The R Package emdi for Estimating and Mapping Regionally Disaggregated Indicators

Ann-Kristin Kreutzmann
Freie Universität Berlin

Sören Pannier
Freie Universität Berlin

Natalia Rojas-Perilla
Freie Universität Berlin

Timo Schmid
Freie Universität Berlin

Matthias Templ
Zurich University
of Applied Sciences

Nikos Tzavidis
University of Southampton

Abstract

The R package **emdi** offers a methodological and computational framework for the estimation of regionally disaggregated indicators using small area estimation methods and provides tools for assessing, processing and presenting the results. A range of indicators that includes the mean of the target variable, the quantiles of its distribution and complex, non-linear indicators or customized indicators can be estimated simultaneously using direct estimation and the empirical best predictor (EBP) approach (Molina and Rao 2010). In the application presented in this paper package **emdi** is used for estimating inequality indicators and the median of the income distributions for small areas in Austria. Because the EBP approach relies on the normality of the mixed model error terms, the user is further assisted by an automatic selection of data-driven transformation parameters. Estimating the uncertainty of small area estimates (using a mean squared error - MSE measure) is achieved by using both parametric bootstrap and semi-parametric wild bootstrap. The additional uncertainty due to the estimation of the transformation parameter is also captured in MSE estimation. The semi-parametric wild bootstrap further protects the user against departures from the assumptions of the mixed model in particular, those of the unit-level error term. The bootstrap schemes are facilitated by computationally efficient code that uses parallel computing. The package supports the users beyond the production of small area estimates. Firstly, tools are provided for exploring the structure of the data and for diagnostic analysis of the model assumptions. Secondly, tools that allow the spatial mapping of the estimates enable the user to create high quality visualizations. Thirdly, results and model summaries can be exported to Excel™ spreadsheets for further reporting purposes.

Keywords: official statistics, parallel computation, small area estimation, visualization.

1. Introduction

In recent years an increased number of policy decisions has been based on statistical information derived from indicators estimated at disaggregated geographical levels using small area estimation methods. Clearly, the more detailed the information provided by official statistics estimates, the better the basis for targeted policies and evaluating intervention programs.

The United Nations suggest further disaggregation of statistical indicators for monitoring the Sustainable Development Goals (SDGs). National Statistical Institutes (NSIs) and other organizations across the world have also recognized the potential of producing small area statistics and their use for informing policy decisions. Examples of NSIs with well-developed programs in the production of small area statistics include the US Bureau of Census, the UK Office for National Statistics (ONS) and the Statistical Office of Italy (ISTAT). Although the term domain is more general as it may include non-geographic dimensions, the term small area estimation (SAE) is the established one. We shall follow the custom in this paper and use the terms area/geography and domain/aggregation interchangeably.

Without loss of generality in this paper we will assume that the primary data sources used to estimate statistical indicators are national socio-economic household sample surveys. Sample surveys are designed to provide estimates with acceptable precision at national and possibly sub-national levels but usually have insufficient sizes to allow for precise estimation at lower geographical levels. Therefore, direct estimation that relies only on the use of survey data can be unreliable or even not possible for domains that are not represented in the sample. In the absence of financial resources for boosting the sample size of surveys, using model-based methodologies can help to obtain reliable estimates for the target domains.

Model-based SAE methods (Rao and Molina 2015; Pfeffermann 2013) work by using statistical models to link survey data, that are only available for a part of the target population, with administrative or census data that are available for the entire population. Despite the wide range of SAE methods that have been proposed by academic researchers, these are so far applied only by a fairly small number of NSIs or other practitioners. This gap between theoretical advances and applications may have several reasons one of which is the lack of suitable, user friendly statistical software. More precisely, software needs not only to be available but it also needs to simplify the application of the methods for the user. The R (R Core Team 2017) package **emdi** (Kreutzmann, Pannier, Rojas-Perilla, Schmid, Templ, and Tzavidis 2017) aims to improve the user experience by simplifying the estimation of small area indicators and corresponding precision estimates. Furthermore, the user benefits from support that extends beyond estimation in particular, evaluating, processing and presenting the results.

Traditionally model-based SAE methods have been employed for estimating simple, linear indicators for example, means and totals using for example, mixed (random) effects models and empirical best linear unbiased predictors (EBLUPs). Several software packages exist. In R, the package **JoSAE** (Breidenbach 2015) includes functions for EBLUPs using unit-level models. Functions in the package **hbsae** (Boonstra 2012) enable the use of unit- and area-level models and can be estimated either by using restricted maximum likelihood (REML) or hierarchical Bayes methods. The package **BayesSAE** (Shi and with contributions from Peng Zhang 2013) also allows for Bayesian methods. The **rsae** package by Schoch (2012) and package **saeRobust** by Warnholz (2016) provide functions for outlier robust small area estimation using unit- or area-level models. Gaussian area-level multinomial mixed-effects models for SAE can be done with the **mme** package (Lopez-Vizcaino, Lombardia, and Morales 2014). In addition, resources in R for SAE are available through the BIAS project (`www.bias-project.uk`) and in the package **SAE2** (Gómez-Rubio, Salvati *et al.* 2008). In Stata, functions `xtmixed` and `gllamm` support the estimation of linear mixed models, which is a popular basis for model-based SAE. EBLUPs can be derived using these functions (West, Welch, and Galecki 2007). Similarly, `PROC MIXED` and `PROC IML` can be used for fitting unit-

and area-level models in SAS as shown in Mukhopadhyay and McDowell (2011).

More recently a widespread application of SAE methods involves the estimation of poverty and inequality indicators and distribution functions (The World Bank 2007). In this case the use of methodologies for estimating means and totals is no longer appropriate since such indicators are complex, non-linear functions of the data. As an example, we refer to the Foster-Greer-Thorbecke indicators (Foster, Greer, and Thorbecke 1984), the Gini coefficient (Gini 1912) and the quantiles of the income distribution. Popular SAE approaches for estimating complex indicators include the Empirical Best Predictor (EBP) (Molina and Rao 2010), the World Bank method (Elbers, Lanjouw, and Lanjouw 2003) and the M-Quantile method (Chambers and Tzavidis 2006; Tzavidis, Marchetti, and Chambers 2010). Although in this paper we focus exclusively on software for implementing the EBP method (Molina and Rao 2010), future version of the package will include the M-Quantile and World Bank methods. The World Bank provides free software for using the World Bank method called PovMap (The World Bank Group 2013). However, this focuses exclusively on poverty mapping. Creating a more general open-source software can help to accelerate the uptake of modern model-based methods. Currently, the most well known package that includes the EBP method is the R package **sae** (Molina and Marhuenda 2015). Package **emdi** attempts to improve some of the less attractive features of existing packages by offering more options and greater flexibility to the user. In particular, package **emdi** offers the following attractive features:

- The package simplifies the estimation of indicators for small areas and its precision estimates by tailored functions.

- These functions return by default estimates for a set of predefined indicators, including the mean, the quantiles of the distribution of the response variable and poverty and inequality indicators.

- Self-defined indicators or indicators available within other packages can be included.

- The user can select the type of data transformation to be used in **emdi**. Data-driven transformation parameters are estimated automatically.

- In contrast to other packages that include only a parametric bootstrap MSE estimator, package **emdi** includes two bootstrap methods, a parametric bootstrap and a semi-parametric wild bootstrap for MSE estimation. Both capture the uncertainty due to the estimation of the transformation parameter. The use of wild bootstrap (Thai, Mentré, Holford, Veyrat-Follet, and Comets 2013; Flachaire 2005) protects the user against departures from the distributional assumptions of the mixed model. This offers additional robustness.

- Parallel computing is provided in a customized manner for reducing the computational time associated with the use of bootstrap.

- Package **emdi** provides predefined functions for diagnostic checks of the underlying model, if model-based estimation is chosen. A mapping tool for spatially plotting the estimated indicators enables the creation of high quality visualization. An informative output summarizing the most relevant results can be exported to Excel™ for presentation and reporting purposes.

The remaining of this paper is structured as follows. Section 2 presents a data set that is used to demonstrate the functionality of the package **emdi**. Information about the estimation methods that are included in the package is given in Section 3. Section 4 describes the core functionality of the package. Examples demonstrate the use of the methods for computing, assessing and presenting the estimates. Section 5 shows how users can extend the set of indicators to be estimated by including customized options. Finally, Section 6 discusses future potential extensions of the package.

2. Data sets

SAE methods make use of multiple data sources. Package **emdi** contains two example data sets (`eusilcA_smp` and `eusilcA_pop`) and one shape file (`shape_austria_dis`) that are used for illustrating its use. The two data sets are based on the data `eusilcP` from the package **simFrame** (Alfons, Templ, and Filzmoser 2010). This data set is a simulated close-to-reality version of the European Union Statistics on Income and Living Conditions (EU-SILC) for Austria from 2006. The original EU-SILC data is obtained from an annual household survey that is nowadays conducted in all EU member states and six other European countries and enables the analysis of income, socio-demographic factors and living conditions. The population data set `eusilcA_pop` differs from the data set provided in Alfons *et al.* (2010) in three ways:

1. The data set `eusilcA_pop` contains information only at the household level.
2. It includes a reduced number (17 instead of 28) of variables. All variables are described in the R documentation of the two data sets.
3. As the data set provided in Alfons *et al.* (2010) contains identifiers for states but not district identifiers, we assigned randomly households to districts for illustrating the use of SAE methods. In this case districts are the target domains.

This leads to a synthetic population with 25000 households in 96 districts. The sample data set `eusilcA_smp` is a sample of this population with 1000 observations drawn by simple random sampling. The first three observations of four selected variables from `eusilcA_pop` are printed below.

```
data("eusilcA_pop")
head(eusilcA_pop, 3)[, c("eqIncome", "eqsize", "cash", "district")]

    eqIncome eqsize cash   district
724 8603.387    1.5    0 Eisenstadt
667 8605.750    1.0    0 Eisenstadt
156 8656.020    1.0    0 Eisenstadt
```

In addition to SAE methods, package **emdi** provides a function called `map_plot` that produces maps of the estimated indicators. In order to demonstrate the use of the function `map_plot` package **emdi** contains a shape file for the 96 Austrian districts which is downloaded from the Global Administrative Areas website (Hijmans 2015). This shape file is saved in `.RData` format and the object `shape_austria_dis` is a `SpatialPolygonsDataFrame`.

```
R> load_shapeaustria()
R> class(shape_austria_dis)

[1] "SpatialPolygonsDataFrame"
attr(,"package")
[1] "sp"
```

3. Statistical methodology

In order to obtain regionally disaggregated indicators, package **emdi** includes direct estimation and currently model-based estimation using the EBP approach by Molina and Rao (2010). The predefined indicators returned by the estimation functions in **emdi** include the mean and quantiles (10%, 25%, 50%, 75%, 90%) of the target variable as well as non-linear indicators of the target variable. A widely used family of indicators measuring income deprivation and inequality is the Foster-Greer-Thorbecke (FGT) one (Foster *et al.* 1984). Package **emdi** includes the FGT measures of Head Count Ratio (HCR) and Poverty Gap (PG). In order to compute the HCR and PG indicators one must use a threshold, also known as poverty line. This line is a minimum level of income deemed adequate for living in a particular country and can be defined in terms of absolute or relative poverty. For instance, the international absolute poverty line is currently set to $ 1.90 per day by the World Bank (The World Bank 2017). Relative poverty means a low income relative to others in a particular country - for instance, below a percentage of the median income in that country. Package **laeken** (Alfons and Templ 2013) uses relative poverty lines defined as 60% of median equivalised disposable income corresponding to the EU definition for poverty lines and thus in this case the HCR is the At-risk-of-poverty rate. In contrast, package **emdi** allows both for absolute and relative poverty lines and the user is free to set the poverty line. Therefore, the threshold can be given as an argument in **emdi** or, alternatively, the user can define an arbitrary function depending on the target variable and sampling weights. As a default, a relative threshold defined as 60% of the median target variable is used. Another family of indicators of interest is the so-called Laeken indicators, endorsed by the European Council in Laeken, Belgium (Council of the European Union 2001). Two examples of Laeken indicators that are well-known for measuring inequality are the Gini coefficient (Gini 1912) and the Income Quintile Share Ratio (QSR) (Eurostat 2004). These two inequality indicators are also available in **emdi**. Therefore, in total **emdi** includes ten predefined indicators that are estimated at domain level using `direct()` estimation and model-based estimation via the `ebp()` method.

Direct estimation relies on the use of sample data only. The definition of direct (point and variance) estimation in **emdi** follows Alfons and Templ (2013). While variance estimation in package **laeken** is only available for the poverty and inequality indicators, package **emdi** also provides non-parametric bootstrap procedures (Alfons and Templ 2013) for estimating the variance of estimates of the mean and the quantiles. The user can apply the function `direct()` as follows,

```
emdi_direct <- direct(y = "eqIncome", smp_data = eusilcA_smp,
        smp_domains = "district", weights = "weight", threshold = 10989.28,
        var = TRUE, boot_type = "naive", B = 50, seed = 123, na.rm = TRUE)
```

Arguments	Short description	Default
`y`	Target variable	
`smp_data`	Survey data	
`smp_domains`	Domain identifier	
`weights`	Sampling weights	No weights
`design`	Variable indicating strata	No design
`threshold`	Threshold for poverty indicators	60% of the median of the target variable
`var`	Variance estimation	No variance estimation
`boot_type`	Type of bootstrap: naive or calibrate	Naive
`B`	Number of bootstrap populations	50
`seed`	Seed for random number generator	123
`X_calib`	Calibration variables	None
`totals`	Population totals	None
`custom_indicator`	Customized indicators	None
`na.rm`	Deletion of observations with missing values	No deletion

Note: All explanations can also be found in the documentation of the `direct` function in the package.

Table 1: Input arguments for function `direct`.

As shown in Table 1 the user has to specify three arguments, which include the target variable, the sample data sets, and the variable name that defines the domain identifier in the sample data. For the remaining arguments suitable defaults are defined.

The implementation of the EBP method in package **emdi** is based on the theory described in Molina and Rao (2010) and Rao and Molina (2015). The underlying model is a unit-level mixed model also known in SAE literature as the nested error linear regression model (Battese, Harter, and Fuller 1988). In its current implementation the EBP method is based on a two-level nested error linear regression model that includes a random area/domain-specific effect and a unit (household or individual)-level error term.

Denote by U a finite population of size N, partitioned into D domains $U_1, U_2, \dots, U_D$ of sizes $N_1, \dots, N_D$, where $i = 1, \dots, D$ refers to an ith domain and $j = 1, \dots, N_i$ to the jth household/individual. Let $\boldsymbol{y}$ be the target variable. Assume $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_p)^T$, the design matrix, containing p explanatory variables. The nested error linear regression model is defined by

$$T(y_{ij}) = \mathbf{x}_{ij}^T\boldsymbol{\beta} + u_i + e_{ij}, \quad j = 1, \dots, n_i, \quad i = 1, \dots, D, \quad u_i \overset{iid}{\sim} N(0, \sigma_u^2), \quad e_{ij} \overset{iid}{\sim} N(0, \sigma_e^2), \tag{1}$$

where T denotes a transformation of the target variable $\mathbf{y}$. $\mathbf{x}_{ij}$ is a vector of unit-level auxiliary variables of dimension $(p+1) \times 1$, $\boldsymbol{\beta}$ is the $(p+1) \times 1$ vector of regression coefficients and u_i and e_{ij} denote the random area and unit-level error terms. The EBP approach works by using at least two data sources, namely a sample dataset used to fit the nested error linear regression model and a population (e.g. census or administrative) dataset used for predicting - under the model - synthetic values of the outcome (income in our application) for the entire population. Both data sources must share identically defined covariates but the target variable is only available in the sample dataset. Under model (1), we assume that the model error terms follow a Gaussian distribution. However, in certain applications - as is the case when analyzing economic variables - this assumption may be unrealistic. Package **emdi** includes the option of using a one-to-one transformation $T(y_{ij})$ of the target variable $\mathbf{y}$ aiming to make the Gaussian assumptions more plausible. A logarithmic transformation is very often used in practice (Molina and Rao 2010). However, this is not necessarily the optimal transformation, for example, when the model error terms do not follow exactly a log-normal distribution. In addition to a logarithmic transformation, package **emdi** allows the use of a data-driven Box-Cox transformation. The Box-Cox transformation is denoted by

$$T(y_{ij}) = \begin{cases} \frac{(y_{ij}+s)^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0; \\ \log(y_{ij}+s) & \text{if } \lambda = 0, \end{cases} \tag{2}$$

where λ is an unknown transformation parameter and s denotes the shift parameter, which is a constant and chosen automatically such that $y_{ij} + s > 0$. A general algorithm for estimating the transformation parameter λ is the residual maximum likelihood (REML), which is described in detail in Rojas-Perilla, Pannier, Schmid, and Tzavidis (2017). One advantage of using the Box-Cox transformation is that it includes the logarithmic and no-transformation as special cases for specific values of the transformation parameter λ. Package **emdi** currently includes the following options: no transformation, logarithmic transformation and Box-Cox transformation. The EBP method is implemented using the following algorithm:

1. For a given transformation obtain $T(y_{ij})$. If the user selects the Box-Cox transformation, the transformation parameter λ is automatically estimated by the **emdi** package.

2. Use the sample data to fit the nested error linear regression model and estimate $\boldsymbol{\theta}$ denoted by $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\beta}}, \hat{\sigma}_u^2, \hat{\sigma}_e^2)$. Also compute $\hat{\gamma}_i = \frac{\hat{\sigma}_u^2}{\hat{\sigma}_u^2 + \frac{\hat{\sigma}_e^2}{n_i}}$.

3. For $l = 1, \ldots, L$:

   (a) For in-sample domains (domains that are part of the sample dataset), generate a synthetic population of the target variable by $T(y_{ij}^{*(l)}) = \mathbf{x}_{ij}^T\hat{\beta} + \hat{u}_i + v_i^* + e_{ij}^*$, with $v_i^* \overset{iid}{\sim} N(0, \hat{\sigma}_u^2(1-\hat{\gamma}_i))$, $e_{ij}^* \overset{iid}{\sim} N(0, \hat{\sigma}_e^2)$ and $\hat{u}_i$, the conditional expectation of u_i given y_i. For out-of-sample domains (domains with no data in the sample) generate a synthetic population by using $T(y_{ij}^{*(l)}) = \mathbf{x}_{ij}^T\hat{\beta} + v_i^* + e_{ij}^*$, with $v_i^* \overset{iid}{\sim} N(0, \hat{\sigma}_u^2)$, $e_{ij}^* \overset{iid}{\sim} N(0, \hat{\sigma}_e^2)$.

   (b) Back-transform to the original scale $\mathbf{y}_i^{(l)} = T^{-1}(\mathbf{y}_i^{*(l)})$ and calculate the target indicator $I_i^{(l)}(\mathbf{y}_i^{(l)})$ in each domain. Note that $I_i^{(l)}$ is used here as a generic notation for any indicator of interest.

4. Compute the final estimates by taking the mean over the L Monte Carlo simulations in each domain, $\hat{I}_i^{EBP} = 1/L \sum\limits_{l=1}^{L} I_i^{(l)}(\mathbf{y}_i^{(l)})$.

Measuring the uncertainty around the EBP estimates is done by using bootstrap methods. Here the uncertainty is quantified by the Mean Squared Error (MSE). Package **emdi** includes two bootstrap schemes. One is parametric bootstrap under model (1) following Molina and

Rao (2010), which additionally includes the uncertainty due to the estimation of the transformation parameter (Rojas-Perilla *et al.* 2017). Using an appropriate transformation often reduces the departures from normality. However, even after transformations, departures from normality may still exist in particular for the unit-level error term. For this reason, **emdi** also includes a variation of semi-parametric wild bootstrap (Flachaire 2005; Thai *et al.* 2013; Rojas-Perilla *et al.* 2017) to protect against departures from the model assumptions. The semi-parametric wild bootstrap is implemented as follows,

1. Fit model (1) (using an appropriate transformation for y) to obtain estimates $\hat{\boldsymbol{\beta}}, \hat{\sigma}_u^2, \hat{\sigma}_e^2, \hat{\lambda}$.
2. Calculate the sample residuals by $\hat{e}_{ij} = y_{ij} - \mathbf{x}_{ij}^T\hat{\boldsymbol{\beta}} - \hat{u}_i$.
3. Scale and center these residuals using $\hat{\sigma}_e$. The scaled and centered residuals are denoted by $\hat{\epsilon}_{ij}$.
4. For $b = 1, ..., B$
   (a) Generate $u_i^{(b)} \overset{iid}{\sim} N(0, \hat{\sigma}_u^2)$.
   (b) Calculate the linear predictor $\eta_{ij}^{(b)}$ by $\eta_{ij}^{(b)} = \mathbf{x}_{ij}^T\hat{\boldsymbol{\beta}} + u_i^{(b)}$.
   (c) Match $\eta_{ij}^{(b)}$ and the sample $\hat{\eta}_k = \mathbf{x}_{ij}^T\hat{\boldsymbol{\beta}} + \hat{u}_i \;\; (k \in n)$ using
   $$\min_{k \in n} \left| \eta_{ij}^{(b)} - \hat{\eta}_k \right|$$
   and define $\tilde{k}$ as the corresponding index.
   (d) Generate weights w from a distribution satisfying the conditions in Feng, He, and Hu (2011) where w is a simple two-point mass distribution with probabilities 0.5 at $w = 1$ and $w = -1$, respectively.
   (e) Calculate the bootstrap population as $T(y_{ij}^{(b)}) = \mathbf{x}_{ij}^T\hat{\boldsymbol{\beta}} + u_i^{(b)} + w_{\tilde{k}}|\hat{\epsilon}_{\tilde{k}}^{(b)}|$.
   (f) Back-transform $T(y_{ij}^{(b)})$ to the original scale and compute the bootstrap population value $I_{i,b}$.
   (g) Select the bootstrap sample and use the EBP method as described above.
   (h) Obtain $\hat{I}_{i,b}^{EBP}$.
5. $\widehat{MSE}_{Wild}\left(\hat{I}_i^{EBP}\right) = B^{-1}\sum_{b=1}^{B}\left(\hat{I}_{i,b}^{EBP} - I_{i,b}\right)^2$.

A simulation study comparing the performance of both MSE estimators is presented in Rojas-Perilla *et al.* (2017). Since the bootstrap schemes presented here are computationally intensive, **emdi** includes an option for parallelization.

The EBP approach is implemented in **emdi**, using function `ebp`. As shown in Table 2 the user has to specify five arguments, which include the structure of the fixed effects of the nested error linear regression model, the two data sets (sample and population), and the variable names that define the domain identifiers in each data set. For the remaining arguments suitable defaults are defined. The choice of a transformation is simplified since the user only has to choose the type of transformation. The shift parameter s and the optimal transformation

Arguments	Short description	Default
`fixed`	Fixed effects formula of the nested error regression model	
`pop_data`	Census or administrative data	
`pop_domains`	Domain identifier for population data, `pop_data`	
`smp_data`	Survey data	
`smp_domains`	Domain identifier for sample data, `smp_data`	
`L`	Number of Monte Carlo iterations	50
`threshold`	Threshold for poverty indicators	60% of the median of the target variable
`transformation`	Type of transformation: no, log or Box-Cox	Box-Cox
`interval`	Interval for the estimation of the optimal transformation parameter	(-1,2)
`MSE`	Mean Squared Error (MSE) estimation	No MSE estimation
`B`	Number of bootstrap populations	50
`boot_type`	Type of bootstrap: parametric or wild	Parametric
`parallel_mode`	Mode of parallelization	Automatic
`cpus`	Number of kernels for parallelization	1
`custom_indicator`	Customized indicators	None
`na.rm`	Deletion of observations with missing values	No deletion

Note: All explanations can also be found in the documentation of the `ebp` function in the package.

Table 2: Input arguments for function `ebp`.

parameter λ in the case of using the Box-Cox transformation are automatically estimated. This distinguishes **emdi** from package **sae** where the user must choose the transformation parameters manually. Since the Box-Cox transformation includes the no transformation and logarithmic transformation as special cases, this is chosen as the default option.

An example of using the `ebp` with the EU-SILC data to compute point and MSE estimates for the predefined indicators and two custom indicators, namely the minimum and maximum equivalised income is provided below:

```
set.seed(100)
emdi_model <- ebp(fixed = eqIncome ~ gender + eqsize + cash + self_empl +
     unempl_ben + age_ben + surv_ben + sick_ben + dis_ben +
     rent + fam_allow + house_allow + cap_inv + tax_adj,
     pop_data = eusilcA_pop, pop_domains = "district",
     smp_data = eusilcA_smp, smp_domains = "district",
     MSE = TRUE,
     custom_indicator = list(my_max = function(y, threshold){max(y)},
                             my_min = function(y, threshold){min(y)}))
```

4. Basic design and core functionality

The previous section presented the statistical methodology that uses either direct estimation or the model-based EBP approach. A key benefit offered by **emdi** is the flexibility for assessing, presenting and exploring the results. The following commands enable this flexibility.

1. Get summary statistics and model diagnostics.
   `> summary()`
2. Graphical presentation of model diagnostics.
   `> plot()`
3. Extract the indicators of interest.
   `> estimators()`
4. Visualize the estimated indicators.
   `> map_plot()`
5. Export the results to Excel™.
   `> write.excel()`

The package **emdi** uses the S3 object system (Chambers and Hastie 1992). All objects created in the package **emdi** by an estimation function (`direct()` and `ebp()`) share the class `emdi`. Objects of class `emdi` comprise nine components, which are presented in Table 3. Some of these components are specific only to model-based estimation, such that they are `NULL` for direct estimation. These components are indicated in the second column of Table 3. Depending on the estimation method, the `emdi` object is also of class `direct` or `model`.

```
R> class(emdi_direct)

[1] "emdi"  "direct"

R> class(emdi_model)

[1] "emdi"  "model"
```

Thus, the commands can be tailored to the estimation method, e.g. model diagnostics (provided by the command `plot()`) are only suitable when a model-based approach is used. In what follows the **emdi** functionalities are illustrated for the object `emdi_model`.

4.1. Data information and model summary

R-users typically use a `summary` method for summarizing the results. For `emdi` objects the summary outputs differ depending on the two classes. The summary for objects obtained by direct estimation gives information about the number of domains in the sample, the total and domain-specific sample sizes. The summary for model-based objects is more extensive. In addition to information about the sample sizes, information about the population size and the number of out-of-sample domains is provided. Since model-based SAE relies on prediction

Position	**Name**	**Short description**	**Available for `direct`**
1	`ind`	Point estimates for indicators per domain	✓
2	`MSE`	Variance/MSE estimates per domain	✓
3	`transform_param`	Transformation and shift parameters	
4	`model`	Fitted linear mixed-effects model as `lme` object	
5	`framework`	List with 8 components describing the data	✓
6	`transformation`	Type of transformation	
7	`method`	Estimation method for transformation parameter	
8	`fixed`	Formula of fixed effects used in the nested error linear regression model	
9	`call`	Image of the function call that produced the object	✓

Note: All explanations can be found in the documentation of the `emdi` object in the package.

Table 3: Components of `emdi` objects.

under the model, including model diagnostics in **emdi** is important for users. A first measure to consider when evaluating the working model is the well known R^2. Nakagawa and Schielzeth (2013) provide a generalization of this measure for linear mixed models. A marginal R^2 and a conditional (a measure that accounts for the random effect) R^2 are implemented via function `r.squaredGLMM` in package **MuMIn** (Barton 2015). The `summary` method uses this function to calculate and present both measures. For the EBP and model-based SAE methods in general the validity of parametric assumptions is crucial. Therefore, **emdi** also outputs residual diagnostics. In particular, results include the skewness and the kurtosis of both sets of residuals (random effects and unit-level) and the results from using the Shapiro-Wilk test for normality (test statistic and p-value). The intra-cluster correlation (ICC) coefficient is further used for assessing the remaining unobserved heterogeneity and hence the importance of random effects for prediction. Finally, the `summary` command gives information about the selected transformation. If the user opts for a Box-Cox transformation, the transformation parameter λ and the shift parameter s will be reported. The summary output of the object `emdi_model` is presented below.

```
R> summary(emdi_model)

Empirical Best Prediction

Call:
ebp(fixed = eqIncome ~ gender + eqsize + cash + self_empl + unempl_ben +
age_ben + surv_ben + sick_ben + dis_ben + rent + fam_allow +
house_allow + cap_inv + tax_adj, pop_data = eusilcA_pop,
pop_domains = "district", smp_data = eusilcA_smp, smp_domains = "district",
MSE = TRUE, custom_indicator = list(my_max = function(y,
threshold) {
        max(y)
}, my_min = function(y, threshold) {
min(y)
}))

Out-of-sample domains:  3
In-sample domains:  93

Sample sizes:
Units in sample:  1000
Units in population:  25000
                   Min. 1st Qu. Median   Mean 3rd Qu. Max.
Sample_domains        1     4.0    8.0  10.75      10  236
Population_domains   26   103.8  180.5 260.40     265 5857

Explanatory measures:
 Marginal_R2 Conditional_R2
   0.6010076      0.8042044

Residual diagnostics:
               Skewness Kurtosis Shapiro_W    Shapiro_p
Error         0.3102283 3.743571 0.9922285 4.136833e-05
Random_effect 0.1534680 2.768637 0.9887039 6.131442e-01

ICC:  0.509275

Transformation:
 Transformation Method Optimal_lambda Shift_parameter
        box.cox   reml      0.6725132               0
```

4.2. Diagnostic plots

In addition to the diagnostics provided by `summary`, **emdi** enables the use of graphical diagnostics (see Figure 1). The `plot` method outputs graphics of residual diagnostics. The first set of plots (Figure 1a) are Normal Quantile-Quantile (Q-Q) plots of Pearson unit-level residuals and standardized random effects. Figure 1b and 1c are kernel density plots of the distribution of the two sets of residuals contrasted against a standard normal distribution. Outliers can have a significant impact on the model fit and hence on prediction. Hence, a Cook's distance plot is also available (Figure 1d), in which the three largest values of the standardized residuals are identified alongside the case identification number for further investigation. Finally, if a Box-Cox transformation is used, a plot of the profile log-likelihood that shows the value of the transformation parameter for which the likelihood is maximized is also produced (Figure 1e). The user can customize the format of all plots. Method `plot` accepts the parameter `label` with the predefined values `blank` (deletes all labels) and `no_title` (axis labels are given, but no plot titles). In addition, a user-defined list that contains specific labels for each plot list can be given. Another parameter available is `color` which accepts a vector with two color specifications. The first color defines the lines in Figure 1a, 1d and 1e and the second one specifies the color of the shapes in Figure 1b and 1c. For the likelihood plot the range in which the likelihood should be computed can be specified by using the parameter `range`. The appearance of the plots benefits from the use of the **ggplot2** package (Wickham 2009; Wickham and Chang 2016). Hence, `plot` accepts a `gg_theme` argument which allows for all customization options of `theme` that is a tool for modifying non-data components of a plot. The plots shown in Figure 1 can be produced as follows,

```
R> plot(emdi_model, label = "no_title", color = c("red3", "red4"))
```

4.3. Selection of indicators

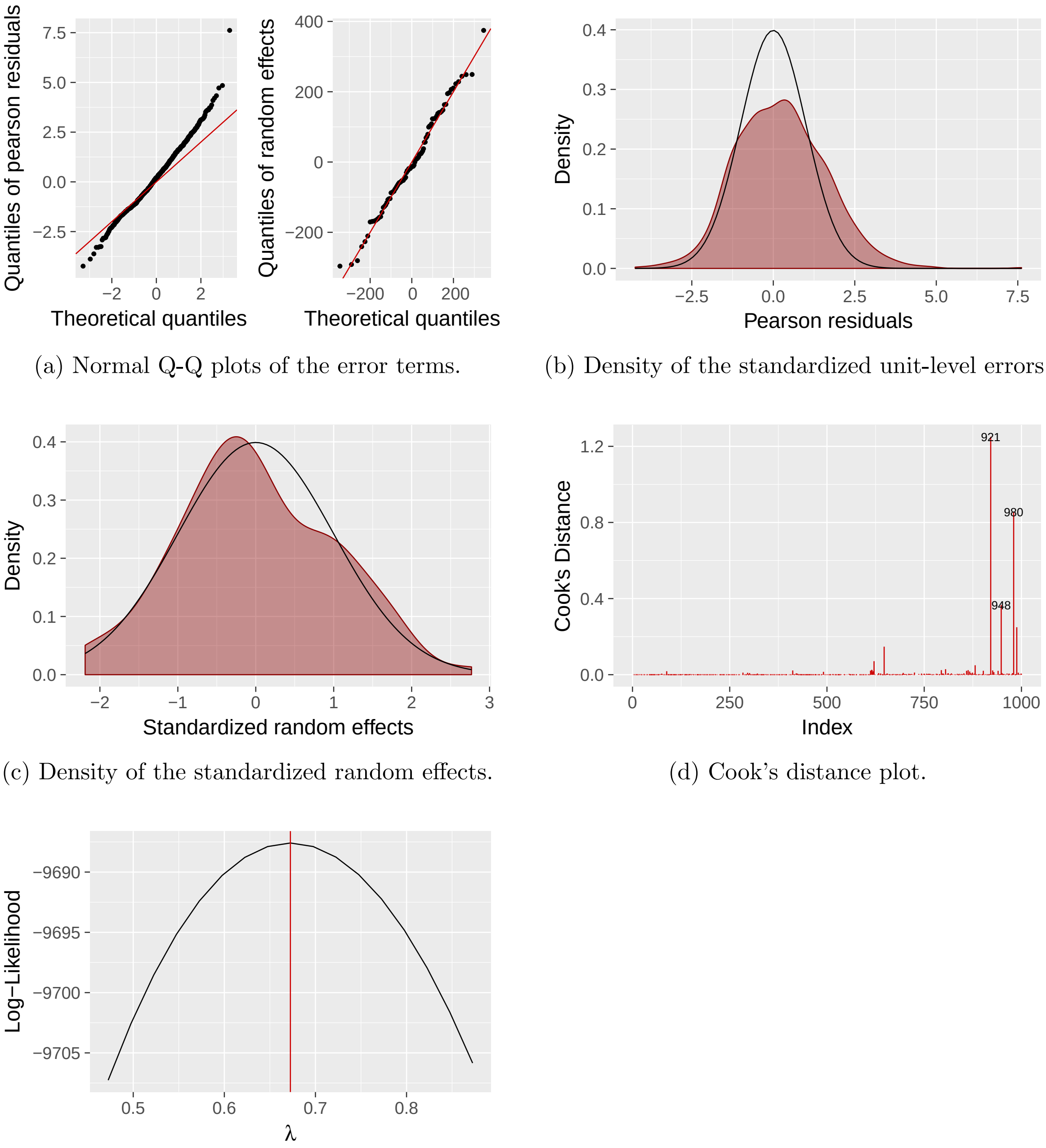
Package **emdi** returns a set of predefined and customized indicators. However, the user may only be interested in some of these or only in individually defined (customized) indicators. A function called `estimators` helps the user to select the indicator or indicators of interest. This is done by using the `indicator` argument that takes a vector of indicator names as an argument, but in addition also accepts keywords defining predefined groups; for example, the keyword `custom` returns only user-defined indicators. In addition to variance and MSE estimates NSIs often use the Coefficient of Variation (CV) as an additional measure of the quality of the estimates. Estimated CVs can be returned alongside MSE estimates. The following example shows how to extract model-based (EBP) estimates of the median and the Gini coefficient and corresponding CVs. Results are presented for the first six domains (Austrian regions).

```
R> head(estimators(emdi_model, indicator = c("Gini", "Median"),
+                    MSE = FALSE, CV = TRUE))
```

```
          Domain      Gini     Gini_CV    Median  Median_CV
1      Amstetten 0.2940598 0.05949443  4360.795 0.24890869
2          Baden 0.2188246 0.08781998  7539.109 0.19356550
3        Bludenz 0.2497461 0.15150355  7276.480 0.31953963
4       Bodensee 0.1816816 0.06335170 12386.151 0.07821249
5 Braunau am Inn 0.2477799 0.06558771  6613.448 0.13537806
6        Bregenz 0.1497810 0.09754445 16269.048 0.09255829
```

4.4. Spatial mapping of the estimates

In SAE maps are a natural way to present the estimates as they help describing the spatial distribution of issues like poverty and inequality. Creating maps can be demanding or laborious in practice. Package **emdi** includes function `map_plot` that simplifies the creation of maps. Given a spatial polygon provided by a shape file and a corresponding `emdi` object `map_plot` produces maps of selected indicators and corresponding MSE and CV estimates. For function `map_plot` to work the user must define a mapping table in the form of a data frame that matches the domain variable in the population data set with the domain variable in the shape file. If the domain identifiers in both data sources match, this table is not



(a) Normal Q-Q plots of the error terms.

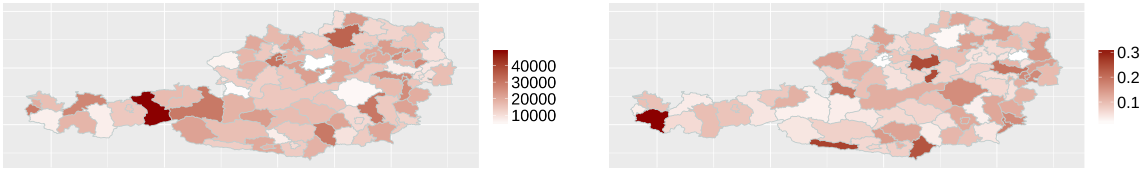
(b) Density of the standardized unit-level errors.

(c) Density of the standardized random effects.

(d) Cook's distance plot.

(e) Profile log-likelihood for the optimal parameter of the Box-Cox transformation.

Figure 1: Graphics obtained by using `plot(emdi_model)`. (a) shows Normal Q-Q plots of the unit-level errors and the random effects. (b) and (c) show kernel density estimates of the distributions of standardized unit-level errors and standardized random effects compared to a standard normal distribution (black density). The Cook's distance plot is displayed in (d) whereby the index of outliers is labeled. The profile log-likelihood for the optimal parameter value of the Box-Cox transformation is shown in (e).



(a) Median income.

(b) CV of the median income estimates.

Figure 2: Maps of point estimates and CVs of the median income for 96 districts in Austria.

required. The parameters `MSE`, `CV` and `indicator` correspond to those in the `estimators` function. The handling of the spatial shape files can be done using package **maptools** (Bivand and Lewin-Koh 2016) in combination with package **rgeos** (Bivand and Rundel 2016). The steps for obtaining a map of median income in Austrian districts and the corresponding CV are outlined below. The resulting maps can be seen in Figure 2.

First, the shape file needs to be loaded.

```
R> load_shapeaustria()
```

Next, the mapping table is defined.

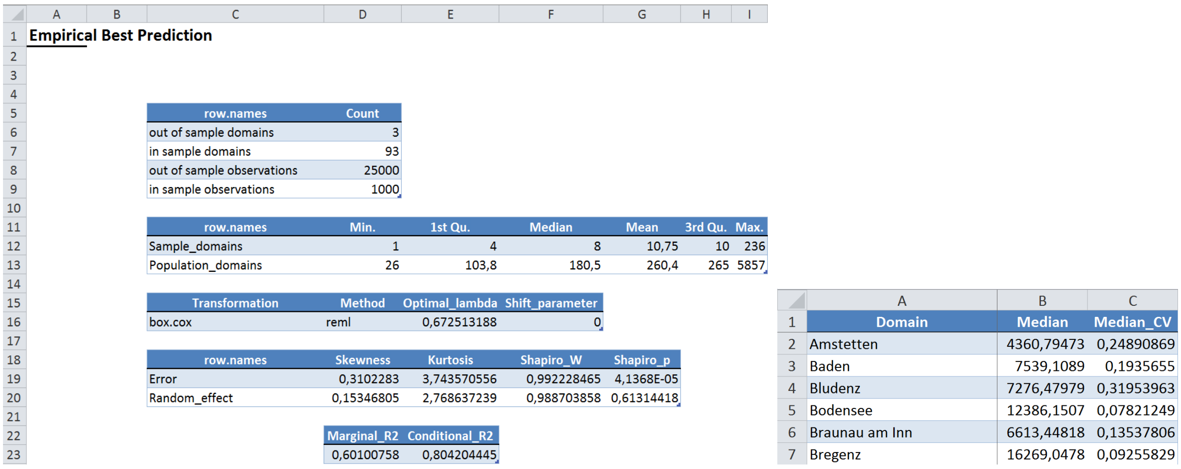
```
R> mapping_table <- data.frame(unique(eusilcA_pop$district),
+                               unique(shape_austria_dis$NAME_2))
```

Finally, two maps are created (cf. Figure 2).

```
R> map_plot(emdi_model, MSE = FALSE, CV = TRUE, map_obj = shape_austria_dis,
+           indicator = "Median", map_dom_id = "NAME_2",
+           map_tab = mapping_table)
```

4.5. Exporting the results

Exporting the results from R to other widely used software such as Excel™ is important for users. For doing so a large set of well established tools already exists. Nevertheless, exporting all model information, including the information contained in the summary output is not straightforward. Function `write.excel` creates a new Excel™ file that contains the summary output in the first sheet and the results from the selected estimators in the following sheet. Again the parameters `MSE`, `CV` and `indicator` correspond to those in the `estimators` function. The link with the Excel™ file format is done by using the package **openxlxs** (Walker 2015). This package does not require a Java™ installation, which offers an advantage over the use of the **xlsx** package (Dragulescu 2014) because Java™ may be seen as a potential security threat. Nevertheless, package **openxlxs** needs a zipping application available to R. Under windows this can be achieved by installing RTools while under MacOS or Linux such an application is available by default. Excel™ outputs can be obtained by the following command. The output is presented in Figure 3.



Empirical Best Prediction

row.names	Count
out of sample domains	3
in sample domains	93
out of sample observations	25000
in sample observations	1000

row.names	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Sample_domains	1	4	8	10,75	10	236
Population_domains	26	103,8	180,5	260,4	265	5857

Transformation	Method	Optimal_lambda	Shift_parameter
box.cox	reml	0,672513188	0

row.names	Skewness	Kurtosis	Shapiro_W	Shapiro_p
Error	0,3102283	3,743570556	0,992228465	4,1368E-05
Random_effect	0,15346805	2,768637239	0,988703858	0,61314418

Marginal_R2	Conditional_R2
0,60100758	0,804204445

Domain	Median	Median_CV
Amstetten	4360,79473	0,24890869
Baden	7539,1089	0,1935655
Bludenz	7276,47979	0,31953963
Bodensee	12386,1507	0,07821249
Braunau am Inn	6613,44818	0,13537806
Bregenz	16269,0478	0,09255829

Figure 3: Export of the summary output and estimates to Excel™.

```
R> write.excel(emdi_model, file ="excel_output.xlsx", indicator = "Median",
+              MSE = FALSE, CV = TRUE)
```

5. Incorporating a foreign estimator

A feature we should pay attention to is the ease by which indicators of foreign packages can be incorporated in package **emdi**. This is demonstrated by using the Theil index from the R package **ineq** (Zeileis 2014). The Theil index describes economic inequality and thus can be also used in the application with the data of this paper. As the function `ineq` only requires a numeric vector of the target variable, it can be straightforwardly wrapped into a form usable within the `direct` or `ebp` functions. Using the function `direct` the Theil index can be estimated as follows.

First, the package **ineq** needs to be installed and loaded.

```
install.packages("ineq")
library("ineq")
```

Subsequently, the function `ineq` with `type = "Theil"` can be given to the argument `custom_indicator`. As the function `direct()` needs the arguments `y`, `weights` and `threshold`, these arguments have to be also specified in the newly defined function. Note that weights are currently only included in direct estimation.

```
R> my_theil = function(y, weights, threshold){
+     ineq(x = y, type = "Theil")
+  }
```

The argument `custom_indicator` always needs to be a named list of self-defined indicators.

```
R> my_indicators <- list(theil = my_theil)
```

```
R> emdi_direct2 <- direct(y = "eqIncome", smp_data = eusilcA_smp,
+          smp_domains = "district", weights = "weight", var = TRUE,
+          B = 50, seed = 123,
+          custom_indicator = my_indicators,
+          na.rm=TRUE)
```

As the Theil index is now part of the `emdi` object, all methods shown in Section 4 can be also used for this newly defined inequality indicator. For instance, by estimating a customized indicator via the function `direct()` a bootstrap variance estimation is provided and the `subset` method can be applied in order to get results for certain districts.

```
R> subset(estimators(emdi_direct2, indicator = "theil", CV = TRUE),
+         Domain == "Wien")

   Domain     theil theil_CV
87   Wien 0.1238717 0.104269
```

6. Conclusion and future developments

In this paper we show how the **emdi** package can simplify the application of SAE methods. This package is, to the best of our knowledge, the first R SAE package that supports the user beyond estimation in the production of complex, non-linear indicators. Another important feature is that data-driven transformation parameters are estimated automatically. Estimating the uncertainty of small area estimates is achieved by using both parametric bootstrap and semi-parametric wild bootstrap. The additional uncertainty due to the estimation of the transformation parameter is also captured in MSE estimation. The complexity in applying SAE methods is considerably reduced, useful diagnostic tools are incorporated and the user is also supported by the availability of tools for presenting, visualizing and further processing the results. Since **emdi** makes the application of SAE methods in R almost as simple as fitting a linear or a generalized linear regression model, it also has the potential to close the gap between theoretical advances in SAE and their application by practitioners.

Additional features will be integrated in future versions of the package. Firstly, the implementation of alternative SAE methods will increase the usage of the package. For example, the ELL (Elbers *et al.* 2003) and M-Quantile (Chambers and Tzavidis 2006; Tzavidis *et al.* 2010) methods complement the EBP approach (Molina and Rao 2010) for estimating disaggregated complex, non-linear indicators. Secondly, including evaluation and diagnostic tools for comparing direct and model-based estimates will assist the user with deciding which estimation method should be preferred. Thirdly, currently **emdi** includes only some possible types of transformations and one estimation method for the transformation parameter, namely REML. Future versions of the package will include a wider range of transformations (e.g. log shift and dual power transformations) and alternative estimation methods (minimization of the skewness or measures of symmetry) for the transformation parameter.

Acknowledgments

Rojas-Perilla, Schmid and Tzavidis gratefully acknowledge support by grant ES/N011619/1

- Innovations in Small Area Estimation Methodologies from the UK Economic and Social Research Council. The work of Kreutzmann and Schmid has been also supported by the German Research Foundation within the project QUESSAMI (SCHM 3113/2-1).

References

Alfons A, Templ M (2013). "Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**." *Journal of Statistical Software*, **54**(15), 1–25. URL http://www.jstatsoft.org/v54/i15/.

Alfons A, Templ M, Filzmoser P (2010). "An Object-Oriented Framework for Statistical Simulation: The R Package **simFrame**." *Journal of Statistical Software*, **37**(3), 1–36. URL http://www.jstatsoft.org/article/view/v037i03.

Barton K (2015). ***MuMIn**: Multi-Model Inference*. R package version 1.15.1, URL https://CRAN.R-project.org/package=MuMIn.

Battese G, Harter R, Fuller W (1988). "An Error-Components Model for Prediction of County Crop Areas Suing Survey and Satellite Data." *Journal of the American Statistical Association*, **83**(401), 28–36.

Bivand R, Lewin-Koh N (2016). ***maptools**: Tools for Reading and Handling Spatial Objects*. R package version 0.8-39, URL https://CRAN.R-project.org/package=maptools.

Bivand R, Rundel C (2016). ***rgeos**: Interface to Geometry Engine - Open Source (GEOS)*. R package version 0.3-19, URL https://CRAN.R-project.org/package=rgeos.

Boonstra H (2012). ***hbsae**: Hierarchical Bayesian Small Area Estimation*. R package version 1.0, URL https://CRAN.R-project.org/package=hbsae.

Breidenbach J (2015). ***JoSAE**: Functions for some Unit-Level Small Area Estimators and their Variances*. R package version 0.2.3, URL https://CRAN.R-project.org/package=JoSAE.

Chambers J, Hastie T (1992). *Statistical Models in S*. Wadsworth & Brooks/Cole computer science series. Wadsworth & Brooks/Cole Advanced Books & Software.

Chambers R, Tzavidis N (2006). "M-quantile Models for Small Area Estimation." *Biometrika*, **93**(2), 255–268.

Council of the European Union (2001). "Report on Indicators in the Field of Poverty and Social Exclusions." *Report*, European Union.

Dragulescu AA (2014). ***xlsx**: Read, write, format Excel 2007 and Excel 97/2000/XP/2003 files*. R package version 0.5.7, URL https://CRAN.R-project.org/package=xlsx.

Elbers C, Lanjouw J, Lanjouw P (2003). "Micro-level Estimation of Poverty and Inequality." *Econometrica*, **71**(1), 355–364.

Eurostat (2004). "Common Cross-Sectional EU Indicators Based on EU-SILC; the Gender Pay Gap." *EU-SILC 131-rev/04*, Unit D-2: Living conditions and social protection, Directorate D: Single Market, Employment and Social statistics, Eurostat, Luxembourg.

Feng X, He X, Hu J (2011). "Wild Bootstrap for Quantile Regression." *Biometrika*, **98**(4), 995.

Flachaire E (2005). "Bootstrapping Heteroskedastic Regression Models: Wild Bootstrap vs. Pairs Bootstrap." *Computational Statistics & Data Analysis*, **49**(2), 361–376.

Foster J, Greer J, Thorbecke E (1984). "A Class of Decomposable Poverty Measures." *Econometrica*, **52**(3), 761–766.

Gini C (1912). *Variabilità e Mutabilità. Contributo allo Studio delle Distribuzioni e delle Relazioni Statistiche.* P. Cuppini, Bologna.

Gómez-Rubio V, Salvati N, *et al.* (2008). ***SAE2**: Small Area Estimation with R.* R package version 0.09.

Hijmans R (2015). "Global Administrative Areas." Version 2.8 [accessed: 20.10.2016], URL http://gadm.org/country.

Kreutzmann AK, Pannier S, Rojas-Perilla N, Schmid T, Templ M, Tzavidis N (2017). ***emdi**: Estimating and Mapping Disaggregated Indicators.* R package version 1.1.0, URL https://CRAN.R-project.org/package=emdi.

Lopez-Vizcaino E, Lombardia M, Morales D (2014). *mme: Multinomial Mixed Effects Models.* R package version 0.1-5, URL https://CRAN.R-project.org/package=mme.

Molina I, Marhuenda Y (2015). "**sae**: An R Package for Small Area Estimation." *The R Journal*, **7**(1), 81–98. URL http://journal.r-project.org/archive/2015-1/molina-marhuenda.pdf.

Molina I, Rao J (2010). "Small Area Estimation of Poverty Indicators." *The Canadian Journal of Statistics*, **38**(3), 369–385.

Mukhopadhyay PK, McDowell A (2011). "Small Area Estimation for Survey Data Analysis Using SAS Software." *Paper 336-2011*, SAS Institute Inc.

Nakagawa S, Schielzeth H (2013). "A General and Simple Method for Obtaining R^2 from Generalized Linear Mixed-effects Models." *Methods in Ecology and Evolution*, **4**(2), 133–142.

Pfeffermann D (2013). "New Important Developments in Small Area Estimation." *Statistical Science*, **28**(1), 40–68.

Rao JNK, Molina I (2015). *Small Area Estimation.* John Wiley & Sons.

R Core Team (2017). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rojas-Perilla N, Pannier S, Schmid T, Tzavidis N (2017). "Data-driven Transformations in Small Area Estimation." Working Paper.

Schoch T (2012). "Robust Unit-Level Small Area Estimation: A Fast Algorithm for Large Datasets." *Austrian Journal of Statistics*, **41**(4), 243–265.

Shi C, with contributions from Peng Zhang (2013). ***BayesSAE**: Bayesian Analysis of Small Area Estimation.* R package version 1.0-1, URL https://CRAN.R-project.org/package=BayesSAE.

Thai HT, Mentré F, Holford NH, Veyrat-Follet C, Comets E (2013). "A Comparison of Bootstrap Approaches for Estimating Uncertainty of Parameters in Linear Mixed-effects Models." *Pharmaceutical statistics*, **12**(3), 129–140.

The World Bank (2007). "More Than a Pretty Picture: Using Poverty Maps to Design Better Policies and Interventions." *Report*, The International Bank for Reconstruction and Development - The World Bank.

The World Bank (2017). "Measuring Poverty." [accessed: 27.04.2017], URL http://www.worldbank.org/en/topic/measuringpoverty.

The World Bank Group (2013). "Software for Poverty Mapping." [accessed: 11.02.2016], URL http://go.worldbank.org/QG9L6V7P20.

Tzavidis N, Marchetti S, Chambers R (2010). "Robust Estimation of Small Area Means and Quantiles." *Australian and New Zealand Journal of Statistics*, **52**(2), 167–186.

Walker A (2015). ***openxlsx**: Read, Write and Edit XLSX Files.* R package version 3.0.0, URL https://CRAN.R-project.org/package=openxlsx.

Warnholz S (2016). ***saeRobust**: Robust Small Area Estimation.* R package version 0.1.0, URL https://CRAN.R-project.org/package=saeRobust.

West B, Welch K, Galecki A (2007). *Linear Mixed Models A Practical Guide Using Statistical Software.* Taylor & Francis Group, LLC.

Wickham H (2009). ***ggplot2**: Elegant Graphics for Data Analysis.* Springer-Verlag, New York.

Wickham H, Chang W (2016). ***ggplot2**: Create Elegant Data Visualisations Using the Grammer of Graphics.* R package version 2.2.1, URL https://cran.r-project.org/web/packages/ggplot2/index.html.

Zeileis A (2014). ***ineq**: Measuring Inequality, Concentration, and Poverty.* R package version 0.2-13, URL https://CRAN.R-project.org/package=ineq.

Affiliation:

Timo Schmid
Institute for Statistics and Econometrics
Faculty of Economics
Freie Universität Berlin
14195 Berlin, Germany

E-mail: timo.schmid@fu-berlin.de
URL: http://www.wiwiss.fu-berlin.de/fachbereich/vwl/Schmid/Team/Schmid.html