values can be saved by means of the `save.mise` argument. Thus, e.g., the user can plot the MISE values against the bandwidths to inspect the MISE function (the reader is referred to the help of the `presmooth` function, where he can find some examples). Section 5.2 contains an example illustrating how `control.presmooth` is used.

Argument	Description
`times`	An object of mode `numeric` giving the observed times. If `dataset` is not NULL it is interpreted as the name of the corresponding variable of the dataset.
`status`	An object of mode `numeric` giving the censoring status of the times coded in the `times` object. If `dataset` is not NULL it is interpreted as the name of the corresponding variable of the dataset.
`dataset`	A data frame in which the variables named in `times` and `status` are interpreted. If NULL, `times` and `status` must be objects of the workspace.
`estimand`	A character string identifying the function to estimate: `"S"`, the default, for S, `"H"` for Λ, `"f"` for f and `"h"` for λ.
`bw.selec`	A character string specifying the bandwidth selection method: `"fixed"`, the default, if no bandwidth selection is done, `"plug-in"` for plug-in bandwidth selection and `"bootstrap"` for bootstrap bandwidth selection.
`presmoothing`	A logical value indicating if the presmoothed estimates (`TRUE`, the default) or their non-presmoothed counterparts (`FALSE`) will be computed.
`fixed.bw`	A numeric vector with the fixed bandwidth(s) used when the value of the `bw.selec` argument is `"fixed"`. It has length 1 for estimating S and Λ, or 2 for f and λ (then, the first element is the presmoothing bandwidth b_1).
`grid.bw.pil`	A numeric vector specifying the grid where the presmoothing pilot bandwidth will be selected using the cross-validation method. Not used in plug-in bandwidth selection for S or Λ estimation.
`grid.bw`	A list of length 1 (for S or Λ estimation) or 2 (for f and λ estimation) whose component(s) is (are) a (two) numeric vector(s) specifying the grid of bandwidths needed for bootstrap bandwidth selection when the value of the `bw.selec` argument is `"bootstrap"`. For S or Λ estimation, it can also be a numeric vector.
`kernel`	A character string specifying the kernel function used. One of `"biweight"`, for biweight kernel (the default), and `"triweight"`, for triweight kernel.
`bound`	A character string specifying the end(s) of the data range where boundary correction is applied. If `"none"`, the default, no correction is done; if `"left"`, `"right"` or `"both"`, the correction is applied at the left, right or both ends.
`x.est`	A numeric vector specifying the points where the estimate is computed.
`control`	A list of control values. The default value is the output returned by the `control.presmooth` function called without arguments.

Table 1: Arguments of the `presmooth` function and their description.

The output produced by `presmooth` is a list of class **survPresmooth**. The package implements a method for printing objects of this class, which by default (i.e., when the object name is entered in the command line) performs only a minimal formatting of the output. In Section 5.2, an example showing how to call explicitly the print method is given.

From a computational point of view, although R is the programming environment for the package, for efficiency reasons the main function (i.e., `presmooth`) makes extensive use of compiled C code.

5.2. Application to a real dataset

Here we present an analysis of a dataset taken from Klein and Moeschberger (2003). This is the `alloauto` dataset included as part of the R package **KMsurv** (Klein, Moeschberger, and Yan 2012). It collects information about a sample of 101 patients with acute myelogenous leukemia reported to the International Bone Marrow Transplant Registry. All patients received a bone marrow transplantation, but they may differ with respect to its type: allogeneic (ALLO) or autologous (AUTO). It should be clear that our purpose when analyzing this dataset is only to illustrate the functionality of the package through a real example, not to answer any substantive questions about the data itself.

In this dataset, event (i.e., death or relapse) times may be right censored by end of follow-up. The incidence of censoring is moderate (50.5%), slightly higher in the ALLO group (56.0%) than in the AUTO group (45.1%). The variables in data frame `alloauto` are: `time`, the time (months) to death or relapse; `delta`, an indicator of death or relapse (0 = alive without relapse, 1 = death or relapse); and `type`, the type of transplant (1 = ALLO, 2 = AUTO). A total of 50 patients had ALLO and 51 AUTO transplants.

Before starting our analysis, we create one separate R object for each group of patients.

```
R> library("KMsurv")
R> data("alloauto")
R> allo <- alloauto[alloauto$type == 1, c("time", "delta")]
R> auto <- alloauto[alloauto$type == 2, c("time", "delta")]
```

Next, it is shown how to use the `presmooth` function to obtain estimates of the functions that characterize the survival time for each of the two groups defined by type of transplant:

```
R> library("survPresmooth")
R> allo.S.pi <- presmooth(times = time, status = delta, dataset = allo,
+    estimand = "S", bw.selec = "plug-in")
R> allo.H.pi   <- presmooth(time, delta, allo, "H", "plug-in")
R> allo.S.boot <- presmooth(time, delta, allo, "S", "bootstrap")
R> allo.H.boot <- presmooth(time, delta, allo, "H", "bootstrap")
R> auto.S.pi   <- presmooth(time, delta, auto, "S", "plug-in")
R> auto.H.pi   <- presmooth(time, delta, auto, "H", "plug-in")
R> auto.S.boot <- presmooth(time, delta, auto, "S", "bootstrap")
R> auto.H.boot <- presmooth(time, delta, auto, "H", "bootstrap")
```

As can be seen from the code, the identity of the curve which is estimated and the bandwidth selection method used are determined by the values passed to the `estimand` and `bw.selec` arguments, respectively. Let us point out that the program sets an upper bound equal to the range of the observed times for any selected bandwidth.

For comparison reasons, it is interesting to obtain the KM and NA estimates for the two groups of patients. As mentioned before, these classical estimators are recovered from the corresponding presmoothed estimators when the presmoothing bandwidth b_1 is zero. With the `presmooth` function this can be done by setting the `bw.selec` argument to `"fixed"` (actually, this is the default value) and the `fixed.bw` argument to `0`:

		Presmoothing bandwidth b_1		Smoothing bandwidth b_2	
Group	Estimand	Plug-in	Bootstrap	Plug-in	Bootstrap
ALLO	S, Λ	4.51	6.06	–	–
	f	8.46	8.56	6.63 (6.87)	10.78
	λ	7.59	6.06	3.91 (4.41)	12.09
AUTO	S, Λ	17.53	7.83	–	–
	f	12.26	11.06	13.89 (14.05)	22.07
	λ	14.60	9.86	12.05 (11.96)	24.76

Table 2: Selected bandwidths for the `alloauto` dataset. The bandwidths between parentheses correspond to the non-presmoothed estimates shown in Figure 2 (see text for details).

```
R> allo.km <- presmooth(time, delta, allo, "S", "fixed", fixed.bw = 0)
R> allo.na <- presmooth(time, delta, allo, "H", "fixed", fixed.bw = 0)
R> auto.km <- presmooth(time, delta, auto, "S", "fixed", fixed.bw = 0)
R> auto.na <- presmooth(time, delta, auto, "H", "fixed", fixed.bw = 0)
```

An alternative method of obtaining these non-presmoothed estimates consists in passing the value `FALSE` to the argument `presmoothing`. For example, `allo.km` could also be computed by

```
R> presmooth(time, delta, allo, "S", presmoothing = FALSE)
```

Figure 1 is a plot of the estimates of the S and Λ functions. It is easily drawn from the objects created by the previous code (i.e., from the information contained in their components `x.est` and `estimate`), by using R's basic plotting facilities. For example, the top left panel is produced by executing:

```
R> plot(allo.S.pi$x.est, allo.S.pi$estimate, type = "s", xlab = "Time",
+    ylab = "Survival", ylim = c(0, 1), main = "Allogeneic transplant",
+    col = "blue")
R> lines(allo.S.boot$x.est, allo.S.boot$estimate, type = "s", col = "red")
R> lines(allo.km$x.est, allo.km$estimate, type = "s", lty = "dotted")
```

A general comparison of the different estimates of Figure 1 reveals mainly minor small-scale differences. As expected, the presmoothed estimates are characterized by jumps that are smaller and more frequent than in the corresponding empirical estimates. This reflects the fact that the presmoothed estimates carry more information on the local behavior of the lifetime distribution. Only in the case of the AUTO group with plug-in bandwidth, striking, large-scale differences affecting the right tail of the estimates are observed. Of course, all these facts are determined by the specific values of the bandwidths, which are collected in Table 2.

The selected bandwidths are saved in the `bandwidth` component of the objects of class **survPresmooth**. They are printed by default by the print method for objects of the class. If a formatted output including other components of the **survPresmooth** object is needed, the `print.survPresmooth` function must be explicitly called, with the name of the component(s) assigned to the `more` argument. For example, to print the pilot bandwidths:

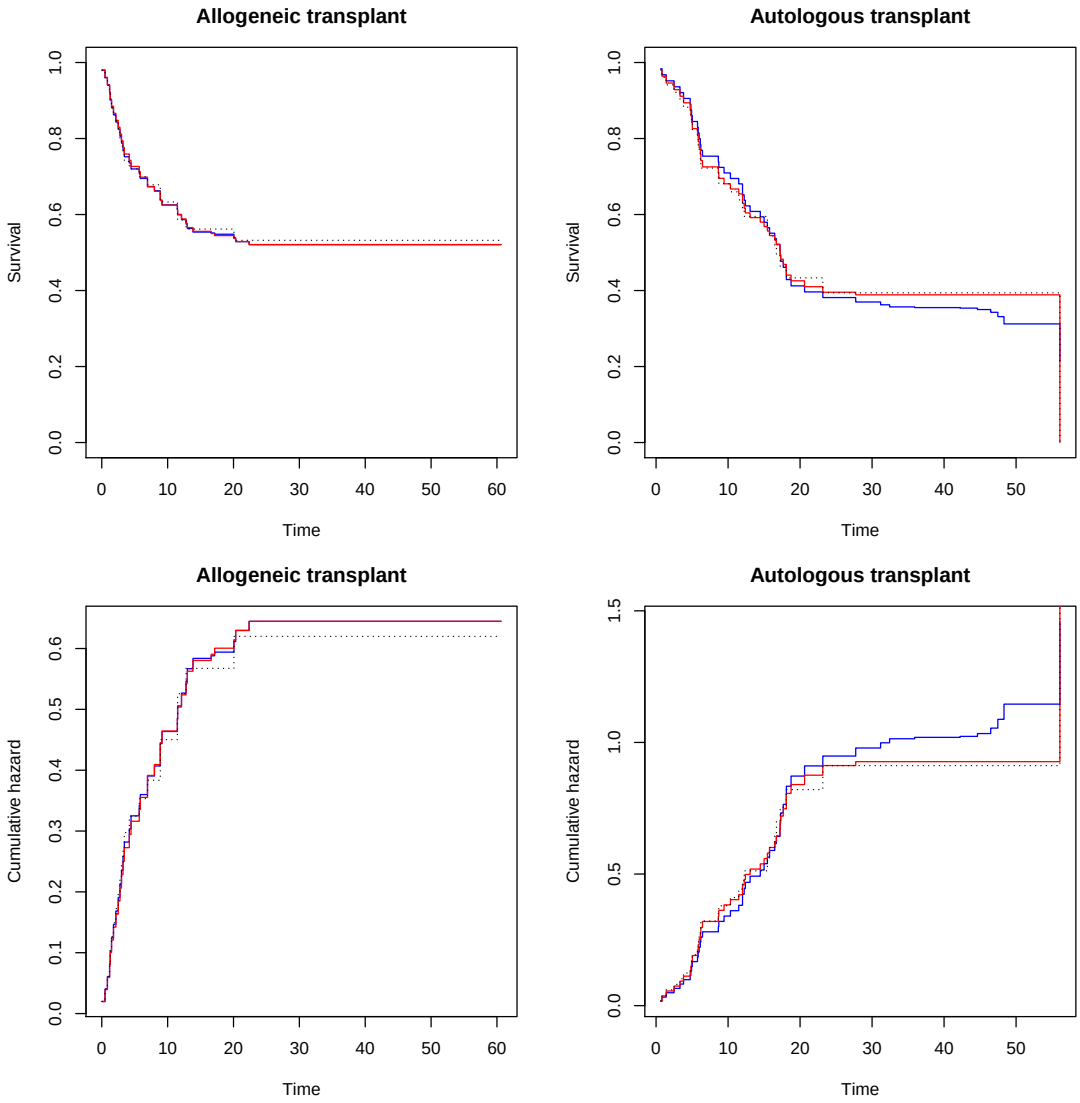


Figure 1: `alloauto` dataset. Estimates of S (top panels) and Λ (bottom panels), conditioned by type of transplant. The presmoothed estimates were obtained with either plug-in (blue lines) or bootstrap (red lines) bandwidth selection. Also shown are the KM and NA estimates of S and Λ, respectively (dotted black lines).

```
R> print(allo.S.pi, more = "pilot.bw")

Presmoothed estimation of the survival function, S(t)

        t      S(t)
1   0.030 0.9800045
2   0.493 0.9601144

....
```

```
49 58.322 0.5208789
50 60.625 0.5208789

Bandwidth selection method: plug-in

Bandwidth(s):
        presmoothing:         4.510372

Pilot bandwidth(s):
[1] 5.612775 8.989902
```

As for the f and λ functions, Figure 2 provides a plot of their presmoothed estimates. The selected plug-in and bootstrap bandwidths are also collected in Table 2. The bootstrap selector seems to give slightly large smoothing bandwidths b_2, which entails smoother estimations than those with the plug-in bandwidth selection. We also show how the estimates change depending on whether the boundary effect is corrected or not.

Here we only give details on the R code run to get the estimates displayed on Figure 2 for the case of f estimation in the ALLO group:

```
R> allo.f.pi <- presmooth(time, delta, allo, "f", "plug-in")
R> allo.f.boot <- presmooth(time, delta, allo, "f", "bootstrap")
R> allo.f.pi.bound <- presmooth(time, delta, allo, "f", "plug-in",
+    bound = "both")
R> allo.f.boot.bound <- presmooth(time, delta, allo, "f", "bootstrap",
+    bound = "both")
```

The estimates are computed at the points given by the `x.est` argument (see Table 1). When, as in the previous lines of code, its value is not explicitly set, `presmooth` computes it internally. With the default value of `x.est`, estimation is done at a sequence of 50 equispaced points between the minimum and the 90th percentile of the observed times. As a guideline, density and hazard estimates at the right tail should be taken very cautiously due to their increased bias and variance.

A warning should be given about computing time, which is usually markedly longer for bootstrap than for plug-in bandwidth selection. Of course, this difference is due to the computer-intensive nature of bootstrap methods. On a machine with an Intel Core i7-3610QM processor and 7.7 GB of memory, the last two lines of code took respectively 3.372 and 14.857 seconds of CPU time.

Our bandwidth selectors for f and λ can be extended to the case without presmoothing, allowing the selection of plug-in and bootstrap smoothing bandwidths for the corresponding classical kernel estimators of these curves. For reference, the classical non-presmoothed estimates of f and λ thus obtained (with plug-in bandwidth selection) have been added to Figure 2 (and the values of the corresponding bandwidths to Table 2). For f, this estimate is computed by:

```
R> allo.f.pi.np <- presmooth(time, delta, allo, "f", "plug-in",
+    presmoothing = FALSE)
```

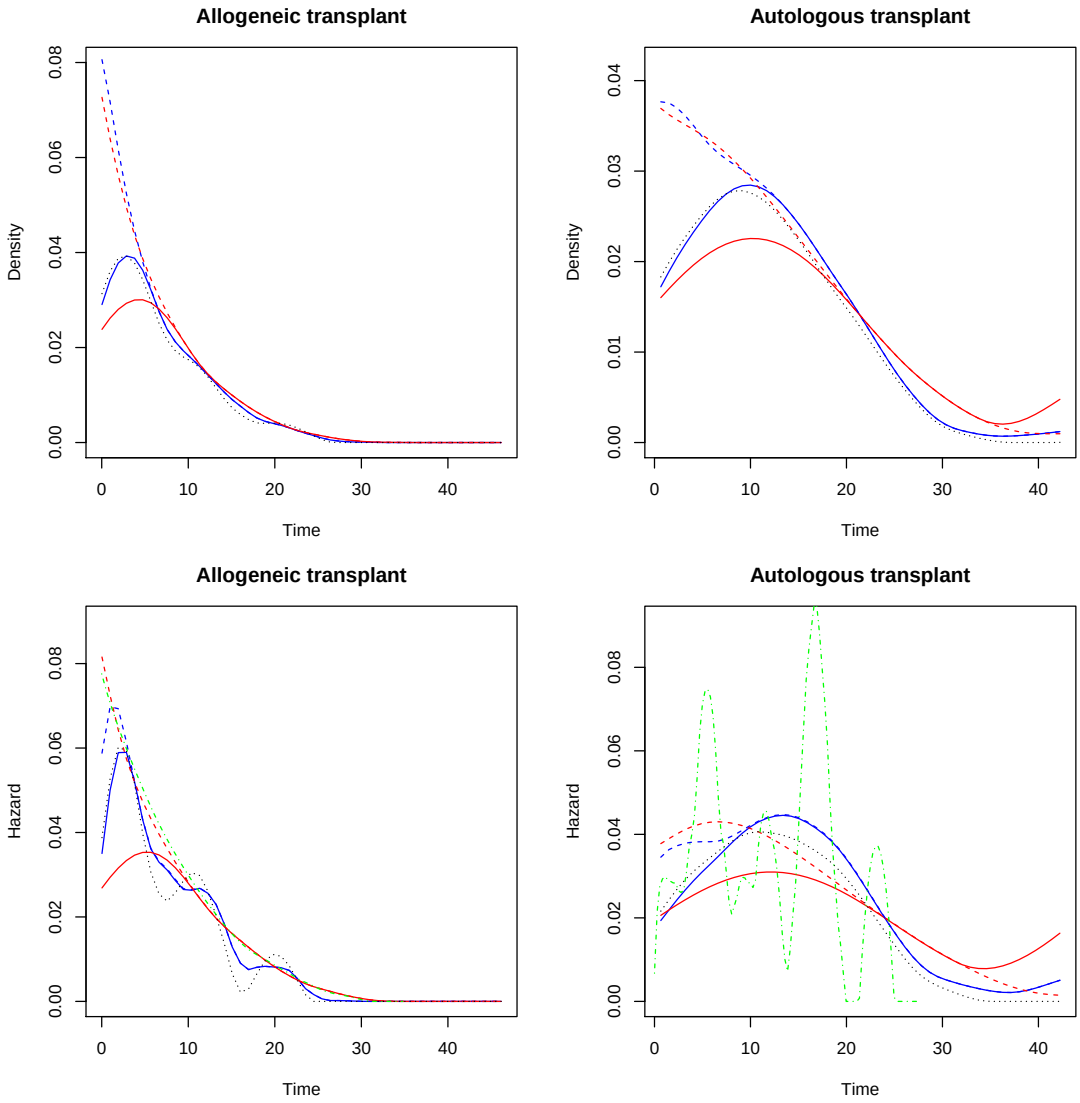


Figure 2: `alloauto` dataset. Estimates of f (top panels) and λ (bottom panels), conditioned by type of transplant. Estimates were obtained with either plug-in (blue lines) or bootstrap (red lines) bandwidth selection, and without (solid lines) or with (dashed lines) correction of the boundary effect. The dotted black lines are non-presmoothed plug-in estimates of f and λ obtained with **survPresmooth**. The dotted-dashed green lines are alternative estimates of λ computed with the R package **muhaz**.

For λ, the plot also shows the hazard estimates obtained with the `muhaz` function in R package **muhaz**, using the default settings for global bandwidth selection (local bandwidth selection, also possible with **muhaz**, is currently not available in **survPresmooth**). Note the clearly undersmoothed shape of the resulting hazard estimate in the AUTO group.

```
R> library("muhaz")
R> allo.muhaz <- muhaz(allo$time, allo$delta, bw.method = "global")
```

	T		C		
Model	α_T	β_T	α_C	β_C	π
I	1	4	1	5	0.48
II	1	0.7	0.25	0.9	0.73
III	1	4	0.8	4	0.71

Table 3: Characteristics of the simulated models I, II and III.

Further aspects of the computation of the presmoothed estimates of S, Λ, f or λ can be fine-tuned by means of other arguments, including the `control` argument and the associated `control.presmooth` function. For example, the following code would compute the presmoothed estimate of S for the AUTO group with bootstrap bandwidth selected from a grid of 150 equispaced bandwidths between 1 and 50, taking $B = 10000$ bootstrap resamples, and a weight function with support on the interval defined by the 10th and 90th percentiles of the observed times:

```
R> presmooth(time, delta, auto, "S", "bootstrap",
+    grid.bw = seq(1, 50, length.out = 150),
+    control = control.presmooth(n.boot = 10000, q.weight = c(0.1, 0.9)))
```

5.3. Simulations

The practical performance of the presmoothed estimators and bandwidth selectors implemented in **survPresmooth** may be shown by means of simulation experiments. We have simulated four different models in order to describe the behavior in (non-cumulative and cumulative) hazard function estimation with varying sample size. For the sake of brevity, we do not give any results for survival and density functions. The models we have simulated try to define scenarios showing different combinations of purportedly influential conditions, like the intensity of censoring, the constant or non-constant nature of the p function, and the increasing, decreasing or non-monotonic nature of the hazard function.

In models I, II and III, both survival and censoring times follow a Weibull distribution with hazard function:

$$\lambda(t) = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1}, \quad t > 0,$$

where α and β are the scale and shape parameters.

The parameters characterizing the survival and censoring times of these models are collected in Table 3. Also shown is the value of the unconditional probability of censoring $\pi = 1 - \int_0^\infty p(t)h(t)dt$, where h is the density of Z.

For model IV we have considered the distribution proposed by Chen (2000). For parameters $\alpha > 0$, $\beta > 0$, Chen's hazard function is

$$\lambda(t) = \alpha\beta t^{\beta-1}\exp(t^\beta), \quad t > 0.$$

It can be shown that λ has a bathtub shape for $\beta < 1$ and it is an increasing function for $\beta \geq 1$ (Chen 2000). In model IV, the survival and censoring times have Chen distributions with

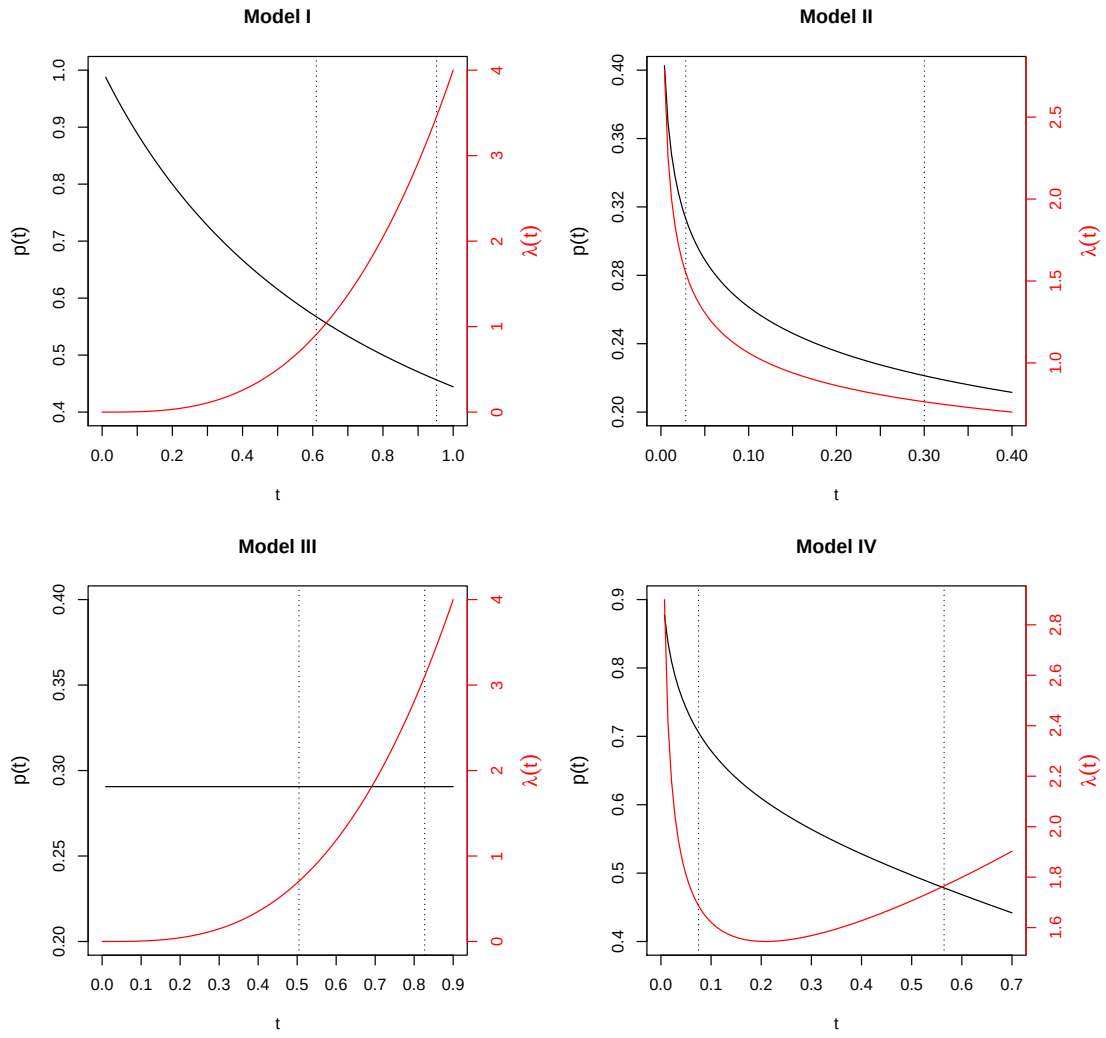


Figure 3: Graphs of p (black) and λ (red) for the simulated models. The dotted vertical lines identify the 20th and 80th quantiles of the observed time, which are the endpoints of the default weight function used for bandwidth selection by **survPresmooth**.

$\alpha = 1$ and β parameter equal to 0.7 and 1.2, respectively. For this choice, the unconditional probability of censoring is 0.41. Plots of the p and λ curves of models I–IV can be found in Figure 3.

A total of 500 independent pseudorandom samples have been drawn from each model for small ($n = 30$), moderate ($n = 150$) and large ($n = 3000$) sample sizes. For each sample, presmoothed and non-presmoothed estimates of Λ and λ have been computed using, where applicable, our plug-in and bootstrap bandwidth selectors (actually, for $n = 3000$, due to computational burden, our experimentation has excluded the bootstrap bandwidth selector). For each simulated sample the integrated squared error (ISE) has been approximated by Simpson's rule for numerical integration. For any bandwidth selector, let us denote by ISE_P the ISE of a presmoothed estimate and by ISE_{NP} that of the corresponding non-presmoothed estimate. We have computed the ratio of ISEs ISE_{NP}/ISE_P as a measure of the relative

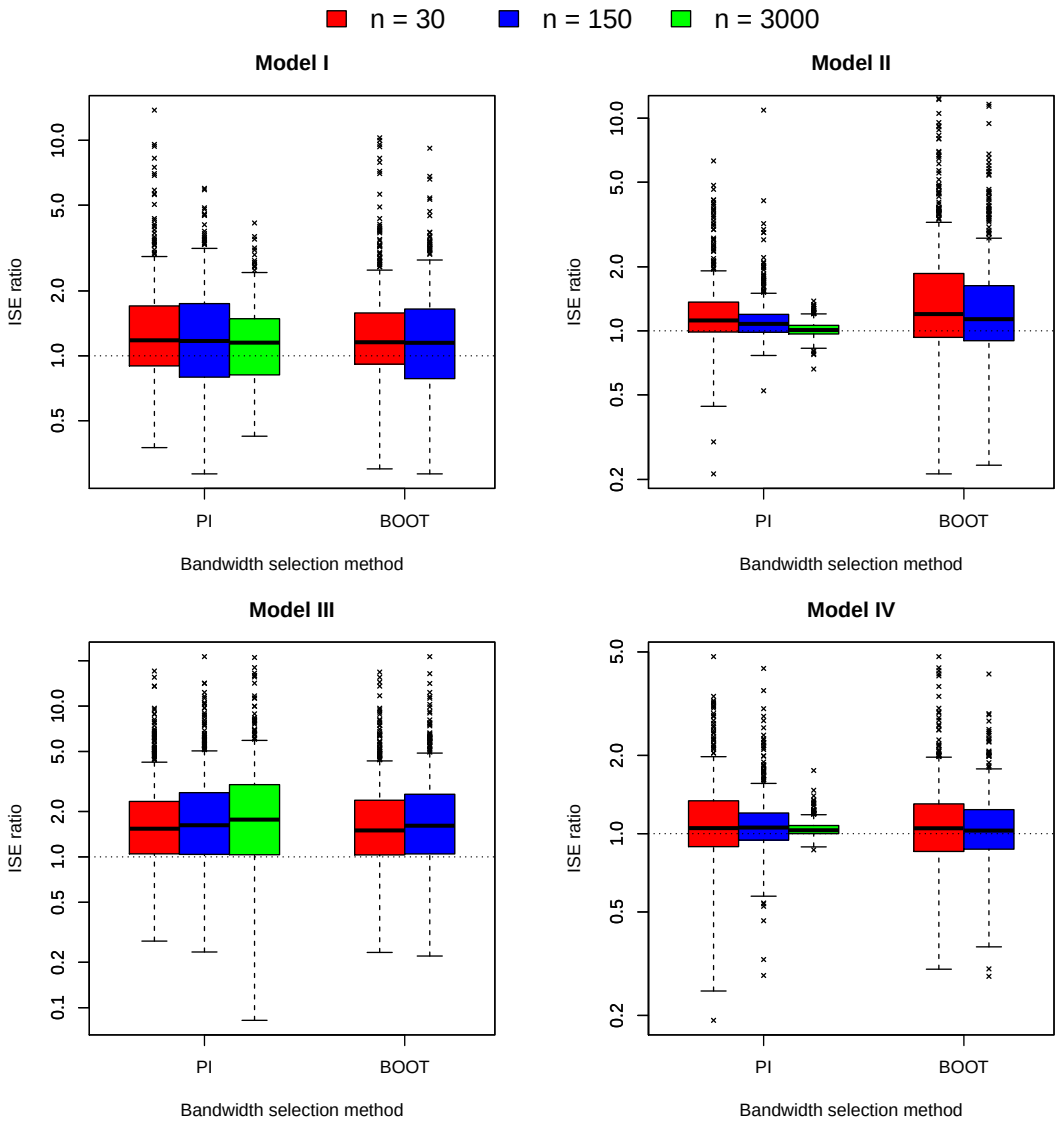


Figure 4: Simulation results: box plots of the ISE_{NP}/ISE_P ratios for the non-presmoothed and presmoothed estimates of Λ (for notation, see text). PI: plug-in bandwidth; BOOT: bootstrap bandwidth.

efficiency of presmoothed and non-presmoothed estimators. When ISE_{NP}/ISE_P takes a value, say r, greater than 1, presmoothing is more efficient for that sample; more specifically, the presmoothed estimator is then r times more efficient than the non-presmoothed one.

The box plots of the sampling distributions of the ISE ratios under the different simulated scenarios are shown in Figure 4 for the case of Λ estimation, and in Figure 5 for λ. In these plots, a logarithm scale has been used to facilitate comparison. The numerical values of the medians of the ISE ratios have been collected in Table 4. It is observed that, whatever the bandwidth selector chosen, for most of the simulated scenarios the presmoothed estimators are more efficient than the non-presmoothed ones. This is more striking for Model *III*; the

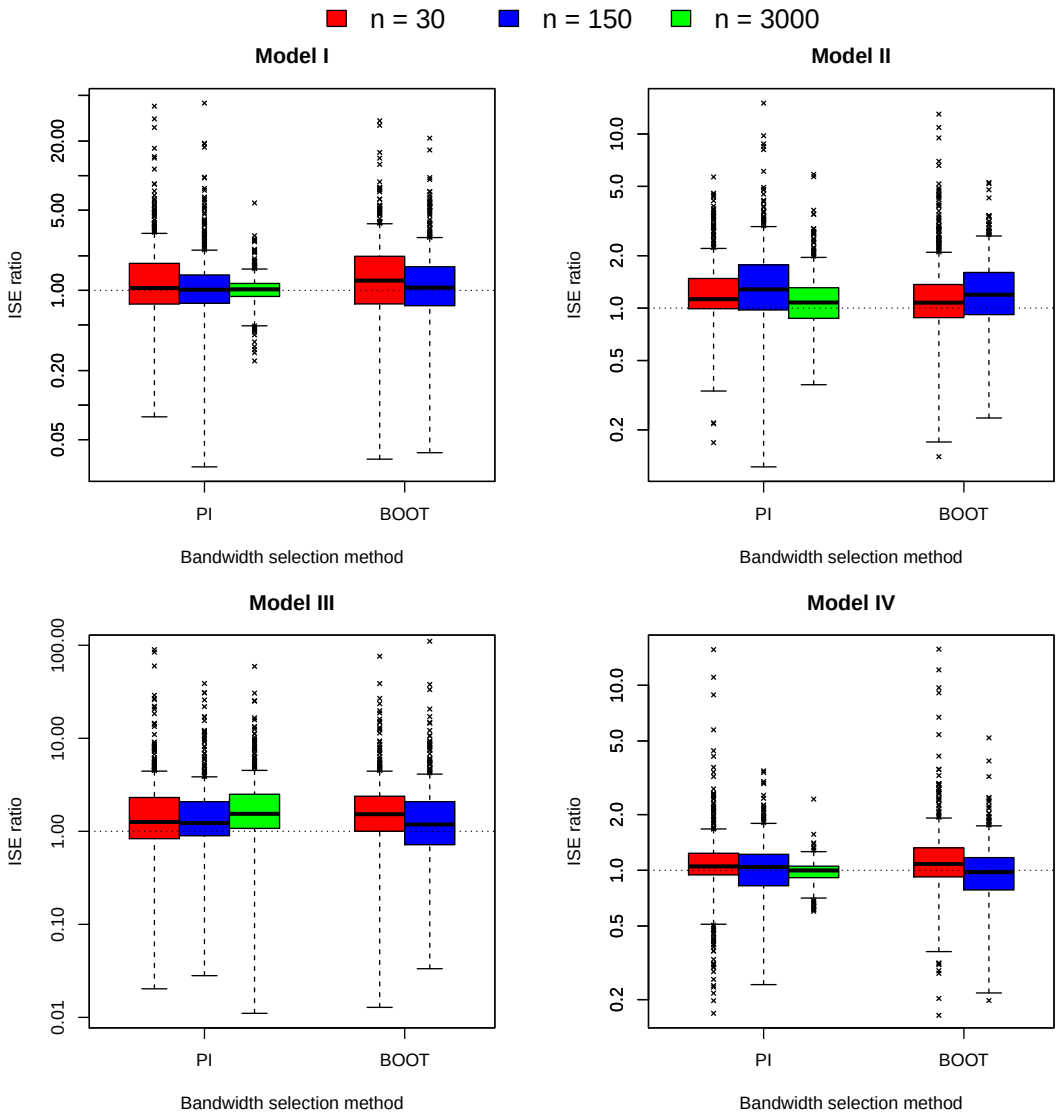


Figure 5: Simulation results: box plots of the ISE_{NP}/ISE_P ratios for the non-presmoothed and presmoothed estimates of λ. Notation is the same as in Figure 4.

reason is that the p function of this model is constant, a condition where first order efficiency is attained (see Cao and Jácome 2004). As expected, the differences between both approaches tend generally to balance as n increases, but quite slowly, with the presmoothed estimators still being more efficient for $n = 3000$ in a majority of scenarios. The exception to this pattern is again Model *III*, where the ISE ratio seems to increase with n. This is hardly surprising since, as noted before, this model simulates a first order efficiency scenario. Overall, these results demonstrate the convenience of presmoothing, and the usefulness of the **survPresmooth** package for analyzing right censored data.

		Plug-in			Bootstrap	
Estimand	Model	$n = 30$	$n = 150$	$n = 3000$	$n = 30$	$n = 150$
Λ	*I*	1.180	1.172	1.151	1.156	1.149
	II	1.119	1.078	1.009	1.199	1.135
	III	1.538	1.621	1.766	1.498	1.610
	IV	1.051	1.056	1.031	1.049	1.028
λ	*I*	1.049	1.014	1.021	1.216	1.057
	II	1.125	1.280	1.077	1.075	1.194
	III	1.261	1.227	1.542	1.528	1.186
	IV	1.050	1.042	0.998	1.082	0.977

Table 4: Simulation results: medians of the ISE_{NP}/ISE_P ratios for the non-presmoothed and presmoothed estimates of Λ and λ (for notation, see text).

6. Conclusions

This paper deals mainly with the implementation in R of the presmoothed estimators of the survival, density, and cumulative and non-cumulative hazard functions of a right-censored lifetime. The new R package **survPresmooth** is introduced and described. Also, the theory underlying presmoothing has been summarized and further evidence showing the advantages of presmoothed estimators over their classical counterparts has been provided. The `presmooth` function of the package computes the presmoothed estimators in a user-friendly way. The function also implements two different methods for computing of the required bandwidths, based on bootstrap and plug-in techniques. Additionally, our software allows to compute well-known classical, non-presmoothed estimators (including, where applicable, their bandwidths), which may be interpreted as particular cases of presmoothed estimators.

There are several topics that are not dealt with by our package. We close the discussion with an enumeration of some of these issues, which give the opportunity for future developments of the package.

Although initially the graphical comparison of two or more distributions (straightforwardly done with **survPresmooth**) may be enough, hypothesis testing of the equality of survival distributions is more satisfactory from a statistical point of view. It is possible to adapt the log-rank test and, in general, all the weighted tests in the literature to the use of presmoothed estimators. However, these "presmoothed tests" remain largely unexplored and they should be carefully worked out before being implemented.

Our package does not provide confidence bands for the estimated functions. A way of constructing them could be based on the bootstrap. The same resampling plan used for bootstrap bandwidth selection could be applied in order to compute the percentiles of the bootstrap distribution of the estimates. The limits of pointwise confidence intervals could be constructed from these percentiles.

Sometimes, in addition to right censoring (RC), lifetimes are also subject to left truncation (LT). The good properties of presmoothing are conserved in the so-called LTRC model: see Jácome and Iglesias-Pérez (2008) for the case of S and Λ estimation, and Jácome and Iglesias-Pérez (2010) for f. This suggests that, in principe, the procedures implemented in **surPresmooth** could also be extended to include LTRC data.

Another issue not considered in **survPresmooth** is the possible presence of covariates. Presmoothing ideas are relatively new, and though survival analysis adjusting for covariates is of great interest, it has been scarcely investigated in the context of presmoothed estimation. For a semiparametric approach see de Uña-Álvarez and Rodríguez-Campos (2004) and Iglesias-Pérez and de Uña-Álvarez (2008).

Finally, let us point out that the properties of presmoothed estimators have been studied only in the setting of independent data, but in some studies survival times may be dependent. Under rather weak conditions for dependence, the KM estimator is still consistent and asymptotically normal (Ying and Wei 1994; Cai 1998). Similar ideas could be applied to try to prove that properties regarding consistency and asymptotic normality of the presmoothed estimators are also valid under the same weak conditions for dependence.

Acknowledgments

This research has been partially supported by the Spanish Ministry of Science and Innovation (Grant MTM2011-22392).

References

Aalen OO (1978). "Nonparametric Inference for a Family of Counting Processes." *The Annals of Statistics*, **6**, 701–726.

Cai Z (1998). "Asymptotic Properties of Kaplan-Meier Estimator for Censored Dependent Data." *Statistics & Probability Letters*, **37**, 381–389.

Cao R, Jácome MA (2004). "Presmoothed Kernel Density Estimator for Censored Data." *Journal of Nonparametric Statistics*, **16**, 289–309.

Cao R, López-de-Ullibarri I (2007). "Product-Type and Presmoothed Hazard Rate Estimators with Censored Data." *Test*, **16**, 355–382.

Cao R, López-de-Ullibarri I, Janssen P, Veraverbeke N (2005). "Presmoothed Kaplan-Meier and Nelson-Aalen Estimators." *Journal of Nonparametric Statistics*, **17**, 31–56.

Chen Z (2000). "A New Two-Parameter Lifetime Distribution with Bathtub Shape or Increasing Failure Rate Function." *Statistics & Probability Letters*, **49**, 155–161.

de Uña-Álvarez J, Rodríguez-Campos MC (2004). "Strong Consistency of Presmoothed Kaplan-Meier Integrals when Covariables Are Present." *Statistics*, **38**, 483–496.

Dikta G (1998). "On Semiparametric Random Censorship Models." *Journal of Statistical Planning and Inference*, **66**, 253–279.

Dikta G (2000). "The Strong Law under Semiparametric Random Censorship Models." *Journal of Statistical Planning and Inference*, **83**, 1–10.

Dikta G (2001). "Weak Representation of the Cumulative Hazard Function under Semiparametric Censorship Models." *Statistics*, **35**, 395–409.

Földes A, Rejtö L, Winter BB (1981). "Strong Consistency Properties of Nonparametric Estimators for Randomly Censored Data. II Estimation of Density and Failure Rate." *Periodica Mathematica Hungarica*, **12**, 15–29.

Gasser T, Müller HG (1979). "Kernel Estimation of Regression Functions." In T Gasser, M Rosenblatt (eds.), *Smoothing Techniques for Curve Estimation*, volume 757 of *Lecture Notes in Mathematics*, pp. 23–68. Springer-Verlag.

Gasser T, Müller HG, Mammitzsch V (1985). "Kernels for Nonparametric Curve Estimation." *Journal of the Royal Statistical Society B*, **47**, 238–252.

Hess K, Gentleman R (2010). ***muhaz**: Hazard Function Estimation in Survival Analysis*. R package version 1.2.5, URL http://CRAN.R-project.org/package=muhaz.

Iglesias-Pérez MC, de Uña-Álvarez J (2008). "Nonparametric Estimation of the Conditional Distribution Function in a Semiparametric Censorship Model." *Journal of Statistical Planning and Inference*, **138**, 3044–3058.

Jácome MA (2005). *Estimación Presuavizada de las Funciones de Densidad y Distribución con Datos Censurados*. Ph.D. thesis, Universidade da Coruña.

Jácome MA, Cao R (2007). "Almost Sure Asymptotic Representation for the Presmoothed Distribution and Density Estimators for Censored Data." *Statistics*, **41**, 517–534.

Jácome MA, Cao R (2008). "Strong Representation of the Presmoothed Quantile Function Estimator for Censored Data." *Statistica Neerlandica*, **62**, 425–440.

Jácome MA, Gijbels I, Cao R (2008). "Comparison of Presmoothing Methods in Kernel Density Estimation under Censoring." *Computational Statistics*, **23**, 381–406.

Jácome MA, Iglesias-Pérez MC (2008). "Presmoothed Estimation with Left-Truncated and Right-Censored Data." *Communications in Statistics – Theory and Methods*, **37**, 2964–2983.

Jácome MA, Iglesias-Pérez MC (2010). "Presmoothed Estimation of the Density Function with Truncated and Censored data." *Statistics*, **44**, 217–234.

Kaplan EL, Meier P (1958). "Nonparametric Estimation from Incomplete Observations." *Journal of the American Statistical Association*, **53**, 457–481.

Klein JP, Moeschberger ML (2003). *Survival Analysis: Techniques for Censored and Truncated Data*. Springer-Verlag.

Klein JP, Moeschberger ML, Yan J (2012). ***KMsurv**: Data Sets from Klein and Moeschberger (1997), Survival Analysis*. R package version 0.1-5, URL http://CRAN.R-project.org/package=KMsurv.

López-de-Ullibarri I, Jácome MA (2013). ***survPresmooth**: Presmoothed Estimation in Survival Analysis*. R package version 1.1-8, URL http://CRAN.R-project.org/package=survPresmooth.

Müller HG, Wang JL (1994). "Hazard Rate Estimation under Random Censoring with Varying Kernels and Bandwidths." *Biometrics*, **50**, 61–76.

Nadaraya EA (1964). "On Estimating Regression." *Theory of Probability and Its Applications*, **10**, 186–190.

Nelson W (1972). "Theory and Applications of Hazard Plotting for Censored Failure Data." *Technometrics*, **14**, 945–965.

Parzen E (1962). "On Estimation of a Probability Density Function and Mode." *The Annals of Mathematical Statistics*, **33**, 1065–1076.

Ramlau-Hansen H (1983). "Smoothing Counting Process Intensities by Means of Kernel Functions." *The Annals of Statistics*, **11**, 453–466.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Rosenblatt M (1956). "Remarks on Some Nonparametric Estimates of a Density Function." *The Annals of Mathematical Statistics*, **27**, 832–837.

Sánchez-Sellero C, González-Manteiga W, Cao R (1999). "Bandwidth Selection in Density Estimation with Truncated and Censored Data." *Annals of the Institute of Statistical Mathematics*, **51**, 51–70.

Stone M (1974). "Cross-Validatory Choice and Assessment of Statistical Predictions." *Journal of the Royal Statistical Society B*, **36**, 111–147.

Tanner MA, Wong WH (1983). "The Estimation of the Hazard Function from Randomly Censored Data by the Kernel Method." *The Annals of Statistics*, **11**, 989–993.

Watson GS (1964). "Smooth Regression Analysis." *Shankya A*, **26**, 359–372.

Yandell BS (1983). "Nonparametric Inference for Rates with Censored Data." *The Annals of Statistics*, **11**, 1119–1135.

Ying Z, Wei LJ (1994). "The Kaplan-Meier Estimate for Dependent Failure Time Observations." *Journal of Multivariate Analysis*, **50**, 17–29.

Affiliation:

Ignacio López-de-Ullibarri
Departamento de Matemáticas
Universidade da Coruña
Escuela Universitaria Politécnica
Ferrol, A Coruña, Spain
E-mail: ilu@udc.es

M. Amalia Jácome
Departamento de Matemáticas
Universidade da Coruña
Facultad de Ciencias
A Coruña, Spain
E-mail: majacome@udc.es

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 54, Issue 11 *Submitted:* 2011-03-21
August 2013 *Accepted:* 2013-03-13