The R Package smicd: Statistical Methods for Interval-Censored Data

by Paul Walter

Abstract  The package allows the use of two new statistical methods for the analysis of interval-censored data: 1) direct estimation/prediction of statistical indicators, and 2) linear (mixed) regression analysis. Direct estimation of statistical indicators, for instance poverty and inequality indicators, is facilitated by a non-parametric kernel density algorithm. The algorithm is able to account for weights in the estimation of statistical indicators. The standard errors of the statistical indicators are estimated with a non-parametric bootstrap. Furthermore, the package offers statistical methods for the estimation of linear and linear mixed regression models with an interval-censored dependent variable, particularly random slope and random intercept models. Parameter estimates are obtained through a stochastic expectation-maximization algorithm. Standard errors are estimated using a non-parametric bootstrap in the linear regression model and by a parametric bootstrap in the linear mixed regression model. To handle departures from the model assumptions, fixed (logarithmic) and data-driven (Box-Cox) transformations are incorporated into the algorithm.

Introduction

Interval-censored or grouped data occurs when only the lower $A_{k-1}$ and upper $A_k$ interval bounds $(A_{k-1}, A_k)$ of a variable are observed and its true value remains unknown. Instead of measuring the variable of interest on a continuous scale, for instance income data, the scale is divided into $n_k$ intervals. The variable $k (1 \leq k \leq n_k)$ indicates in which of the $n_k$ intervals an observation falls into. This leads to a loss of information since the shape of the distribution within the intervals remains unknown. In the field of survey statistics, asking for interval-censored data is often done in order to avoid item non-response and thus increase data quality. Item non-response is avoided because interval-censored data offers a higher level of data privacy protection (Hagenaars and Vos, 1988; Moore and Welniak, 2000). Among others, popular surveys and censuses that collect interval-weights in the estimation of statistical indicators. The standard errors of the statistical indicators are estimated with a non-parametric bootstrap. Furthermore, the package offers statistical methods for the estimation of linear and linear mixed regression models with an interval-censored dependent variable, particularly random slope and random intercept models. Parameter estimates are obtained through a stochastic expectation-maximization algorithm. Standard errors are estimated using a non-parametric bootstrap in the linear regression model and by a parametric bootstrap in the linear mixed regression model. To handle departures from the model assumptions, fixed (logarithmic) and data-driven (Box-Cox) transformations are incorporated into the algorithm.

For the estimation of statistical indicators from interval-censored data different approaches are described in the literature. These approaches can be broadly categorized into four groups: Estimation on the midpoints (Fahrmeir et al., 2016), linear interpolation of the distribution function, non-parametric modeling via splines (Berger and Escobar, 2016) and fitting a parametric distribution function to the censored data (Dagum, 2008; McDonald, 1984; Bandourian et al., 2002). Some of these methods are implemented in R packages available on the Comprehensive R Archive Network (CRAN). The method of linear interpolation is implemented for the estimation of quantiles in the R package actuar (Goulet et al., 2020; Dutang et al., 2008). The package also enables the estimation of the mean on the interval midpoints. Fitting a parametric distribution to interval-censored data can be done by using the R package fitdistrplus (Delignette-Muller et al., 2020; Delignette-Muller and Dutang, 2015).

In survey statistics, interval-censored data is often collected for income or wealth variables. Thus, the performance of the above-mentioned methods is commonly evaluated by simulation studies that rely on data that follows some kind of income distribution. The German statistical office (DESTATIS) uses the method of linear interpolation for the estimation of statistical indicators from interval-censored income data collected by the German Microcensus. This approach gives the same results as assuming a uniform distribution within the income intervals. Estimation results are reasonably accurate if the estimated indicators do not depend on the whole shape of the distribution, e.g., the median (Lenau and Münnich, 2016). Fitting a parametric distribution to the data enables the estimation of indicators that rely on the whole shape of the distribution. This method works well when the data is censored to
only a few equidistant intervals (Lenau and Münnich, 2016). Non-parametric modeling via splines shows especially good results for a high number of intervals in ascending order (Lenau and Münnich, 2016). However, according to Lenau and Münnich (2016) all of the above-mentioned methods show large biases and variances when the estimation is based on a small number of intervals. Therefore, a novel kernel density estimation (KDE) algorithm is implemented in the smicd package that overcomes the drawbacks of the previously mentioned methods (Walter, 2019, 2020). The algorithm bases the estimation of statistical indicators on pseudo samples that are drawn from a fitted non-parametric distribution. The method automatically adapts to the shape of the true unknown distribution and provides reliable estimates for different interval-censoring scenarios. It can be applied via the function kdeAlgo().

Similarly to the direct estimation of statistical indicators from interval-censored data, there exists a variety of ad-hoc approaches and explicitly formulated mathematical methods for the estimation of linear regression models with an interval-censored dependent variable. The following methods and approaches are used for handling interval-censored dependent variables within linear regression models: Ordinary least squares (OLS) regression on the midpoints (Thompson and Nelson, 2003), ordered logit- or probit-regression (McCullagh, 1980), and regression methodology formulated for left-, right-, and interval-censored data (Tobin, 1958; Rosett and Nelson, 1975; Stewart, 1983). All of these methods are implemented in different R packages available on CRAN. OLS regression on the midpoints is applicable by using the lm() function from the stats package (R Core Team, 2020), ordered logit regression is implemented in the MASS package (Ripley, 2019; Venables and Ripley, 2002), and interval regression is implemented in the survival (Therneau, 2020; Therneau and Grambsch, 2000) package.

While OLS regression on the midpoints of the intervals is easily applied, it comes with the disadvantage of giving biased estimation results (Cameron, 1987). This approach disregards the uncertainty stemming from the unknown true distribution of the data within the intervals and therefore leads to biased parameter estimates. Its performance relies on the number of intervals and estimation results are only comparable to more advanced methods when the number of intervals is very large (Fryer and Pethybridge, 1972). Conceptualizing the model as an ordered logit or probit regression is feasible by treating the dependent variable as an ordered factor variable (McCullagh, 1980). However, this approach also neglects the unknown distribution of the data within the intervals. Furthermore, the predicted values are not on a continuous scale but are in terms of probability of belonging to a certain group. To overcome these disadvantages and obtain unbiased estimation results Stewart (1983) introduces regression methodology for models with an interval-censored dependent variable. Walter (2019) further develop his approach and introduce a novel stochastic expectation-maximization (SEM) algorithm for the estimation of linear regression models with an interval-censored dependent variable that is implemented in the smicd package. The model parameters are unbiasedly estimated as long as the model assumptions are fulfilled. The function semLm() provides the SEM algorithm and enables the use of fixed (logarithmic) and data-driven (Box-Cox) transformations (Box and Cox, 1964). The Box-Cox transformation automatically adapts to the shape of the data and transforms the dependent variable in order to meet the model assumption.

In order to analyze longitudinal or clustered data (e.g., students within schools) linear mixed regression models are applicable. These kinds, of models control for the correlated structure of the data by including random effects in addition to the usual fixed effects. In order to deal with an interval-censored dependent variable in linear mixed regression models there are several approaches described in the literature. Linear mixed regression models, just like linear regression models, can be estimated on the interval midpoints of the censored-dependent variable. Furthermore, conceptualizing the model as an ordered logit or probit regression model is feasible (Agresti, 2010). These approaches inherit the same advantages and disadvantages as previously discussed. Linear mixed regression on the midpoints can be applied by the ImeI (Bates et al., 2020b, 2015) or nlme (Pinheiro et al., 2020) package and the ordered logit regression is implemented in the ordinal package (Christensen, 2019). To my knowledge, there are no R packages for the estimation of linear mixed regression models with an interval-censored dependent variable. Therefore, the package smicd contains the SEM algorithm proposed by Walter (2019) for the estimation of linear mixed regression models with an interval-censored dependent variable. If the model assumptions are fulfilled, the method gives unbiased estimation results. The function semLm() enables the estimation of the regression parameters and it also allows for the usage of the logarithmic and Box-Cox transformation in order to fulfill the model assumptions (Curka et al., 2006).

The paper is structured into two main sections. Section 2.2 deals with the direct estimation of statistical indicators from interval-censored data whereas Section 2.3 introduces linear and linear mixed regression models with an interval-censored dependent variable. Both sections have been divided into three subsections: first the statistical methodology is introduced, then the core functions of the smicd package are presented, and finally, illustrative examples with two different data sets are provided. In Section 2.4 the main results are summarized and an outlook is given.
Direct estimation of statistical indicators

In the following three subsections, the methodology for the direct estimation of statistical indicators from interval-censored data is introduced, the core functionality of the function kdeAlgo() is presented and statistical indicators are estimated using the synthetic EU-SILC (European Union Statistics on Income and Living Conditions) data set from Austria.

Methodology: Direct estimation of statistical indicators

In order to estimate statistical indicators from interval-censored data the proposed algorithm generates metric pseudo samples of an interval-censored variable. These pseudo samples can be used to estimate any statistical indicator. They are drawn from a non-parametrically estimated kernel density. Kernel density estimation was first introduced by (Rosenblatt, 1956) and (Parzen, 1962). By its application the density \( f(x) \) of a continuous independently and identically distributed random variable is estimated without assuming any distributional shape of the data. The estimator is defined as

\[
\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{x-x_i}{h} \right), \quad i = 1, \ldots, n,
\]

where \( K(\cdot) \) is a kernel function, \( h > 0 \) the bandwidth and \( x = \{x_1, x_2, \ldots, x_n\} \) denotes a sample of size \( n \). The performance of the estimator is determined by the optimal choice of \( h \). The selection of an optimal \( h \) is widely discussed in the literature, see Jones et al. (1996); Loader (1999); Zambom and Dias (2012). When working with interval-censored data, a standard KDE cannot be applied since \( x \) is not observed on a continuous scale. Nevertheless, its unobserved true distribution is of continuous form. As an ad hoc solution the density \( \hat{f}_h(x) \) can be estimated based on the interval midpoints. The resulting density estimate will be spiky unless the bandwidth is sufficiently large. A large bandwidth, however, leads to a loss of information (Wang and Wertelecki, 2013). Therefore, Walter (2019) proposes an iterative KDE algorithm for density estimation from interval-censored data. The approach is based on Groß et al. (2017) who introduce a similar KDE algorithm in a two-dimensional setting with an equidistant interval width. Walter (2019) shows that the algorithm can be adjusted to one-dimensional data with an arbitrary class width. For the estimation of linear and non-linear statistical indicators the unknown distribution of \( x \) has to be reconstructed by using the observed interval \( k = \{k_1, k_2, \ldots, k_n\} \) that an observation falls into. From Bayes’ theorem (Bayes, 1763) it follows that the conditional distribution of \( (x|k) \):

\[
\pi(x|k) \propto \pi(k|x) \pi(x)
\]

with \( \pi(k|x) \) is defined by a product of a Dirac distribution \( \pi(k|x) = \prod_{i=1}^{n} \pi(k_i|x_i) \) with

\[
\pi(k_i|x_i) = \begin{cases} 1 & \text{if } A_{k_{i-1}} \leq x_i \leq A_{k_i}, \\ 0 & \text{else}, \end{cases}
\]

for \( i = 1, \ldots, n \). Since \( \pi(x) \) is unknown it is replaced by a kernel density estimate \( \hat{f}_h(x) \).

Estimation and computational details

To fit the model, pseudosamples of \( x_i \) are drawn from the conditional distribution

\[
\pi(x_i|k) \propto I(A_{k_{i-1}} \leq x_i \leq A_{k_i}) f(x_i),
\]

where \( I(\cdot) \) denotes the indicator function. The conditional distribution of \( \pi(x_i|k_i) \) is given by the product of a uniform distribution and density \( f(x_i) \). As the density is unknown it is replaced by an estimate \( \hat{f}_h(x) \), which is obtained by the KDE. In particular, \( x_i \) is repeatedly drawn from the given interval \( (A_{k_{i-1}}, A_{k_i}) \) by using the current density estimate \( \hat{f}_h(x) \) as a sampling weight. The explicit steps of the iterative algorithm as given in Walter (2019) are stated below:

1. Use the midpoints of the intervals as pseudo \( \tilde{x}_i \) for the unknown \( x_i \). Estimate a pilot estimate of \( \hat{f}_h(x) \), by applying KDE. Note: Choose a sufficiently large bandwidth \( h \) in order to avoid rounding spikes.
2. Evaluate \( \hat{f}_h(x) \) on an equal-spaced grid \( G = \{g_1, \ldots, g_j\} \) with grid points \( g_1, \ldots, g_j \). The width of the grid is denoted by \( \delta_x \). It is given by

\[
\delta_x = \frac{|A_0 - A_n|}{j - 1}.
\]
and the grid is defined as:
\[ G = \{ g_1 = A_0, g_2 = A_0 + \delta g, g_3 = A_0 + 2\delta g, \ldots, g_{j-1} = A_0 + (j-2)\delta g, g_j = A_n \}. \]

3. Sample from \( \pi(x|k) \) by drawing randomly from \( G_k = \{ g_j | g_j \in (A_{k-1}, A_k) \} \) with sampling weights \( \hat{f}_h(\tilde{x}_i) \) for \( k = 1, \ldots, n_k \). The sample size for each interval is given by the number of observations within each interval. Obtain \( \tilde{x}_i \) for \( i = 1, \ldots, n \).

4. Estimate any statistical indicator of interest \( \hat{I} \) using \( \tilde{x}_i \).

5. Recompute the density \( \hat{f}_h(x) \), using the pseudo samples \( \tilde{x}_i \) obtained in iteration Step 3.

6. Repeat Steps 2-5, with \( B^{(KDE)} \) burn-in and \( M^{(KDE)} \) additional iterations.

7. Discard the \( B^{(KDE)} \) burn-in iterations and estimate the final \( \hat{I} \) by averaging the obtained \( M^{(KDE)} \) estimates.

For open-ended intervals, e.g., \((15000, +\infty)\) the upper bound has to be replaced by a finite number. Walter (2019) shows through model-based simulations that a value of three times the value of the lower bound (15000, 45000) gives appropriate estimation results when working with income data.

The variance of the statistical indicators is estimated by bootstrapping. Bootstrap methods were first introduced by Efron (1979). These methods serve as an estimation procedure when the variance cannot be stated as closed-form solution (Shao and Tu, 1995). While bootstrapping avoids the problem of the non-availability of a closed-form solution, it comes with the disadvantage of long computational times. In the package, a non-parametric bootstrap that accounts for the additional uncertainty coming from the interval-censored data is implemented. This non-parametric bootstrap is introduced in Walter (2019).

### Core functionality: Direct estimation of statistical indicators

The presented KDE algorithm is implemented in the function `kdeAlgo()` (see Table 1). The arguments and default settings of `kdeAlgo()` are briefly summarized in Table 2. The function gives back an S3 object of class "kdeAlgo". A detailed explanation of all components of an "kdeAlgo" object can be found in the package documentation. The generic functions `plot()` and `print()` can be applied to "kdeAlgo" objects to output the main estimation results (see Table 1). In the next section the function `kdeAlgo()` is used to estimate a variety of statistical indicators from interval-censored EU-SILC data and its arguments are explained in more detail.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>kdeAlgo()</code></td>
<td>Estimates statistical indicators and its standard errors from interval-censored data</td>
</tr>
<tr>
<td><code>plot()</code></td>
<td>Plots convergence of the estimated statistical indicators and estimated density of the pseudo ( \tilde{x}_i )</td>
</tr>
<tr>
<td><code>print()</code></td>
<td>Prints estimated statistical indicators and its standard errors</td>
</tr>
</tbody>
</table>

### Example: Direct estimation of statistical indicators

To demonstrate the function `kdeAlgo()`, the equivalized household income and the corresponding household sample weight from the Austrian synthetic EU-SILC survey data set is used. The data set is included in the `laeken` package (Alfons et al., 2020; Alfons and Templ, 2013). Its synthetic nature makes it unusable for inferential statistics. However, the data set has the advantage over the scientific use file by being freely available which enables the easy reproducibility of the presented example. Since the total disposable household income is measured on a continuous scale, it is censored to 22 intervals for demonstration purposes. For a realistic censoring scheme the interval bounds are chosen such that they closely follow the interval bounds used in the German Microcensus from 2013 (Statistisches Bundesamt, 2014). The German Microcensus is a representative household survey that covers 830,000 persons in 370,000 households (1% of the German population) in which income is only collected as interval-censored variable (Statistisches Bundesamt, 2016).

In a first step the variable total disposable household income called `hhincome_net` is interval censored according to 22 intervals using the function `cut()`. The vector of interval bounds is called `intervals` and the newly obtained interval-censored income variable is called `c.hhincome`.
### Table 2: Arguments of function kdeAlgo().

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>xclass</td>
<td>Interval-censored variable</td>
<td></td>
</tr>
<tr>
<td>classes</td>
<td>Numeric vector of interval bounds</td>
<td></td>
</tr>
<tr>
<td>threshold</td>
<td>Threshold used for poverty indicators (% of the median of the target variable)</td>
<td>0.6</td>
</tr>
<tr>
<td>burnin</td>
<td>Number of burn-in iterations $B^{(KDE)}$</td>
<td>80</td>
</tr>
<tr>
<td>samples</td>
<td>Number of additional iterations $M^{(KDE)}$</td>
<td>400</td>
</tr>
<tr>
<td>bootstrap.se</td>
<td>If TRUE, standard errors of the statistical indicators are estimated</td>
<td>FALSE</td>
</tr>
<tr>
<td>b</td>
<td>Number of bootstraps for the estimation of the standard errors</td>
<td>100</td>
</tr>
<tr>
<td>bw</td>
<td>Smoothing bandwidth used &quot;nrd0&quot;</td>
<td></td>
</tr>
<tr>
<td>evalpoints</td>
<td>Number of evaluation grid points</td>
<td>4000</td>
</tr>
<tr>
<td>adjust</td>
<td>Bandwidth multiplier $bw = adjust \times bw$</td>
<td>1</td>
</tr>
<tr>
<td>custom_indicator</td>
<td>A list of user-defined statistical indicators</td>
<td>NULL</td>
</tr>
<tr>
<td>upper</td>
<td>If upper bound of the upper interval is $+\infty$ e.g., $(15000, +\infty)$, then $+\infty$ is replaced by $15000 \times upper$</td>
<td>3</td>
</tr>
<tr>
<td>weights</td>
<td>Survey weights</td>
<td>NULL</td>
</tr>
<tr>
<td>oecd</td>
<td>Household weights of equivalence scale</td>
<td>NULL</td>
</tr>
</tbody>
</table>

R> intervals <- c(0, 150, 300, 500, 700, 900, 1100, 1300, 1500, 1700, 2000, 2300, 2600, 2900, 3200, 3600, 4000, 4500, 5000, 5500, 6000, 7500, Inf)
R> c.hhincome <- cut(hhincome_net, breaks = intervals)

In order to get a descriptive overview of the distribution of the censored income data the function `table()` is applied.

R> table(c hhincome)
c hhincome

<table>
<thead>
<tr>
<th>(0,150]</th>
<th>(150,300]</th>
<th>(300,500]</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
<td>113</td>
<td>280</td>
</tr>
<tr>
<td>(500,700]</td>
<td>(700,900]</td>
<td>(900,1.1e+03]</td>
</tr>
<tr>
<td>462</td>
<td>1137</td>
<td>1433</td>
</tr>
<tr>
<td>(1.1e+03,1.3e+03]</td>
<td>(1.3e+03,1.5e+03]</td>
<td>(1.5e+03,1.7e+03]</td>
</tr>
<tr>
<td>2040</td>
<td>1811</td>
<td>1671</td>
</tr>
<tr>
<td>(1.7e+03,2.0e+03]</td>
<td>(2.0e+03,2.3e+03]</td>
<td>(2.3e+03,2.6e+03]</td>
</tr>
<tr>
<td>2006</td>
<td>1383</td>
<td>849</td>
</tr>
<tr>
<td>(2.6e+03,2.9e+03]</td>
<td>(2.9e+03,3.2e+03]</td>
<td>(3.2e+03,3.6e+03]</td>
</tr>
<tr>
<td>508</td>
<td>389</td>
<td>242</td>
</tr>
<tr>
<td>(3.6e+03,4.0e+03]</td>
<td>(4.0e+03,4.5e+03]</td>
<td>(4.5e+03,5.0e+03]</td>
</tr>
<tr>
<td>158</td>
<td>187</td>
<td>61</td>
</tr>
<tr>
<td>(5.5e+03,5.5e+03]</td>
<td>(5.5e+03,6.0e+03]</td>
<td>(6.0e+03,7.0e+03]</td>
</tr>
<tr>
<td>21</td>
<td>18</td>
<td>52</td>
</tr>
<tr>
<td>(7.5e+03,Inf]</td>
<td>17</td>
<td></td>
</tr>
</tbody>
</table>

Most incomes are in interval [1100, 1300] and only 17 incomes are in the upper interval. For the estimation of the statistical indicators the function `kdeAlgo()` of the `smicd` package is called with the following arguments.

R> Indicators <- kdeAlgo(
+     xclass = c hhincome, classes = intervals,
+     bootstrap.se = TRUE, custom_indicator =
+     list(
+         quant05 = function(y, threshold, weights) {
+             wtd.quantile(y, probs = 0.05, weights)
+         }
+     )
+ )
The variable c.hhincome is assigned to the argument xclass and the vector of interval bounds intervals is assigned to the argument classes. The default settings of the arguments burnin, samples, bw, evalpoints, adjust and upper are retained. Simulation results from Walter (2019) and Groß et al. (2017) show that these settings give good results when working with income data. Changing these arguments has an impact on the performance of the KDE algorithm. As default, the statistical indicators: mean, Gini coefficient, headcount ratio (HCR), the quantiles (10%, 25%, 50%, 75%, 90%), the poverty gap (PGAP) and the quintile share ratio (QSR) are estimated (Gini, 1912; Foster et al., 1984). The HCR and PGAP rely on a poverty threshold. The default choice of the threshold argument is 60% of the median of the target variable as suggested by Eurostat (2014). Besides the mentioned indicators, any other statistical indicator can be estimated via the argument custom_indicator. In the example the argument is assigned a list that holds functions to estimate the 5% and 95% quantile. The custom indicators must depend on the target variable, the threshold (even if it is not needed for the specified indicator) and optionally on the weights argument, if the estimation of a weighted indicator is required. To estimate the standard errors of all indicators bootstrap.se = TRUE and the number of bootstrap samples is 100 (the default value as suggested in Walter (2019)). Lastly, the household weight (hhweight) is assigned to the argument weights in order to estimate weighted statistical indicators. It can also be controlled for households of different sizes, by assigning oecd a variable with household equivalence weights. By applying the print() function to the "kdeAlgo" object the estimated statistical indicators (default and custom indicators) as well as their standard errors are printed. For instance, in this example the estimated mean is about 1,658 Euro and its standard error is 8.486.

R> print(Indicators)
Value:
  mean  gini  hcr  quant10  quant25  quant50
1658.329  0.265  0.145  802.227  1117.714  1507.947
quant75  quant90  pgap  qsr  quant05  quant95
2020.063  2654.707  0.040  3.920  630.326  3142.296

Standard error:
  mean  gini  hcr  quant10  quant25  quant50
  8.486  0.002  0.002  5.839  5.977  6.605
quant75  quant90  pgap  qsr  quant05  quant95
  10.548  21.622  0.001  0.044  10.327  24.401

For demonstration purposes, the statistical indicators are also estimated using the continuous household income variable from the synthetic EU-SILC data set (Table 3). The estimation results of the KDE algorithm using the interval-censored data are very close to those based on the continuous data. Slightly larger deviations are observable for the more extreme quantiles. This is due to the fact that these quantile estimates fall into intervals with a lower number of observations (compared to the other quantile estimates). Estimation results for these quantiles could potentially be further improved by increasing the number of evalpoints of the kdeAlgo() function.

Table 3: Estimated weighted statistical indicators using the continuous household income variable from the synthetic EU-SILC data set.

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>gini</th>
<th>hcr</th>
<th>quant10</th>
<th>quant25</th>
<th>quant50</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1657.910</td>
<td>0.265</td>
<td>0.144</td>
<td>805.468</td>
<td>1114.028</td>
<td>1508.657</td>
</tr>
<tr>
<td>quant75</td>
<td>quant90</td>
<td>pgap</td>
<td>qsr</td>
<td>quant05</td>
<td>quant95</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2017.585</td>
<td>2653.617</td>
<td>0.040</td>
<td>3.960</td>
<td>619.666</td>
<td>3153.425</td>
</tr>
</tbody>
</table>

In Walter (2019) the performance of the KDE algorithm is evaluated via detailed simulation studies. By applying the function plot(), "kdeAlgo" objects can be plotted. Thereby, convergence plots for all estimated statistical indicators and a plot of the estimated final density are obtained.

R> plot(Indicators)
Figure 1 shows convergence plots for two of the estimated indicators. Additionally, a plot of the estimated final density with a histogram of the observed data in the background is given in Figure 2. In Figure 1 the estimated statistical indicator (Gini, 10% quantile) for each iteration step of the KDE algorithm and the average over the estimates up to iteration step M (excluding the burn-in iterations) are plotted. A vertical line marks the end of the burn-in period. The horizontal line gives the value of the final estimate (average over the M iterations). All convergence plots indicate that the number of iterations is chosen sufficiently large for the estimates to converge.

If convergence were not achieved the arguments burnin and samples should be increased. It is notable that the estimated 10% quantile has the same value for almost all iteration steps. This is the case because the quantile, as any other statistical indicator, is estimated using the pseudo samples that are drawn on 4,000 grid points $G$. Estimating a quantile on only 4,000 unique outcomes (pseudo values) leads to equal quantile estimates for numerous iteration steps of the KDE algorithm.

Lastly, it should be mentioned that the computation time can be very long if the estimation of the standard errors is enabled. Hence, if the estimation of the standard errors is not required, the argument bootstrap.se should be set to FALSE. Furthermore, it should always be checked how many iterations are needed for the desired statistical indicator to converge. Reducing the number of required iterations (arguments burnin and samples) lowers the computation time significantly (with and without standard errors).
Figure 2: Estimated final density with a histogram of the observed distribution of the data in the background.

Regression analysis

In the following three subsections the statistical methodology for linear and linear mixed regression models with an interval-censored dependent variable is introduced, the core functionality of the functions semLM() and semLME() is presented and examination scores of students from schools in London are exemplary modeled.

Methodology: Regression analysis

The theoretical introduction of the new regression method, proposed by Walter (2019), is presented for linear mixed regression models. The theory for linear regression models can be obtained by simplifying the introduced method. In its standard form the linear mixed regression model serves to analyze the linear relationship between a continuous dependent variable and some independent variables (Goldstein, 2010). Random parameters (random slopes and random intercepts) are included in the model to account for correlated data e.g., students within schools. The model in matrix notation (Laird and Ware, 1983) is given by

$$ y = X\beta + Zv + e, $$

where $y$ is a $n \times 1$ column vector of the dependent variable, $n$ is the sample size, $X$ is a $n \times p$ matrix where $p$ is equal to the number of predictors, $\beta$ is a column vector of the fixed effects regression parameters of size $p \times 1$, $Z$ is the $n \times q$ design matrix with $q$ random effects, $v$ is a $q \times 1$ vector of random effects, and $e$ is the residual vector of size $n \times 1$. The distribution of the random effects is given by

$$ v \sim N(0, G), $$

where $G = \begin{bmatrix} \sigma_0^2 & \sigma_{01} & \cdots & \sigma_{0q} \\ \sigma_{10} & \sigma_1^2 & \cdots & \sigma_{1q} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{q0} & \sigma_{q1} & \cdots & \sigma_q^2 \end{bmatrix}$, and

and the distribution of the residuals is given by $e \sim N(0, R)$ with $R = I_n \sigma_e^2$ where $I_n$ is the identity matrix and $\sigma_e^2$ is the residual variance. The random effects $v$ and the residuals $e$ are assumed to be independent. For a more detailed introduction of mixed models see Searle et al. (1992); McCulloch et al. (2008); Snijders and Bosker (2011). In the case of an interval-censored dependent variable the parameters of Model (1) have to be estimated without observing $y$ on a continuous scale. Instead, only the interval identifier $k$, now defined as $n \times 1$ column vector, is observed. Open-ended interval bounds $A_0 = -\infty$ and $A_n = +\infty$ and unequal interval widths are allowed. Since the true distribution of $y$ is unknown the aim is to reconstruct the distribution of $y$ using the known intervals $k$ and the linear relationship stated in Model (1). As presented in Walter (2019) in order to reconstruct the unknown distribution of $f(y|X, Z, v, k, \theta)$, where $\theta = (\beta, R, G)$, the Bayes theorem (Bayes, 1763) is applied.
Hence,
\[ f(y|X, Z, v, k, \theta) \propto f(k|y, X, Z, v, \theta) f(y|X, Z, v, \theta), \]
with \( f(k|y, X, Z, v, \theta) = f(k|y) \) because the conditional distribution of the interval identifier \( k \) only depends on \( y \). It is given by \( f(k|y) = r \) with \( r \) being a \( n \times 1 \) column vector \( r = (r_1, r_2, \ldots, r_n)^T \) with
\[
r_i = \begin{cases} 
1 & \text{if } A_{k-1} \leq y_i \leq A_k, \\
0 & \text{else},
\end{cases}
\]
for \( i = 1, \ldots, n \) and
\[ f(y|X, Z, v, \theta) \sim N(X\beta + Zv, R). \] (2)
The relationship in Equation (2) follows from the linear mixed model assumptions (Model (1)). The unknown parameters \( \theta = (\beta, R, G) \) are estimated based on pseudo samples \( \tilde{y} \) (since \( y \) is unknown) that are iteratively drawn from \( f(y|X, Z, v, k, \theta) \). The next subsection states the computational details of the SEM algorithm.

**Estimation and computational details**

To fit Model (1), the parameter vector \( \tilde{\theta} = (\tilde{\beta}, \tilde{R}, \tilde{G}) \) is estimated and pseudo samples of the unknown \( y \) are iteratively generated by the following SEM algorithm. The pseudo samples \( \tilde{y} \) are drawn from the conditional distribution
\[ f(y|X, Z, v, k, \theta) \propto I(A_{k-1} \leq y \leq A_k) \times N(X\beta + Zv, R), \]
where \( I(\cdot) \) denotes the indicator function. Hence, for \( y \) with explanatory variables \( X \) the corresponding \( \tilde{y} \) is drawn from \( N(X\tilde{\beta} + Z\tilde{v}, \tilde{R}) \) conditional on the given interval \( (A_{k-1} \leq y \leq A_k) \). If \( \tilde{\theta} \) is estimated the conditional distribution \( f(y|X, Z, v, k, \theta) \) follows a two-sided truncated normal distribution. Its probability density function equals
\[ \tilde{f}(y|X, Z, \tilde{v}, k, \tilde{\theta}) = \frac{\phi\left(\frac{y - \tilde{\beta}}{\tilde{v}}\right)}{\Phi\left(\frac{A_k - \tilde{\beta}}{\tilde{v}}\right) - \Phi\left(\frac{A_{k-1} - \tilde{\beta}}{\tilde{v}}\right)}, \] (3)
with \( \tilde{\mu} = X\tilde{\beta} + Z\tilde{v} \), \( \phi(\cdot) \) denotes the probability density function of the standard normal distribution and \( \Phi(\cdot) \) denotes its cumulative distribution function. From its definition it follows that \( \Phi\left(\frac{A_k - \tilde{\beta}}{\tilde{v}}\right) = 1 \) if \( A_k = +\infty \) and \( \Phi\left(\frac{A_{k-1} - \tilde{\beta}}{\tilde{v}}\right) = 0 \) if \( A_{k-1} = -\infty \). The steps of the SEM algorithm as described in Walter (2019) are:

1. **Estimation step:** \( \hat{\theta} = (\hat{\beta}, \hat{R}, \hat{G}) \) from Model (1) using the midpoints of the intervals as substitutes for the unknown \( y \). The parameters are estimated by restricted maximum likelihood theory (REML) (Thompson, 1962).
2. **Stochastic step:** For \( i = 1, \ldots, n \), draw randomly from \( N(X\hat{\beta} + Z\hat{v}, \hat{R}) \) within the given interval \( (A_{k-1} \leq y \leq A_k) \) (the two-sided truncated normal distribution given in Equation (3)) obtaining \( (\tilde{y}, X, Z) \). The drawn pseudo \( \tilde{y} \) are used as replacements for the unobserved \( y \).
3. **Maximization step:** Reestimate the parameter vector \( \hat{\theta} \) from Model (1) by using the pseudo samples \( (\tilde{y}, X, Z) \) from Step 2. Again, parameter estimation is carried out by REML.
4. **Iterate Steps 2-3** \( B^{(SEM)} + M^{(SEM)} \) times, with \( B^{(SEM)} \) burn-in iterations and \( M^{(SEM)} \) additional iterations.
5. **Discard the burn-in iterations** \( B^{(SEM)} \) and estimate \( \hat{\theta} \) by averaging the obtained \( M^{(SEM)} \) estimates.

If open-ended intervals \( A_0 = -\infty \) and \( A_n = +\infty \) are present, the midpoints \( M_1 \) and \( M_n \) of these intervals in iteration Step 1 are computed as follows:
\[ M_1 = (A_1 - \bar{D}) / 2, \]
\[ M_n = (A_{n-1} + \bar{D}) / 2, \]
where
\[ \bar{D} = \frac{1}{(n_k - 2)} \sum_{k=2}^{n_k} |A_{k-1} - A_k|. \]
These midpoints serve as proxies for the unknown interval midpoints in Step 1 of the algorithm. The SEM algorithm for the linear regression model is obtained by simplifying the conditional distribution \( f(y|X, Z, v, \theta) \sim N(X\beta + Zv, \mathbf{R}) \) to \( f(y|X, \beta, \sigma^2_e) \sim N(X\beta, \sigma^2_e) \) according to the model assumptions of a linear regression model. In the SEM algorithm for linear models it is then drawn from \( N(X\beta, \sigma^2_e) \) within the given interval.

The standard errors of the regression parameters are estimated using bootstrap methods. For the linear regression model a non-parametric bootstrap (Efron and Stein, 1981; Efron, 1982; Efron and Tibshirani, 1986, 1993) and for the linear mixed regression model a parametric bootstrap (Wang et al., 2006; Thai et al., 2013) is used to estimate the standard errors. The non-parametric as well as the parametric bootstrap are further developed to account for the additional uncertainty that is due to the interval-censored dependent variable. Both newly proposed bootstraps are available in the \texttt{smicd} package and the theory is explained in (Walter, 2019).

To assure that the model assumptions are fulfilled the logarithmic and the Box-Cox transformations are incorporated into the function \texttt{semLm()} and \texttt{semLme()}.

**Core functionality: Regression analysis**

The introduced SEM algorithm is implemented in the functions described in Table 4. The arguments and default settings of the estimation functions \texttt{semLm()} and \texttt{semLme()} are summarized in Table 5. Both functions return a an \texttt{S3} object of class \texttt{"sem" "lm"} or \texttt{"sem" "lme"}. A detailed explanation of all the components of these objects can be found in the \texttt{smicd} package documentation. The generic functions \texttt{plot()}, \texttt{print()} and \texttt{summary()} can be applied to objects of class \texttt{"sem" "lm"} and \texttt{"sem" "lme"} in order to summarize the main estimation results. In the next section the functionality of \texttt{semLm()} and \texttt{semLme()} is demonstrated based on an illustrative example.

### Table 4: Implemented functions for the estimation of linear and linear mixed regression models.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{semLm()}</td>
<td>Estimates linear regression models with an interval-censored dependent variable</td>
</tr>
<tr>
<td>\texttt{semLme()}</td>
<td>Estimates linear mixed regression models with an interval-censored dependent variable</td>
</tr>
<tr>
<td>\texttt{plot()}</td>
<td>Plots convergence of the estimated parameters and estimated density of the pseudo ( \tilde{y} ) from the last iteration step</td>
</tr>
<tr>
<td>\texttt{print()}</td>
<td>Prints basic information of the estimated linear and linear mixed regression models</td>
</tr>
<tr>
<td>\texttt{summary()}</td>
<td>Summary of the estimated linear and linear mixed regression models</td>
</tr>
</tbody>
</table>

### Table 5: Arguments of functions \texttt{semLm()} and \texttt{semLme()}.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{formula}</td>
<td>A two-sided linear formula object</td>
<td></td>
</tr>
<tr>
<td>\texttt{data}</td>
<td>A data frame containing the variables of the model</td>
<td></td>
</tr>
<tr>
<td>\texttt{classes}</td>
<td>Numeric vector of interval bounds</td>
<td></td>
</tr>
<tr>
<td>\texttt{burnin}</td>
<td>Burn-in iterations</td>
<td>40</td>
</tr>
<tr>
<td>\texttt{samples}</td>
<td>Additional iterations</td>
<td>200</td>
</tr>
<tr>
<td>\texttt{trafo}</td>
<td>Transformation of the dependent variable: None, logarithmic or Box-Cox transformation</td>
<td>\texttt{&quot;None&quot;}</td>
</tr>
<tr>
<td>\texttt{adjust}</td>
<td>Extends the number of iterations for the estimation of the Box-Cox transformation parameter: ( (\text{burnin} + \text{samples}) \times \text{adjust} )</td>
<td>2</td>
</tr>
<tr>
<td>\texttt{bootstrap.se}</td>
<td>If TRUE standard errors and confidence intervals of the regression parameters are estimated</td>
<td>\texttt{FALSE}</td>
</tr>
<tr>
<td>\texttt{b}</td>
<td>Number of bootstraps for the estimation of the standard errors</td>
<td>100</td>
</tr>
</tbody>
</table>
Example: Regression analysis

To demonstrate the functions \texttt{semLm()} and \texttt{semLme()} the famous London school data set that is analyzed in Goldstein et al. (1993) is used. The data set contains the examination results of 4,059 students from 65 schools in six Inner London Education Authorities. The data set is available in the R package \texttt{mlmRev} (Bates et al., 2020a) and also included in the package \texttt{smicd}. The variables used in the following example are: general certificate of secondary examination scores (examsc), the standardized London reading test scores at the age of 11 years (standLRT), the sex of the student (sex), and the school identifier (school). In the original data set the variable examsc is measured on a continuous scale. In order to demonstrate the functionality of the functions \texttt{semLm()} and \texttt{semLme()} the variable is arbitrarily censored to nine intervals. As before, the censoring is carried out by the function \texttt{cut()} and the vector of interval bounds is called \texttt{intervals}.

\r
\begin{verbatim}
R> intervals <- c(1, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.7, 8.5, Inf)
R> Exam$examsc.class <- cut(Exam$examsc, intervals)
\end{verbatim}

The newly created interval-censored variable is called \texttt{examsc.class}. The distribution is visualized by applying the function \texttt{table()}.

\r
\begin{verbatim}
R> table(Exam$examsc.class)
\end{verbatim}

\begin{verbatim}
(1,1.5] (1.5,2.5] (2.5,3.5] (3.5,4.5] (4.5,5.5] (5.5,6.5] (6.5,7.7] (7.7,8.5] (8.5,Inf]
 1 32 249 937 1606 951 267 15 1
\end{verbatim}

It can be seen that most examination scores are concentrated in the center intervals. To fit the linear regression model the function \texttt{semLM()} is called.

\r
\begin{verbatim}
R> LM <- semLm(
+   formula = examsc.class ~ standLRT + sex, data = Exam,
+   classes = intervals, bootstrap.se = TRUE
+ )
\end{verbatim}

The formula argument is assigned the model equation, where \texttt{examsc.class} is regressed on \texttt{standLRT} and \texttt{sex}. The argument data is assigned the name of the data set \texttt{Exam} and the vector of interval bounds \texttt{intervals} is assigned to the \texttt{classes} argument. The arguments \texttt{burnin} and \texttt{samples} are left as defaults. The specified number of default iterations is sufficiently large for most regression models, however, convergence of the parameters has to be checked by plotting the estimation results with the function \texttt{plot()} after the estimation. No transformation is specified for the interval-censored dependent variable therefore \texttt{trafo} is assigned its default value. The argument \texttt{adjust} is only relevant if the Box-Cox transformation \texttt{trafo="bc"} is chosen. In this case the number of iterations for the estimation of the Box-Cox transformation parameter \(\lambda\) can be specified by this argument. The convergence of the transformation parameter \(\lambda\) has to be checked using the function \texttt{plot()}. More information on the Box-Cox transformation and on the estimation of the transformation parameter is given in Walter (2019). For the estimation of the standard errors of the regression parameters the argument \texttt{bootstrap.se} is set to \texttt{TRUE}. The number of bootstrap samples \(b\) is 100, its default value, which again is reasonable for most settings. A summary of the estimation results is obtained by the application of the function \texttt{summary()}.  

\r
\begin{verbatim}
R> summary(LM)
Call:
semLm(formula = examsc.class ~ standLRT + sex, data = Exam, 
   classes = intervals, bootstrap.se = TRUE)
Fixed effects:
   Estimate Std.Error Lower 95%-level Upper 95%-level
(Intercept) 5.0702791 0.01816102 5.0326739 5.1033905
standLRT 0.5908015 0.01349275 0.5614197 0.6163845
sexM -0.1715966 0.03093346 -0.2308930 -0.1010877

Multiple R-squared: 0.3501 Adjusted R-squared: 0.3498

Variable examsc.class is divided into 9 intervals.
\end{verbatim}

The output shows the function call, the estimated regression coefficients, the bootstrapped standard errors, and the confidence intervals as well as the R-squared and the adjusted R-squared. Furthermore, the output reminds the user that the dependent variable is censored to nine intervals. All estimates are
interpreted as in a linear regression model with a continuous dependent variable, hence, if standLRT increases by one unit and all other parameters are kept constant, examsc.class increases by 0.59 on average. The bootstrapped confidence intervals indicate that all regressors have a significant effect on the dependent variable.

By using the generic function plot() on an object of class "sem" "lm" convergence plots of each estimated regression parameter and of the estimated residual variance are obtained. Furthermore, the density of the generated pseudo \( \tilde{y} \) variable from the last iteration step is plotted with a histogram of the observed distribution of the interval-censored variable examsc.class in the background.

\[
R> \text{plot}(LM)
\]

![Convergence plots of estimated model parameters (\( \beta_1 \) and \( \sigma_e \)).](image)

In Figure 3 a selection of convergence plots is given and in Figure 4 the density of the pseudo \( \tilde{y} \) from the last iteration step of the SEM algorithm is plotted. In the convergence plots the estimated parameter and the average up to iteration step \( M \) (excluding \( B \)) are plotted for each iteration step of the SEM algorithm. A vertical line indicates the end of the burn-in period (40 iterations). The final parameter estimate marked by the horizontal line is obtained by averaging the \( M^{(SEM)} \) additional iterations (200). The selected 240 iterations are enough to obtain reliable estimates in this example, because the estimates have converged.

As already mentioned, the smicd package also enables the estimation of linear mixed regression models by the function semLme(). In the London school data set students are nested within schools, therefore it is necessary to control for the correlation within-schools. In order to do that the variable
school is specified as a random intercept. If necessary, a random slope parameter could also be included in the model. Again, the variable sex is included as an additional regressor. Hence, the formula argument is assigned the following model equation \( \text{examsc.class} \sim \text{standLRT} + \text{sex} + (1 | \text{school}) \). So far, the function semLme() enables the estimation of linear mixed models with a maximum of one random slope and one random intercept parameter. Regarding all other arguments, the same specifications as before are made.

```r
R> LME <- semLme(
+   formula = examsc.class ~ standLRT + sex + (1 | school),
+   data = Exam, classes = intervals, bootstrap.se = TRUE
+ )
```

By using the generic function summary() the estimation results are printed. In addition to the fixed effects, the estimated random effects are obtained as in the \texttt{lme4} and \texttt{nlme} packages. Since the R-squared and the adjusted R-squared are not defined for mixed models the summary() function prints the marginal R-squared and conditional R-squared (Nakagawa and Schielzeth, 2013; Johnson, 2014).

```r
> summary(LME)
Call:
semLme(formula = examsc.class ~ standLRT + sex + (1 | school),
data = Exam, classes = intervals, bootstrap.se = TRUE)

Random effects:
Groups   Name   Variance  Std.Dev.
school (Intercept) 0.08755431 0.2958958
Residual           0.58417586 0.7643140

Fixed effects:
               Estimate  Std.Error Lower 95%-level Upper 95%-level
(Intercept)  5.0777581 0.0005188930  5.0769749  5.0791095
standLRT    0.5605049 0.0003665976  0.5599456  0.5613711
sexM       -0.1711065 0.0008159909 -0.1724193 -0.1692369

Marginal R-squared: 0.324 Conditional R-squared: 0.4121
Variable examsc.class is divided into 9 intervals.
```

Again, interpretation is the same as in linear mixed models with a continuous dependent variable. By applying the generic function plot() to an “sem” “lme” object the same plots as for the linear regression model are plotted.
Discussion and outlook

Asking for interval-censored data can lead to lower item non-response rates and increased data quality. While item non-response is potentially avoided, applying traditional statistical methods becomes infeasible because the true distribution of the data within each interval is unknown. The functions of the smicd package enable researchers to easily analyze this kind of data. The paper briefly introduces the new statistical methodology and presents, in detail, the core functions of the package:

- `kdeAlgo()` for the direct estimation of any statistical indicator,
- `semLm()` to estimate linear models with an interval-censored dependent variable,
- `semLme()` to estimate linear mixed models with an interval-censored dependent variable.

The functions are applied in order to estimate statistical indicators from interval-censored synthetic EU-SILC income data and to analyze interval-censored examination scores of students from London with linear and linear mixed regression models.

Further developments of the smicd package will include the possibility to estimate the bootstrapped standard errors in parallel computing environments. Additionally, it is planned to allow for the use of survey weights in the linear (mixed) regression models.

Bibliography


B. Ripley. MASS: Support Functions and Datasets for Venables and Ripley’s MASS, 2019. URL: https://CRAN.R-project.org/package=MASS. R package version 7.3-51.5. [p2]


Paul Walter
Freie Universität Berlin
Garystraße 21, 14195 Berlin
Germany
paul.walter@fu-berlin.de