Journal of Statistical Software

November 2010, Volume 37, Issue 7. *http://www.jstatsoft.org/*

DTDA: An R Package to Analyze Randomly Truncated Data

Carla Moreira
University of Vigo

Jacobo de Uña-Álvarez
University of Vigo

Rosa M. Crujeiras
University of Santiago de Compostela

Abstract

In this paper, the R package **DTDA** for analyzing truncated data is described. This package contains tools for performing three different but related algorithms to compute the nonparametric maximum likelihood estimator of the survival function in the presence of random truncation. More precisely, the package implements the algorithms proposed by Efron and Petrosian (1999) and Shen (2008), for analyzing randomly one-sided and two-sided (i.e., doubly) truncated data. These algorithms and some recent extensions are briefly reviewed. Two real data sets are used to show how **DTDA** package works in practice.

Keywords: double truncation, nonparametric maximum likelihood, observational bias, survival analysis.

1. Introduction

Randomly truncated data appear in a variety of fields, including Astronomy, Survival Analysis, Epidemiology, or Economics. Under random truncation, only values falling in a random set which varies accross individuals are observed. For the recorded values, the truncation set is also observed. However, when the value of interest falls out of the corresponding random set, nothing is observed. This issue typically introduces a remarkable observational bias, and hence proper corrections in statistical data analysis and inference are needed.

Methods for computing the nonparametric maximum likelihood estimator (NPMLE) of a distribution function (DF) observed under random truncation have been proposed since the seminal paper by Turnbull (1976). Interestingly, the difficulties in the construction of the NPMLE heavily depend on the specific truncation pattern, i.e., on the class of allowed truncation sets. Probably, the most investigated pattern of truncation is left-truncation, for which

the truncation set is an interval unbounded from above. In epidemiological studies and industrial life-testing, left-truncation arises e.g., when performing some cross-sectional sampling, under which only individuals "in progress" at a given date (also referred as prevalent cases) are eligible. As a result, large progression times are more probably observed, and this may dramatically damage the observation of the DF of interest. For left-truncated data, the NPMLE has an explicit form and it can be computed from a simple algorithm that goes back to Lynden-Bell (1971). See Woodroofe (1985) and Stute (1993) for the statistical analysis of this estimator. The right-truncated scenario, under which the truncation sets are intervals unbounded from below, can be dealt with similarly by means of a sign change. Inference becomes more complicated, however, when other ways of truncation appear.

In many applications, the truncation sets are bounded intervals, that is, the variable of interest X^* is only observed when it falls on a (subject-specific) random interval $[U^*, V^*]$. Efron and Petrosian (1999) motivated this double-truncation issue by means of data on quasars, which are only detected when their luminosity lies between two observational limits. In epidemiology, doubly-truncated data are also encountered. For example, acquired immunodeficiency syndrome (AIDS) incubation times (from human immunodeficiency virus (HIV) infection) databases report information restricted to those individuals diagnosed prior to some specific date. This typically introduces a strong observational bias associated to right-truncation, i.e., relatively small incubation times are more probably observed. Besides, since HIV was unknown before 1982, there is some left-truncation effect too. Bilker and Wang (1996) noticed this problem and they discussed the relative impact of each type of truncation in the final sample. Moreira and de Uña-Álvarez (2010) motivated the random double-truncation phenomenon by analyzing the age at diagnosis for childhood cancer patients; as for the AIDS example, in this case the double truncation emerges from the fact that the recruited subjects are those with terminating event falling on a given observational window. Note that left (or right) truncation can be obtained from double-truncation by letting V^* (respectively U^*) be degenerated at infinity (respectively minus infinity).

A cumbersome issue with doubly-truncated data is that the NPMLE has no explicit form, and it must be computed iteratively. This complicates the analysis of its statistical properties, posing also a challenge in the design of suitable algorithms for its practical computation. See Efron and Petrosian (1999) and Shen (2008) for technical details. For the best of our knowledge, there is no package oriented to the computation of the NPMLE under double-truncation. The **DTDA** package described in this work fills this gap. **DTDA** has been implemented in R (R Development Core Team 2010) system for statistical computing. This package also allows for the analysis of one-sided (left or right) truncated data. The package **DTDA** contains three different algorithms for the approximation of the NPMLE under double-truncation (in its more general version), as well as some recent extensions, e.g., bootstrap confidence bands (Moreira and de Uña-Álvarez 2010). As it will be described below, it provides useful numerical outputs and automatic graphical displays too. Results in this document have been obtained with version 2.1-1, available from `http://CRAN.R-project.org/package=DTDA`.

The paper is organized as follows. In Section 2, a brief review of the existing algorithms to compute the NPMLE under double-truncation is given. In Section 3 the **DTDA** is described and its usage is illustrated through the analysis of two real data sets. Finally, Section 4 is devoted to conclusions and future possible extensions of the package.

2. Doubly truncated data algorithms

This section gives an introduction to the NPMLE for doubly truncated data, jointly with a review on the existing algorithms to approximate this estimator in practice. Let X^* be the lifetime of ultimate interest, with DF F, and let (U^*, V^*) be the pair of truncation times, with joint DF K. Under double truncation, only those (U^*, X^*, V^*) with $U^* \leq X^* \leq V^*$ are observed; otherwise, no information is available. For any distribution function W denote the left and right endpoints of its support by $a_W = inf\{t : W(t) > 0\}$ and $b_W = inf\{t : W(t) = 1\}$, respectively. Let $K_1(u) = K(u, \infty)$ and $K_2(v) = K(-\infty, v)$ be the marginal distribution function of U^* and V^*, respectively. When $a_{K_1} \leq a_F \leq a_{K_2}$ and $b_{K_1} \leq b_F \leq b_{K_2}$, F and K are both identifiable (see Woodroofe 1985). Let (U_i, X_i, V_i), $i = 1, \ldots, n$, denote the sample, which we assumed to be ordered with respect to the X_i's (this is relevant for the algorithm described in Section 2.2). Under the assumption of independence between X^* and (U^*, V^*), the full likelihood of the sample is given by

$$L(f, k) = \prod_{j=1}^{n} \frac{f_j k_j}{\sum_{i=1}^{n} F_i k_i},$$

where $f = (f_1, f_2, \ldots, f_n)$ and $k = (k_1, k_2, \ldots, k_n)$ are probability masses assigning probability f_i on X_i and k_i on (U_i, V_i) respectively, and F_i is defined through $F_i = \sum_{m=1}^{n} f_m J_{im}$, where

$$J_{im} = I_{[U_i \leq X_m \leq V_i]} = \begin{cases} 1 & \text{if} \quad U_i \leq X_m \leq V_i, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Here, we assume without loss of generality, that the NPMLE is a discrete distribution supported by the set of observed data (Turnbull 1976). The quantity F_i will represent the amount of mass contributed by the lifetime DF on the truncation interval $[U_i, V_i]$. As noted by Shen (2008), the full likelihood, $L(f, k)$, can be decomposed as a product of the conditional likelihood of the X_i's given the (U_i, V_i)'s, say $L_1(f)$, and the marginal likelihood of the (U_i, V_i)'s, say $L_2(f, k)$:

$$L(f, k) = \prod_{j=1}^{n} \frac{f_j}{F_j} \times \prod_{j=1}^{n} \frac{F_j k_j}{\sum_{i=1}^{n} F_i k_i} = L_1(f) \times L_2(f, k). \tag{2}$$

The first term in the decomposition in equation (2) plays a very important role in the algorithms introduced by Efron and Petrosian (1999).

2.1. First Efron-Petrosian algorithm

The conditional NPMLE of F (Efron and Petrosian 1999) is defined as the maximizer of $L_1(f)$ in equation (2):

$$\hat{f} = \underset{f}{\operatorname{argmax}}\, L_1(f). \tag{3}$$

This criterion leads to an estimator $\hat{f}$ satisfying, for all $j = 1, \ldots, n$

$$\frac{1}{\hat{f}_j} = \sum_{i=1}^{n} J_{ij} \frac{1}{\hat{F}_i}, \tag{4}$$

where $\hat{F}_i = \sum_{m=1}^{n} \hat{f}_m J_{im}$. Equation (4) was used by Efron and Petrosian (1999) to introduce the following iterative algorithm to compute $\hat{f}$ in (3).

First Efron-Petrosian algorithm

Step EP$_0$ Compute the initial vector of F_i's, say $\hat{F}_{(0)}$, from the initial probability mass, $\hat{f}_{(0)} = (1/n, \ldots, 1/n)$ which assigns uniform weights, that is, for $i = 1, \ldots, n$:

$$\hat{F}_{(0)i} = \sum_{m=1}^{n} (1/n) J_{im}.$$

Step EP$_1$ Apply equation (4) to get an improved estimator $\hat{f}_{(1)}$ and compute the $\hat{F}_{(1)}$ pertaining to $\hat{f}_{(1)}$.

Step EP$_2$ Repeat Steps EP$_0$ and EP$_1$ until a convergence criterion is reached, remembering to rescale the density estimator obtained after each application of equation (4) .

As claimed by Efron and Petrosian (1999), this algorithm often converges quite slowly. The authors suggested a different algorithm based on an adaptation of Lynden-Bell (1971) method for computing the NPMLE in the case of one-sided truncation. This method is described as the second Efron-Petrosian algorithm in the next section.

2.2. Second Efron-Petrosian algorithm

The survival curve $G = (G_1, G_2, \ldots, G_n)$ and the hazard function $h = (h_1, h_2, \ldots, h_n)$ attached to $f = (f_1, f_2, \ldots, f_n)$ are in general defined, for all $m = 1, \ldots, n$ as follows:

$$G_m = \sum_{i \geq m} f_i, \quad \text{and} \quad h_m = f_m / G_m.$$

As usual, one can always recover the survival function G and the density f from h, for all $m = 1, \ldots, n$ via the relationships:

$$G_m = \exp\left\{ \sum_{i<m} \log(1 - h_i) \right\} \quad \text{and} \quad f_m = G_m - G_{m+1},$$

with the conventions

$$G_{n+1} = 0 \quad \text{and} \quad \sum_{i<1} \log(1 - h_i) = 0.$$

For doubly-truncated data it happens that the NPMLE, namely $\hat{f}$, has hazard function $\hat{h}$ satisfying

$$\frac{1}{\hat{h}_m} = N_m + \sum_{i=1}^{n} J_{im} \hat{Q}_i, \tag{5}$$

where N_m, $m = 1, \ldots, n$, denotes the size of the risk set at time X_m if only left-truncation is considered (Woodroofe 1985), i.e.,

$$N_m = \sum_{i=1}^{n} I_{[U_i \leq X_m \leq X_i]},$$

J_{im} are the inclusion indicators defined in (1), and

$$\hat{Q}_i = \hat{G}_{V_i+}/\hat{F}_i \tag{6}$$

(Efron and Petrosian 1999) with $\hat{G}_{V_i+} = \sum_k \left\{ \hat{f}_k : X_k > V_i \right\}$. The numerator of equation (6) is the MLE probability of exceeding V_i, the upper observational limit for X_i. In the case of left truncation, $\hat{Q}_i = 0$ since $V_i = \infty$, and (5) takes the form

$$\frac{1}{\hat{h}_m} = N_m, \quad m = 1, \ldots, n \tag{7}$$

which is just Lynden-Bell (1971) estimate. In this situation, equation (5) gives the NPMLE directly, without any iteration. When dealing with two-sided truncation, equation (5) was used by Efron and Petrosian (1999) to introduce the following iterative algorithm to compute $\hat{f}$.

Second Efron-Petrosian algorithm

Step L_0 Compute the initial estimate $\hat{f}_{(0)}$ from the initial $\hat{h}_{(0)}$ defined in equation (7).

Step L_1 Apply equation (5) to get an improved estimator $\hat{h}_{(1)}$ and compute the $\hat{F}_{(1)}$ pertaining to the corresponding $\hat{f}_{(1)}$.

Step L_2 Repeat Steps L_0 and L_1 until a convergence criterion is reached.

2.3. Shen algorithm

The two different algorithms presented above are suitable if the main aim is to estimate the lifetime DF. However, in some circumstances it may be interesting to display some estimator of the truncation times distribution. This will be the case, for example, when analyzing the truncation pattern, which may be informative about different features of the process under investigation. The problem of estimating the DF of the truncation times was first discussed by Shen (2008), who provided an algorithm to jointly compute the DF of both the lifetime and the truncation random variables.

In order to introduce Shen (2008) algorithm, interchange the roles of the X_i's and the (U_i, V_i)'s in the decomposition of equation (2). Hence, the full likelihood can be also written as the product

$$L(f,k) = \prod_{j=1}^{n} \frac{k_j}{K_j} \times \prod_{j=1}^{n} \frac{K_j f_j}{\sum_{i=1}^{n} K_i f_i} = L_1(k) \times L_2(k,f)$$

where $K_i = \sum_{m=1}^{n} k_m I_{[U_m \leq X_i \leq V_m]} = \sum_{m=1}^{n} k_m J_{mi}$, for $i = 1 \ldots, n$. Here, $L_1(k)$ denotes the conditional likelihood of the (U_i, V_i)'s and $L_2(k,f)$ refers to the marginal likelihood of the

X_i's. Note that K_i stands for the probability of getting a truncation interval around X_i and hence it provides information about the relative probability of observing each of the recruited lifetimes.

Maximization of $L_1(k)$ leads to a $\hat{k} = \text{argmax}_k\, L_1(k)$ such that:

$$\frac{1}{\hat{k}_j} = \sum_{i=1}^{n} J_{ji} \frac{1}{\hat{K}_i}, \quad j = 1, \ldots, n \tag{8}$$

where $\hat{K}_i = \sum_{m=1}^{n} \hat{k}_m J_{mi}$. Shen (2008) proved that the solutions to equations (4) and (8) are not only the conditional but also the unconditional NPMLE's of F and K respectively, and that both estimators can be obtained in a simultaneous way by solving the following two equations, for $j = 1, \ldots, n$:

$$\hat{f}_j = \left[\sum_{i=1}^{n} \frac{1}{\hat{K}_i}\right]^{-1} \frac{1}{\hat{K}_j}, \tag{9}$$

$$\hat{k}_j = \left[\sum_{i=1}^{n} \frac{1}{\hat{F}_i}\right]^{-1} \frac{1}{\hat{F}_j}. \tag{10}$$

The expressions in (9) and (10) were used by Shen (2008) to introduce the following iterative algorithm to compute $\hat{f}$ and $\hat{k}$.

Shen algorithm

Step S$_0$ Compute the initial estimate $\hat{F}_{(0)}$ from $\hat{f}_{(0)} = (1/n, \ldots, 1/n)$.

Step S$_1$ Apply the formula in (10) to get the first step estimator of k, namely $\hat{k}_{(1)}$, and compute the $\hat{K}_{(1)}$ pertaining to $\hat{k}_{(1)}$.

Step S$_2$ Apply the formula in (9) to get the first step estimator of f, $\hat{f}_{(1)}$, and compute its corresponding $\hat{F}_{(1)}$.

Step S$_3$ Repeat Steps S$_1$ and S$_2$ until a convergence criterion is reached.

This algorithm and the other two discussed by Efron and Petrosian (1999) are implemented in the package **DTDA**.

As convergence criterion in all the algorithms above, we have used that the maximum pointwise error when estimating f in two consecutive steps should be below an error threshold, namely $1e{-}06$. In addition, this is an usual precision level for several packages in R.

2.4. Bootstrap approximation of the NPMLE

The asymptotic distribution of the NPMLE for doubly truncated data is not easy to determine. This is mainly because the estimator has a non-explicit form. The available results, Shen (2008), do no provide answers to important practical issues such as the computation of standard errors and the construction of confidence limits.

Moreira and de Uña-Álvarez (2010) proposed the simple bootstrap as a suitable method to approximate the finite sample distribution of the NPMLE for doubly truncated data, extending the ideas in Gross and Lai (1996) for the one-sided truncated scenario. Gross and Lai (1996) and Moreira and de Uña-Álvarez (2010) also presented a critical comparison with the obvious bootstrap method. Both procedures can be briefly explained as follows.

The simple bootstrap draws (with replacement) independent random vectors (indexed by b) (U_{ib}, V_{ib}, X_{ib}) , $i = 1, ..., n$, from the empirical distribution that puts weight $1/n$ at each of the observations (U_i, V_i, X_i), $i = 1, ..., n$. This allows for the construction of every b-th bootstrap resample, and then the procedure is repeated a large number of times B to approximate the distribution of a given statistic.

The obvious bootstrap starts by estimating the distributions of X^* and (U^*, V^*) on the basis of the observable data; this can be done by following the algorithm described in Section 2.3 and proposed by Shen (2008). Then, the resamples for X^* and (U^*, V^*), say X_{ib} and (U_{ib}, V_{ib}), $i = 1, \ldots, n$, are independently obtained with probability $P(X_{ib} = X_j) = \hat{f}_j$ and $P\left[(U_{ib}, V_{ib}) = (U_j, V_j)\right] = \hat{k}_j$, $j = 1, \ldots, n$, to draw the b-th bootstrap resample. Note that X_{ib} does not need to fall in the interval $[U_{ib}, V_{ib}]$. These samples are rejected, and hence the obvious bootstrap requires more computations than the simple bootstrap. Unlike for the simple bootstrap, the obvious bootstrap may provide new combinations of lifetimes and truncation times in the bootstrap resamples; this explains why both methods are not equivalent for randomly truncated data (see Gross and Lai 1996 and Moreira and de Uña-Álvarez 2010 for further details).

The simple bootstrap method is usually preferred to the obvious bootstrap method not only because it is substantially simpler to implement but also because it completely dispenses with the stringent assumptions (continuity of the underlying distributions, independence between the truncation times and the lifetimes, see Shen 2008) that are needed for consistent estimation of F and K in the obvious bootstrap method. The obvious bootstrap may be preferred, however, if one wants to incorporate the independence assumption in the resamples, so they can reproduce in a more precise way the sampling nature in the independent case. It should be also noticed that if the algorithms in Efron and Petrosian (1999) are to be used, then it is not possible to apply the obvious bootstrap. This is because these algorithms do not provide an empirical version of the truncation times joint distribution.

After any of simple or obvious bootstrap resampling methods are performed, the $100(1-\alpha)\%$ confidence limits for a given target can be computed in the usual way. To this end, from the large number B of values of the estimator, the upper and lower $100(\alpha/2)\%$ of values are eliminated to compute the limits. This idea is incorporated in the **DTDA** package, as it is explained below.

3. Package DTDA in practice

The **DTDA** package contains different algorithms for analyzing randomly truncated data, including one-sided and two-sided (i.e., doubly) truncated data. This section shows the usage of **DTDA** by analyzing two real data sets. The first one concerns doubly truncated data, while the second example only includes right truncation.

The new package incorporates the iterative methods introduced by Efron and Petrosian (1999) and Shen (2008) which have been presented and discussed in the previous sections. Estimation

of the lifetime DF and of the truncation times joint and marginal DFs is possible, together with the corresponding pointwise confidence limits based on bootstrap methods. Graphical displays can be automatically generated.

The **DTDA** package is composed of three functions (objects) that enable users to fit the proposed models and methods. In summary the three functions are:

`efron.petrosian()` computes the NPMLE of a lifetime DF observed under one-sided (right or left) and two-sided (double) truncation with the first algorithm of Efron and Petrosian (1999). It also provides simple bootstrap pointwise confidence limits.

`lynden()` computes the NPMLE of a lifetime DF observed under one-sided (right or left) and two-sided (double) truncation with the second algorithm of Efron and Petrosian (1999), based on an extension of Lynden-Bell's method for one-sided truncation. Simple bootstrap pointwise confidence limits are obtained.

`shen()` computes the NPMLE of a lifetime DF observed under one-sided (right or left) and two-sided (double) truncation with the algorithm proposed by Shen (2008). The NPMLE of the joint distribution of the truncation times along with its marginal distributions are also computed. Simple or obvious bootstrap pointwise confidence limits can be generated.

Table 1 shows a summary of the arguments in the three functions. It should be noted that only `X`, `U` and `V` are required arguments. The structure of the data input is as follows: each individual is represented by a single line of data. The variable `X` represents the lifetime of ultimate interest and it can not be `NA`. The variable `U` represents the left truncation times. If there is no left truncation, by putting `U = NA` the program (and the algorithm) are prepared for dealing with this type of data. The same happens with the variable `V`, which represents the right truncation times: if there is no right truncation (i.e., if the data are only left truncated), just set `V = NA`. If the values of the variables `U` and `V` are such that they do not really provide truncation from right and from left, the estimators obtained from the package should coincide with the ordinary empirical estimator which puts mass $1/n$ at each data point. This will happen if all the left truncation times are smaller than the minimum of the lifetimes, and all the right truncation times are greater than the maximum of them.

3.1. An example with doubly truncated data

In Astronomy, one of the main goals of the quasar investigations is to study luminosity evolution (Efron and Petrosian 1999, Shen 2008). The motivating example presented in the paper of Efron and Petrosian (1999) concerns a set of measurements on quasars in which there is double truncation because the quasars are observed only if their luminosity occurs within a certain finite interval, bounded at both ends, determined by limits of detection.

The original data set studied by Efron and Petrosian (1999), comprised independently collected quadruplets (z_i, m_i, a_i, b_i), $i = 1, \ldots, n$, where z_i is the redshift of the ith quasar and m_i is the apparent magnitude. Due to experimental constraints, the distribution of each luminosity in the log-scale ($y_i = t(z_i, m_i)$) is truncated to a known interval $[a_i, b_i]$, where t represents a transformation which depends on the cosmological model assumed (see Efron and Petrosian (1999) for details). Quasars with apparent magnitude above b_i were too dim to

efron.petrosian(), lynden() and shen() arguments	
X	Numeric vector with the times of ultimate interest.
U	Numeric vector with the left truncation times. If there are no truncation times from the left, put U = NA.
V	Numeric vector with the right truncation times. If there are no truncation times from the left, put V = NA.
wt	Numeric vector of non-negative initial solution, with the same length as X. Default value is set to $1/n$, being n the length of X.
error	Numeric value. Maximum pointwise error when estimating the density associated to X (f) in two consecutive steps. If this is missing, it is 1e-06.
nmaxit	Numeric value. Maximum number of iterations. If this is missing, it is set to nmaxit = 100.
boot	Logical. If TRUE (default), the simple bootstrap method is applied to lifetime distribution estimation. Pointwise confidence bands are provided*.
B	Numeric value. Number of bootstrap resamples . The default NA is equivalent to B = 500.
alpha	Numeric value. (1− alpha) is the nominal coverage for the pointwise confidence intervals.
display.F	Logical. Default is FALSE. If TRUE, the estimated cumulative distribution function associated to X, (F) is plotted*.
display.S	Logical. Default is FALSE. If TRUE, the estimated survival function associated to X, (S) is plotted*.
shen() arguments	
boot.type	A character string giving the bootstrap type to be used. This must be one of "simple" or "obvious", with default "simple".
display.FS	Logical. Default is FALSE. If TRUE, the estimated cumulative distribution function and the estimated survival function associated to X, (F) and (S) respectively, are plotted.
display.UV	Logical. Default is FALSE. If TRUE, the marginal distributions of U (fU) and V (fV), are plotted.
plot.joint	Logical. Default is FALSE. If TRUE, the joint distribution of the truncation times is plotted.
plot.type	A character string giving the plot type to be used to represent the joint distribution of the truncation times. This must be one of "image" or "persp", with default NULL.

Table 1: Summary of the arguments of the efron.petrosian(), lynden() and shen() functions. The arguments marked with * in the first part of the table are included in shen() with further options.

yield dependent redshifts, and hence they were excluded from the study. The lower limit a_i was used to avoid confusion with non quasar stellar objects. The $n = 210$ quadruplets investigated by Efron and Petrosian (1999) were kindly provided by the authors. At the beginning of Section 2 we referred to some identifiability conditions for the estimation of the population DFs. For this data set, the extreme ordered statistics of the adjusted log luminosities (-2.34

and 2.08) are relatively close to the minimum lower bound (-2.40) and the maximum upper bound (2.58) respectively, suggesting that $a_{K_1} \leq a_F$ or $b_F \leq b_{K_2}$ could be violated. Note that, in general, the obtained estimator for F can only be regarded as an estimator of F conditionally on $X^* \in [a_{K_1}, b_{K_2}]$.

In this section the usage of the three functions `efron.petrosian()`, `lynden()` and `shen()` is illustrated by analyzing the quasars data set. The practical application is mainly focused on the function `shen()` because, unlike the other two functions, it provides not only the estimators for the 'lifetime' DF but also the curves corresponding to the truncation times. Besides, the computation of confidence limits throughout the two bootstrap resampling methods discussed previously in Section 2.4 is also provided. Numerical outputs for the function `efron.petrosian()` will not be given, since they are just a subset of the results displayed here. However, since the algorithm behind the function `lynden()` is somehow different, some of the results obtained with this function are also shown.

The data are incorporated in the matrix object `Quasars`; the second and the third columns correspond to the left and right truncation times respectively, while the first column is reserved for the variable of interest (in this example, log of quasar luminosity). Using `shen()` the estimated cumulative distribution can be analyzed, jointly with the estimated survival and other values of interest provided by the next output (edited to show only the first and last lines of output):

```
> fit1 <- shen(Quasars[, 1], Quasars[, 2], Quasars[, 3], display.FS = TRUE,
+     display.UV = TRUE, nmaxit = 10000)

n.iterations 43
S0 9.997212e-07
events 210
B 500
alpha 0.05
Boot simple
       time n.event density cumulative.df survival
 -2.3449016       1 0.48893       0.48893  1.00000
 -2.1438677       1 0.09826       0.58719  0.51107
 -1.8699029       1 0.04961       0.63681  0.41281
 -1.8583955       1 0.04961       0.68642  0.36319
 -1.7929619       1 0.03729       0.72371  0.31358
 -1.4058456       1 0.01574       0.73945  0.27629
 -1.4052073       1 0.01574       0.75519  0.26055

...

  1.3904633       1 0.00014       0.99903  0.00111
  1.4346404       1 0.00014       0.99917  0.00097
  1.5695026       1 0.00015       0.99931  0.00083
  1.5888410       1 0.00015       0.99946  0.00069
  1.6662626       1 0.00015       0.99961  0.00054
  1.7041021       1 0.00016       0.99977  0.00039
  2.0846553       1 0.00023       1.00000  0.00023
```

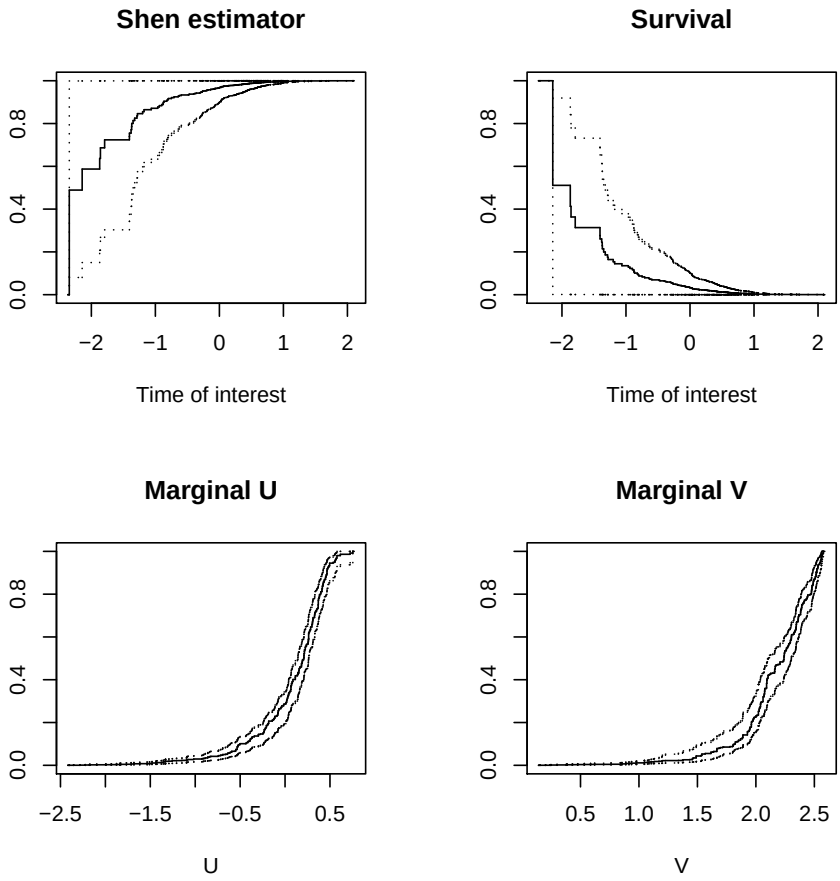

Figure 1: Cumulative DF and Survival function of quasars luminosities (top) and marginal DF of each of the truncation variables (bottom), applying `shen()`, with `"simple"` bootstrap for 95% pointwise confidence bands.

Note that the output provides information about the observed adjusted log luminosities, the number of events (which will be 1 in the case of no ties), the estimated density at each point, and the cumulative curves (cumulative DF and survival function). There is some preliminary information about the confidence level used for the computation of the bootstrap confidence limits as well as the number of iterations when computing the NPMLE and the maximum pointwise error when estimating f in two consecutive steps. The default stop criterion here is `1e-06`.

Automatic graphical displays are obtained when changing the default `FALSE` to `TRUE` for the arguments `display.FS` (cumulative DF and survival function) and `display.UV` (marginal DFs of the truncation variables). These plots are reported in Figure 1, which includes the 95% confidence bands based on the simple bootstrap. These bands can be skipped by setting `boot = FALSE`; alternative bands based on the obvious bootstrap can be displayed by setting `boot.type = "obvious"`. Similarly, a graphical plot of the bivariate DF of the truncation variables is obtained by setting `display.joint = TRUE`. This output is reported in

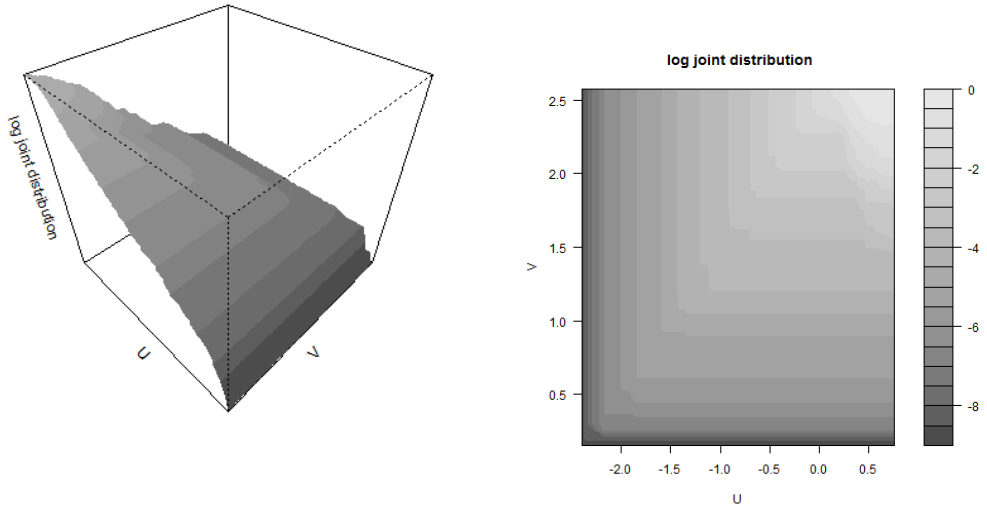

Figure 2: Bivariate distribution, in log scale of the truncation variables, for the quasar data, using option `"persp"` (left panel) and option option `"image"` (right panel).

Figure 2 when choosing two different types of plotting: `plot.type = "persp"` (left panel) and `plot.type = "image"` (right panel), considering in both cases the log joint distribution.

As it was already mentioned, adjusted log luminosity databases report information restricted to those quasars with apparent magnitude within a limit of detection interval. This introduces a strong observational bias since relatively small and large luminosities are less probably observed. This feature can be observed in Figure 3 (left panel), which shows that adjusted log luminosities below zero are observed with a particularly small probability. This display was constructed from the output `biasf` of `shen()` function, which contains the estimated quantities $P(U^* < x < V^*)$ representing the probability that the detection interval contains a lifetime (i.e., adjusted log-luminosity) of magnitude x. In the untruncated case, the curve in Figure 3, left, should be flat; under truncation, however, different shapes representing the observational bias will be obtained.

In order to compare the confidence bands obtained when using the two different bootstrap methods `"simple"` and `"obvious"`, Figure 3 (right panel) shows the estimated log survival function for the quasar data together with the 95% pointwise confidence bands. The pointwise confidence bands using `"simple"` bootstrap are shown in green, whereas the confidence bands with `"obvious"` bootstrap are plotted in red. It can be seen that these methods produce in general different results; this is not surprising, since they are not equivalent as discussed in Section 2.

The second algorithm proposed by Efron and Petrosian (1999), as mentioned at the beginning of this Section, may report results somehow different to those corresponding to the first algorithm. Besides, both algorithms, although oriented to maximize the same likelihood, follow different steps, and hence it is not surprising that the solutions may be slightly different in particular cases. As it can be observed in the next output, the number of iterations needed

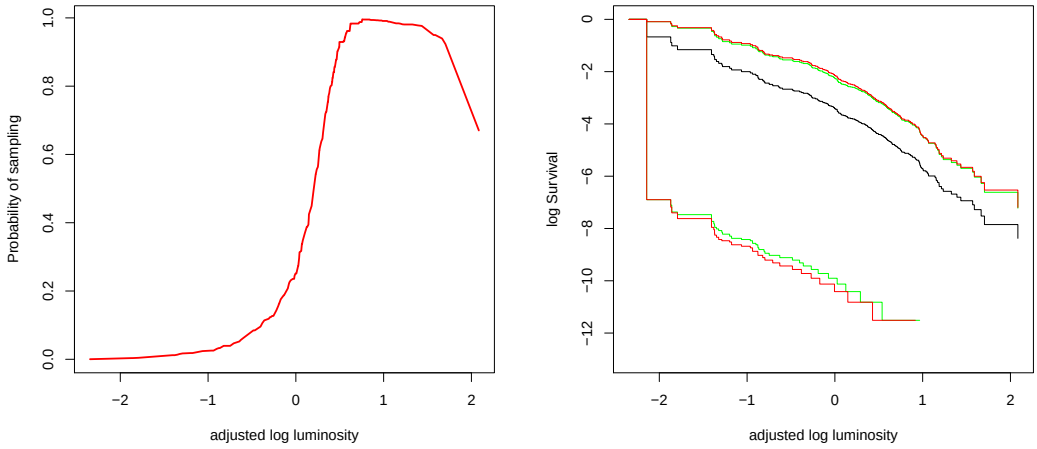

Figure 3: Bias function for the quasar data (left panel). Estimated log survival for the quasar data, using `shen()` function and 95% pointwise confidence bands for simple (red) and obvious (green) bootstrap methods (right panel).

to meet the stop criterion is quite smaller here compared with the previous output obtained in `fit1` using `shen()` function (7 against 43). This feature is in agreement with the discussion in Efron and Petrosian (1999) (see pp. 828–829). Note that this numerical output coincides with that of the function `shen()` (which uses the first algorithm in Efron and Petrosian (1999), for the computation of the lifetime density and DF). This does not need to be the case in general (see our second example in Section 3.2), although the differences between the solutions provided by both functions should not be large. In general, the function `lynden()` could be recommended to save computational time.

```
> fit2 <- lynden(Quasars[, 1], Quasars[, 2], Quasars[, 3], boot = FALSE,
+     display.F = FALSE, display.S = FALSE)

n.iterations 7
S0 4.525812e-07
events 210
       time n.event density cumulative.df survival  hazard
 -2.3449016       1 0.48893       0.48893  1.00000 0.48893
 -2.1438677       1 0.09826       0.58720  0.51107 0.19227
 -1.8699029       1 0.04961       0.63681  0.41280 0.12018
 -1.8583955       1 0.04961       0.68642  0.36319 0.13660
 -1.7929619       1 0.03729       0.72372  0.31358 0.11893
 -1.4058456       1 0.01574       0.73945  0.27628 0.05697
 -1.4052073       1 0.01574       0.75519  0.26055 0.06041

...
```

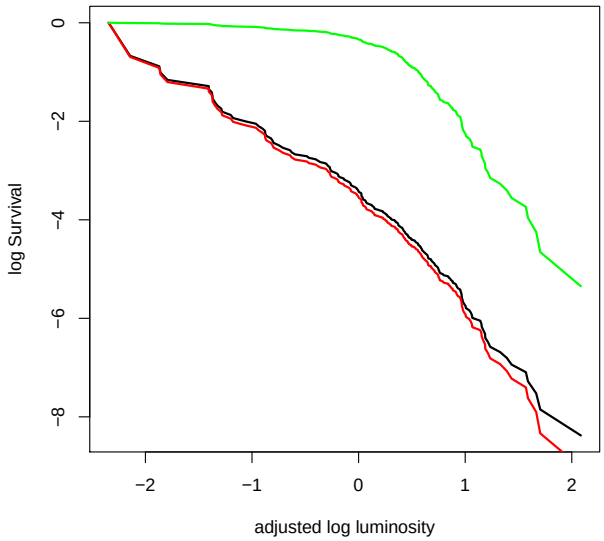

Figure 4: Estimated log survival as a function of the adjusted log luminosity evolution for the quasar data, using the NPMLE of Efron and Petrosian (black line), the Lynden-Bell estimate ignoring upper truncation (red curve), and ignoring both left and right truncation (green curve).

```
1.3904633       1 0.00014       0.99903  0.00111 0.12527
1.4346404       1 0.00014       0.99917  0.00097 0.14335
1.5695026       1 0.00015       0.99931  0.00083 0.17655
1.5888410       1 0.00015       0.99946  0.00069 0.21441
1.6662626       1 0.00015       0.99961  0.00054 0.27745
1.7041021       1 0.00016       0.99977  0.00039 0.39891
2.0846553       1 0.00023       1.00000  0.00023 1.00000
```

An issue that has been overseen in many applications is that of the important bias associated to random truncation. For the quasar data, ignoring the left truncation may be very important, as suggested by our Figure 3. The plot in Figure 4 was depicted by using the `lynden()` function applied to several situations. The first one is that considering the double truncation, as performed above (in `fit2`). The second one (saved as `fit3` below) ignores right truncation; this can be easily done by setting `V = NA`. Finally, `fit4` below contains the output of the function `lynden()` when removing both (right and left) truncation times. For doing this, `V = NA` must be kept and at the same time, ignorable lower truncation bounds must be introduced (since `U = NA` does not work in the presence of `V = NA`). This latter output just provides the ordinary survival function which attaches mass $1/n$ to each of the adjusted log luminosities. Figure 4 reveals the strong impact of left truncation in the estimation of the DF of the quasar luminosities. This is in agreement with the observational bias depicted in Figure 3, left.

```
> fit3 <- lynden(Quasars[, 1], Quasars[, 2], V = NA, boot = FALSE,
+     display.F = FALSE, display.S = FALSE)
> fit4 <- lynden(Quasars[, 1], U = min(Quasars[, 1]) - 1, V = NA,
+     boot = FALSE, display.F = FALSE, display.S = FALSE)
```

3.2. An example with right-truncated data

Induction Times for AIDS data from Lagakos, Barraj, and de Gruttola (1988) are used to illustrate a situation in which one-sided (rather than two-sided) truncation appears. This data set is available from the book by Klein and Moeshberger (2003, Table 1.10, pp. 20). The data include information on the infection and induction times for 258 adults and 37 children who were infected with HIV virus and developed AIDS by 1996-06-30 The data consist on the time in years, measured from 1978-04-01, when adults were infected by the virus from a contaminated blood transfusion, and the waiting time to development of AIDS, measured from the date of infection. In this sampling scheme, only individuals who had developed AIDS before the end of the study period were included and so the induction times suffer from right truncation.

Let X be the induction time, that is, the time from HIV infection to the diagnosis of AIDS; and denote by T the time from HIV infection to the end of the study, which plays the role of right truncation time. Only those individuals (X, T) with $X \leq T$ are observed. In this example the sole information included is the infection and the induction times for the 258 adults. These variables X and T are reported in the second and the third column, respectively, of the matrix `AIDSdata` called below.

In order to perform the data analysis, the function `shen()` is used, setting `U = NA` to inform about the absence of left-truncation. As it can be seen in the next numerical output, the algorithm converged after 19 iterations; it can be also noticed that (unlike for the quasar data example) there is a clear presence of ties in this data set.

```
> fit5 <- shen(AIDS[, 2], U = NA, AIDS[, 3], boot = TRUE, display.FS = TRUE,
+     display.UV = TRUE, nmaxit = 1e+05)

case U=NA
n.iterations 19
S0 7.677726e-07
events 258
B 500
alpha 0.05
Boot simple
 time n.event density cumulative.df survival
 0.25       6 0.00341       0.00341  1.00000
 0.50       2 0.00114       0.00455  0.99659
 0.75      13 0.00739       0.01193  0.99545
 1.00      15 0.00895       0.02088  0.98807
 1.25      16 0.00983       0.03071  0.97912
 1.50      23 0.01643       0.04714  0.96929
 1.75      13 0.01021       0.05735  0.95286
```

```
 2.00      14 0.01181       0.06916  0.94265
 2.25      20 0.02065       0.08981  0.93084
 2.50      15 0.01727       0.10708  0.91019
 2.75      14 0.01806       0.12514  0.89292
 3.00      21 0.03327       0.15841  0.87486
 3.25      13 0.02367       0.18208  0.84159
 3.50       8 0.01755       0.19963  0.81792
 3.75       5 0.01406       0.21369  0.80037
 4.00      11 0.03731       0.25099  0.78631
 4.25       9 0.03530       0.28629  0.74901
 4.50       6 0.02771       0.31400  0.71371
 4.75       5 0.02962       0.34362  0.68600
 5.00       8 0.05849       0.40211  0.65638
 5.25       9 0.08617       0.48827  0.59789
 5.50       4 0.05580       0.54407  0.51173
 5.75       2 0.04030       0.58438  0.45593
 6.00       1 0.02164       0.60602  0.41562
 6.25       1 0.03565       0.64167  0.39398
 6.50       2 0.09167       0.73333  0.35833
 6.75       1 0.06667       0.80000  0.26667
 7.25       1 0.20000       1.00000  0.20000
```

The automatic graphical displays of the command line above is given in Figure 5. The confidence bands (based on the simple bootstrap) are wider for large incubation times, in accordance to the under-information at these points, related to the right-truncation phenomenon. Since this data set is one-sided truncated, the best algorithm here is the second one proposed in Efron and Petrosian (1999), which is just Lynden-Bell (1971) method. As discussed in Section 2, this algorithm converges after one iteration under one-sided truncation (indeed, the estimator has an explicit form in this case). The following output displays the numerical results achieved by the function `lynden()`. Unlike for the quasar data example, notice that the figures are not exactly the same as those reported by the function `shen()`.

```
> fit6 <- lynden(AIDS[, 2], U = NA, AIDS[, 3], boot = FALSE)

case U=NA
n.iterations 1
S0 3.079134e-17
events 258
 time n.event density cumulative.df survival  hazard
 0.25       6 0.00761       0.00761  1.00000 0.00761
 0.50       2 0.00233       0.00994  0.99239 0.00235
 0.75      13 0.00881       0.01875  0.99006 0.00889
 1.00      15 0.01021       0.02896  0.98125 0.01041
 1.25      16 0.01105       0.04001  0.97104 0.01138
 1.50      23 0.01683       0.05685  0.95999 0.01754
 1.75      13 0.01116       0.06801  0.94315 0.01184
 2.00      14 0.01275       0.08076  0.93199 0.01368
```

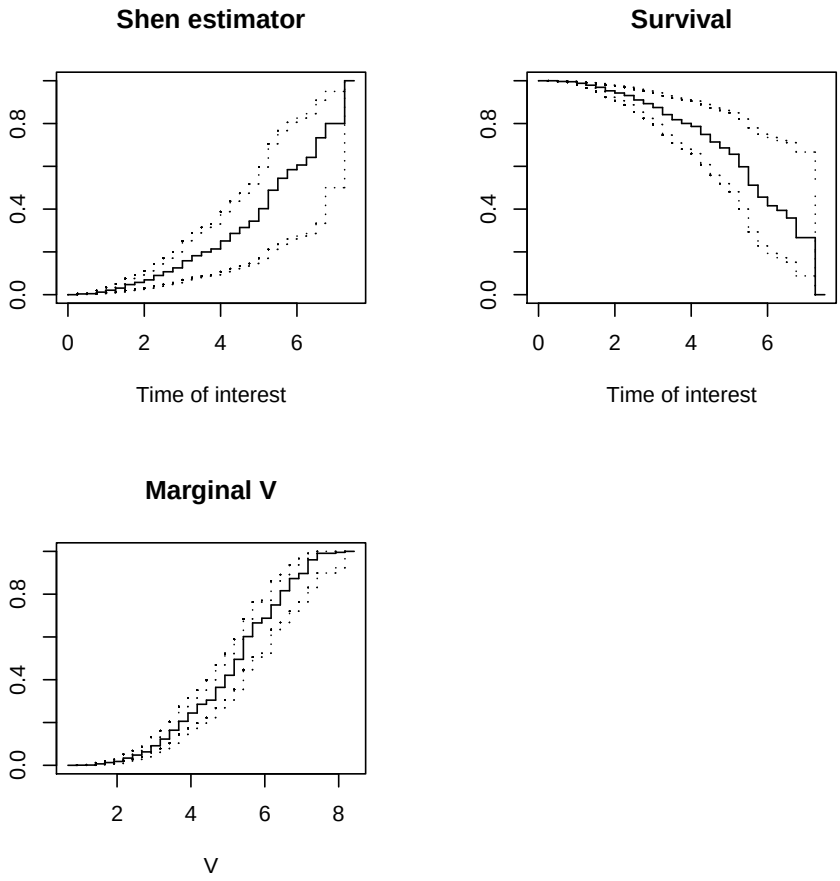

Figure 5: NPMLE obtained by `shen()` function for the cumulative DF and survival function of the AIDS induction times (top) and marginal DF of the right-truncation variable (bottom), together with the 95% pointwise confidence bands based on simple bootstrap method.

```
2.25      20 0.02101      0.10176  0.91924 0.02285
2.50      15 0.01792      0.11968  0.89824 0.01995
2.75      14 0.01869      0.13837  0.88032 0.02123
3.00      21 0.03251      0.17088  0.86163 0.03774
3.25      13 0.02385      0.19473  0.82912 0.02877
3.50       8 0.01800      0.21273  0.80527 0.02235
3.75       5 0.01457      0.22729  0.78727 0.01850
4.00      11 0.03669      0.26399  0.77271 0.04749
4.25       9 0.03489      0.29888  0.73601 0.04741
4.50       6 0.02778      0.32667  0.70112 0.03963
4.75       5 0.02968      0.35634  0.67333 0.04408
5.00       8 0.05634      0.41269  0.64366 0.08754
5.25       9 0.08051      0.49320  0.58731 0.13709
5.50       4 0.05400      0.54720  0.50680 0.10655
```

```
5.75      2 0.03978      0.58698  0.45280 0.08786
6.00      1 0.02174      0.60872  0.41302 0.05264
6.25      1 0.03581      0.64453  0.39128 0.09151
6.50      2 0.08880      0.73333  0.35547 0.24982
6.75      1 0.06667      0.80000  0.26667 0.25000
7.25      1 0.20000      1.00000  0.20000 1.00000
```

4. Conclusions

This paper discusses the implementation in R of several algorithms for computing the NPMLE of the cumulative DF in the presence of random truncation. The **DTDA** package implements in a friendly way the methods proposed by Efron and Petrosian (1999) and Shen (2008). For the best of our knowledge, this is the first contribution of this type to deal with the non-standard (and sometimes ignored) issue of random truncation. The package **DTDA** provides not only the numerical outputs of main interest but also automatic graphical displays of several curves, such as the cumulative DF and the survival function of the lifetime as well as the marginal and joint DFs of the truncation times. Besides, two different bootstrap methods are implemented for the computation of confidence limits.

The function `lynden()` may give results somehow different to those provided by the functions `efron.petrosian()` or `shen()`. The algorithm behind `lynden()`, although oriented to maximize the same likelihood as `shen()` and `efron.petrosian()`, follows different steps, and hence it is not surprising that the solutions may be slightly different in particular cases. We should also point out the slow speed of convergence of the algorithms EP_0–EP_2 and S_0–S_3 when compared to L_0–L_2 (Efron and Petrosian 1999, p. 828); see also our application to quasar data above. Although this seems to be typically the case, we have found special situations in which `shen()` or `efron.petrosian()` may converge in fewer steps than `lynden()`. So a definite conclusion about this point can not be given.

An interesting extension of the package would be the implementation of smooth estimates for e.g., density and hazard rate functions. This could be done by computing kernel estimators, which are obtained from the NPMLE by convolution with a kernel function, providing a smooth estimator. See for example Wand and Jones (1995) for access to related literature. Finally, adaptation of the implemented methods to the context of regression with truncated responses could be provided, by using the empirical estimators computed by **DTDA** to weight the residuals, in the spirit of Stute (1993) for the censored case, see also Sánchez-Sellero, González-Manteiga, and Van Keilegom (2005) for left-truncated, right censored responses. However, this is a field of research which remains unexplored for doubly truncated data, and new methods should be carefully worked out before this extension is possible.

Ackowledgments

We thank three anonymous referees and the Associate Editor for their careful reading of the paper and suggestions which have greatly improved the manuscript. Work supported from Spanish Ministerio de Ciencia e Innovación by the research Grants MTM2008-03129 and MTM2008-03010, by the Xunta de Galicia Grant PGIDIT07PXIB300191PR and SFRH/BD/60262/2009 Grant of Portuguese Fundação Ciência e Tecnologia. The authors are also grateful to the

Xunta de Galicia for financial support under the INBIOMED project, (DXPCTSUG, Ref. 2009/063). We also thank Professors Bradley Efron and Vahe Petrosian who kindly provided the quasar data.

References

Bilker WB, Wang MC (1996). "A Semiparametric Extension of the Mann-Whitney Test for Randomly Truncated Data." *Biometrics*, **52**, 10–20.

Efron B, Petrosian V (1999). "Nonparametric Methods for Doubly Truncated Data." *Journal of the American Statistical Association*, **94**, 824–834.

Gross ST, Lai TL (1996). "Bootstrap Methods for Truncated and Censored Data." *Statistica Sinica*, **6**, 509–530.

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

Lagakos SW, Barraj LM, de Gruttola V (1988). "Nonparametric Analysis of Truncated Survival Data, with Applications to AIDS." *Biometrika*, **75**, 515–523.

Lynden-Bell D (1971). "A Method for Allowing for Known Observational Selection in Small Samples Applied to 3CR Quasars." *Monthly Notices of the Royal Astronomical Society*, **155**, 95–118.

Moreira C, de Uña-Álvarez J (2010). "Bootstrappping the NPMLE for Doubly Truncated Data." *Journal of Nonparametric Statistics*, **22**, 567–583.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Shen P (2008). "Nonparametric Analysis of Doubly Truncated Data." *Annals of the Institute of Statistical Mathematics*, **62**(5), 835–853.

Stute W (1993). "Almost Sure Representations of the Product-Limit Estimator for Truncated Data." *The Annals of Statistics*, **21**, 146–156.

Sánchez-Sellero C, González-Manteiga W, Van Keilegom I (2005). "Uniform Representation of Product-Limit Integrals with Applications." *Scandinavian Journal of Statistics*, **32**, 563–581.

Turnbull BW (1976). "The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data." *Journal of the Royal Statistical Society B*, **38**, 290–295.

Wand MP, Jones MC (1995). *Kernel Smoothing*. Chapman and Hall, London.

Woodroofe M (1985). "Estimating a Distribution Function with Truncated Data." *The Annals of Statistics*, **13**, 163–177.

Affiliation:

Carla Moreira, Jacobo de Uña-Álvarez, Rosa M. Crujeiras
Department of Statistics and OR
Faculty of Economics-University of Vigo
36310 Vigo, Spain
E-mail: carlamgmm@gmail.com, jacobo@uvigo.es, rosa.crujeiras@usc.es
URL: http://webs.uvigo.es/depc05/,
http://webs.uvigo.es/jacobo/,
http://eio.usc.es/pub/Crujeiras

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 37, Issue 7 *Submitted:* 2009-12-11
November 2010 *Accepted:* 2010-07-07