The R Package SAEforest

Patrick Krennmair
Freie Universität Berlin

Abstract

The R package **SAEforest** promotes the use of Mixed Effects Random Forests (MERFs) for applications of Small Area Estimation. The package effectively combines functions for the estimation of spatially disaggregated linear and non-linear indicators using survey sample data. Models increase the precision of direct estimates from survey data, combining unit-level or aggregated covariate information from auxiliary data. Included procedures facilitate the estimation of domain-level economic and inequality metrics and assess associated uncertainty. The package provides procedures to simplify the analysis of model performance of MERFs and enables the visualization of predictive relations from covariates. Additionally, the package includes a function for fine-tuning of required hyper-parameters. General emphasis lies on straightforward interpretation and mapping of results.

Keywords: official statistics, mixed effects random forests, small area estimation, poverty mapping.

1. Introduction

Reliably measurable metrics are imperative to monitor demographic, economic and social development. Typically national statistical offices produce and administer elaborate statistical indicators based on survey data. With increasing availability of (alternative) data sources, research institutes and multilateral organizations aim to quantify precise information at a finer geographical resolution. The terms 'domain' or 'area' define separate entities within a joint population, such as (but not limited to) districts within a country. Many surveys are designed to produce accurate estimates at national (or sub-national levels). With deliberated disaggregation of domains, the accuracy of direct estimates decreases with domain-specific sample sizes and model-based small area estimation (SAE) offers promising tools. By combining auxiliary data sources via models with survey data, SAE methods implicitly increase the effective precision of domain-specific indicators of target variable y_{ij}. Overviews of existing methods for SAE are found in Pfeffermann (2013), Rao and Molina (2015) or Tzavidis *et al.* (2018).

Predominant models for SAE are conceptualized within the regression-setting and the majority relies on linear mixed models (LMM) to account for the hierarchical structure of survey data (Rao and Molina 2015). The predictive performance of parametric models relies on the fulfilment of (Gaussian) model-assumptions, but economic and inequality data is often highly skewed and characterized by deviations from the normal distribution. Jiang and Rao (2020) maintain that methodological improvements in SAE must focus on robustification of models

against model-failure (e.g. providing insurances against model-misspecification, valid variable selection and the effective handling of outliers). Optimality results of parametric LMMs depend on the validity of model-assumptions, which becomes challenging for applications dealing with social and economic inequality data. Existing strategies to cope with deviations from (Gaussian) assumptions are, for instance, (data-driven) transformation strategies of the dependent variable (Molina and Martín 2018; Sugasawa and Kubokawa 2019; Rojas-Perilla *et al.* 2020) or less restrictive assumptions on unit-level models (Diallo and Rao 2018; Graf *et al.* 2019). In the presence of outliers, means can be determined using robustified LMMs (Sinha and Rao 2009) or M-quantile approaches (Chambers and Tzavidis 2006), which estimate non-linear indicators without a formal specification of random effects (Tzavidis *et al.* 2010; Marchetti and Tzavidis 2021). Opsomer *et al.* (2008) use penalized splines regression for the estimation of are-level means, dealing with non-linearities by treating spline coefficients as additional random effects.

Machine learning methods offer non-linear and nonparametric alternatives, combining excellent predictive performance and a reduced risk of model-misspecification. Krennmair and Schmid (2022) introduce mixed effects random forests (MERF) as versatile tools for applications in model-based SAE. MERFs combine advantages of regression forests (e.g. implicit model-selection and robust predictive performance in the presence of outliers) with the ability to model hierarchical dependencies. Package **SAEforest** provides a coherent user-friendly framework facilitating the use of MERFs for the estimation of spatially disaggregated (non-)linear indicators and their respective uncertainty, measured by reliable mean squared errors (MSE).

In recent years, ongoing methodological contributions in (model-based) SAE are increasingly complemented by the development of open-source R-packages. I aim to give a comprehensive overview of existing SAE related packages on the Comprehensive R Archive Network (CRAN) focussing on unit-level models. Moreover, I aim to discuss existing packages dealing with random forests under dependent data sources, to motivate the functionality of the **SAEforest** package:

The package **sae** (Molina and Marhuenda 2015) offers a suitable collection of SAE methods for point and uncertainty estimates for area and unit-level models. Package **emdi** (Kreutzmann *et al.* 2019) focusses on the estimation of disaggregated economic and inequality indicators (and respective uncertainty) and insures against model-misspecification implementing an EBP under data-driven transformations (Rojas-Perilla *et al.* 2020). The package treats the EBP by Molina and Rao (2010) as a special case and combines computationally efficient methods with a genuine workflow on data processing and presentation of results. Additional packages for unit-level survey data are package **JoSAE** (Breidenbach 2018), which focuses on models coping with heteroskedasticity. From a Bayesian perspective, the package **hbsae** (Boonstra 2012) combines functions for various unit- and area-level models, bridging frequentist and Bayesian perspectives. A complete Bayesian workflow for the estimation demographic and health indicators is found in package **SUMMER** (Li *et al.* 2021). Outlier-robust estimators from a Bayesian perspective are provided by package **robustsae** (Ghosh *et al.* 2016) and from a more frequentist perspective by **saeRobust** (Warnholz 2018) or the **rsae** package Schoch (2014).

Existing packages for dependent data and tree-based machine learning methods are not concerned with topics of SAE and hardly focus on inference. The package **LongituRF** (Capitaine 2020), bundles functions that allow for time-invariant covariance structures and rely on a

semi-parametric unit-level mixed model for regression trees and forests. Although the primary focus of package **MixRF** (Wang and Chen 2016) is the imputation of clustered and incomplete data, the package comprises a genuine function, with which MERFs can be estimated. Functions from package **RandomForestGLS** (Saha *et al.* 2021) model spatial random effects as Gaussian processes by developing dependency adjusted split-criteria handling dependent error processes similarly to generalized least squares. Package **splinetree** (Neufeld and Heggeseth 2019), builds regression trees and random forests for longitudinal or dependent data using a spline projection method.

The major aim of package **SAEforest** is the provision of a complete and coherent use of MERFs for SAE. Current packages with a focus on random forests for dependent data are not intended to estimate SAE indicators and associated measures of uncertainty. On the other hand, existing unit-level SAE packages neglect tree-based methods. The use of MERFs in SAE promotes general flexibility for domain-level predictions and package **SAEforest** combines methods on the estimation of point and MSE estimates for various indicators.

Implemented estimators rely on the empirical and methodological contributions introducing MERFs for SAE of means by Krennmair and Schmid (2022), for non-linear indicators by Krennmair *et al.* (2022a) as well as in the case of aggregated auxiliary information by (Krennmair *et al.* 2022b). The flexibility of the package does not only stem from methodological aspects, but from the provision of a genuine workflow for practitioners of SAE. **SAEforest** puts emphasis on the integration of methods and generic functions that facilitate the summary and visualization of results. Additionally, predefined tools for diagnostics and the tuning of MERF hyper-parameters are available, such as the number of trees (`num.trees`) or the number of randomized split-candidates at each node (`mtry`). Implemented functions for MERFs are easily adaptable and allow for potential extensions to advanced patterns of correlation and multilevel structures.

The paper is organized as follows: Section 2 provides an overview of the statistical methodology used in the package. This includes a formal introduction to MERFs, details on the estimation of domain-level means with unit-level and aggregated covariates, as well as the estimation of non-linear indicators and corresponding MSEs. Section 3 describes data sources used as examples in the package. The core functionality of the package and its features are explained in Section 4. Section 5 summarizes methods and results and raises ideas for further research.

2. Statistical Methodology

This section introduces a general mixed model enabling a simultaneous discussion of traditional LMM-based models in SAE, such as the nested error regression model of Battese *et al.* (1988) and semi-parametric interpretations, such as the model of Krennmair and Schmid (2022) using MERFs. Machine learning methods are popular alternatives for predictive modelling in various scientific disciplines (Varian 2014; Efron 2020). Tree-based, data-driven prediction algorithms (such as random forests (Breiman 2001)) combine flexible modelling properties without explicit model assumption. Moreover, they identify complex higher-order relations in covariates and show robustness properties in the presences of outliers (Hastie *et al.* 2009; Biau and Scornet 2016). Thus, random forests contribute to the robustification of models against model-failure (Jiang and Rao 2020). In order to become a genuine tool for SAE,

predictive data-driven procedures must meet basic premises of survey and inference theory, such as the handling of hierarchically dependent data structures and measures of uncertainty for produced indicators.

In the following subsections, we will discuss the estimation of reliable domain-specific statistical indicators from survey data using MERFs and focus on their respective MSEs. Additional emphasis lies on the estimation of area-level means without population micro-data. The methods introduced are illustrated as part of an example on synthetic Austrian income data in Section 4 and rely on the theoretical and empirical methods provided by Krennmair and Schmid (2022) and Krennmair *et al.* (2022b) for means and Krennmair *et al.* (2022a) for non-linear indicators.

2.1. A general mixed effects model for SAE and MERFs

We assume a finite population U of size N consisting of D domains $U_1, U_2, ..., U_D$ with $N_1, N_2, ..., N_D$ units, where index $i = 1, ..., D$ denotes respective areas. For every individual observation j in area i in the sample, we observe the continuous target variable y_{ij}. We draw sample s of size n from population U and sampled observations are assigned to D respective areas resulting in sample sizes $n_1, n_2, ..., n_D$. A sub-sample from area i is denoted by s_i and corresponding non-sampled observations are denoted by r_i. The p predictive covariates $\mathbf{x}_{ij} = (x_1, x_2, ..., x_p)^\intercal$ are assumed to be available for every unit within the sample s. The following general mixed effects regression model describes the relationship between $\mathbf{x}_{ij}$ and y_{ij}:

$$y_{ij} = f(\mathbf{x}_{ij}) + u_i + e_{ij} \quad \text{with} \quad u_i \sim N(0, \sigma_u^2) \quad \text{and} \quad e_{ij} \sim N(0, \sigma_e^2). \tag{1}$$

Function $f(\mathbf{x}_{ij})$ models the conditional mean of y_{ij} given $\mathbf{x}_{ij}$. The hierarchical structure of observations is captured by area-specific random intercepts u_i and we assume independence between u_i and unit-level errors e_{ij}.

For instance, defining $f(\mathbf{x}_{ij}) = \mathbf{x}_{ij}^\intercal \beta$ with $\beta = (\beta_1, ..., \beta_p)^\intercal$ resembles the definition of the nested error regression model by Battese *et al.* (1988), which serves as basis for a majority of unit-level SAE-models. Well known examples are the EBP by Molina and Rao (2010) or the EBP under data-driven transformations by Rojas-Perilla *et al.* (2020). Under known optimality results of LMMs, optimal estimates of fixed effects $\hat{\beta}$ and variance components $\hat{\sigma}_u^2, \hat{\sigma}_e^2$ are obtained by maximum likelihood (ML) or restricted maximum likelihood (REML) (Rao and Molina 2015).

We combine predictive advantages of random forests with the ability to model hierarchical structures of survey data with random effects by defining f in Model 1 to be a random forest (Breiman 2001). Resulting MERFs rely on a procedure reminiscent of the EM-algorithm (Hajjem *et al.* 2014) to obtain optimal estimates on model-components $\hat{f}$, $\hat{u}$, $\hat{\sigma}_u^2$, and $\hat{\sigma}_e^2$. The proposed MERF algorithm fits parameters for Model 1 (where f is a random forest) by iteratively estimating a) the forest function, assuming the random effects term to be correct and b) the random effects part, assuming the Out-of-Bag-predictions (OOB-predictions) from the forest to be correct. OOB-predictions correspond to the unused observations in the internal bootstrap step prior to the construction of each forest's sub-tree (Breiman 2001; Biau and Scornet 2016). We estimate variance components $\hat{\sigma}_\epsilon^2$ and $\hat{\sigma}_u^2$ by implicitly taking the expectation of ML estimators given the data. Computationally, the MERF algorithm is implemented in the function `MERFranger` of **SAEforest**. Note that step a) is realized using package **ranger** (Wright and Ziegler 2017), while the estimation of variance components and

random effects builds on package **lme4** (Bates *et al.* 2015). The convergence of the algorithm is monitored by marginal changes of log-likelihood of the composite semi-parametric model. For further methodological details, we refer to Krennmair and Schmid (2022). The proposed estimator for model-based predictions is given by:

$$\hat{\mu}_{ij}^{\text{MERF}} = \hat{f}(\mathbf{x}_{ij}) + \hat{u}_i = \hat{f}(\mathbf{x}_{ij}) + \frac{\hat{\sigma}_u^2}{\hat{\sigma}_u^2 + \hat{\sigma}_e^2/n_i} \left(\frac{1}{n_i} \sum_{j \in s_i} (y_{ij} - \hat{f}^{\text{OOB}}(\mathbf{x}_{ij})) \right), \tag{2}$$

2.2. Flexible Domain prediction of Means under unit-level and aggregated level covariates

The predictions $\hat{\mu}_{ij}^{\text{MERF}}$ (2) depend on auxiliary unit-level information to estimate unit-level conditional means for the continuous dependent variable. In the context of SAE, however, researchers are mainly interested in estimating and mapping indicators such as area-level means or metrics measuring income deprivation and inequality (Rao and Molina 2015). For now, we will focus on the construction of area-level means depending on the availability of unit-level or aggregated auxiliary covariate information. The construction of domain-specific cumulative distribution functions (CDFs) from which non-linear indicators can be obtained will be discussed in Section 2.3.

For unit-level (i.e. $\mathbf{x}_{ij}$) supplementary data (usually census or administrative data), we calculate the mean-estimator for each area i by:

$$\hat{\mu}_i^{\text{MERF}} = \bar{\hat{f}}_i(\mathbf{x}_{ij}) + \hat{u}_i = \bar{\hat{f}}_i(\mathbf{x}_{ij}) + \frac{\hat{\sigma}_u^2}{\hat{\sigma}_u^2 + \hat{\sigma}_e^2/n_i} \left(\frac{1}{n_i} \sum_{j \in s_i} (y_{ij} - \hat{f}^{\text{OOB}}(\mathbf{x}_{ij})) \right), \tag{3}$$

$$\text{where} \quad \bar{\hat{f}}_i(\mathbf{x}_{ij}) = \frac{1}{N_i} \sum_{j \in U_i} \hat{f}(\mathbf{x}_{ij}).$$

We exploit the fact that random forest estimates of the fixed part $\hat{f}()$ express the conditional mean on unit-level and that $\hat{u}_i$ is the BLUP for the linear part of Model 1 (Krennmair and Schmid 2022). For non-sampled areas, the proposed estimator for the area-level mean reduces to the fixed part from the random forest:

$$\hat{\mu}_i = \bar{\hat{f}}(\mathbf{x}_{ij}).$$

The access to auxiliary population micro-data for covariates imposes a limitation for researchers and practitioners. As a direct consequence of non-linearity and non-continuity of random forests, we observe that $f(\bar{\mathbf{x}}_i) \neq \bar{f}_i(\mathbf{x}_{ij})$ and aggregated auxiliary information cannot directly be processed into predictions on μ_i in Equation 2. Krennmair *et al.* (2022b) solve this issue by incorporating aggregate census-level covariate information through calibration weights w_{ij}, balancing unit-level predictions from MERFs in Equation 2 in coherence with the area-wise covariate means from census data. In short, the estimator for area-level means under limited auxiliary information is given by:

$$\hat{\mu}_i^{\text{MERFagg}} = \sum_{j=1}^{n_i} \hat{w}_{ij} \left[\hat{f}(\mathbf{x}_{ij}) + \hat{u}_i \right]. \tag{4}$$

The optimal estimates from survey data for required model-components $\hat{f}$ and $\hat{u}_i$ using the MERF algorithm are similar to Equation 2. The $\mathbf{x}_{ij}$ for Estimator 4 are unit-level covariates from the survey and population-level auxiliary information is incorporated through optimal calibration weights $\hat{w}_{ij}$ maximizing the profile empirical likelihood (EL) function $\prod_{j=1}^{n_i} w_{ij}$ under the following three constraints:

- $\sum_{j=1}^{n_i} w_{ij}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_{\text{pop},i}) = 0$, monitoring the area-wise sum of distances between survey data and the population-level mean, denoted as $\bar{\mathbf{x}}_{\text{pop},i}$, for auxiliary covariates;
- $w_{ij} \geq 0$, preventing the cancellation of weights;
- $\sum_{j=1}^{n_i} w_{ij} = 1$, ensuring the normalization of weights.

The Lagrange multiplier method is suitable to find optimal weights (Owen 1990, 2001) and Krennmair *et al.* (2022b) discuss technical conditions for the feasibility of solutions in the context of SAE and propose a best practice strategy that is implemented in this package.

Irrespectively of the quality of auxiliary data sources (aggregated or unit-level), the function `SAEforest_model` provides methods to asses the uncertainty of point estimates with domain-specific MSEs. The quantification of uncertainty of domain-indicators is challenging, yet essential for the assessment of reliability of area-level estimates. Approximating the analytical MSE of domain-level indicators with estimated variance components remains challenging even in the base-scenario of LMMs with block diagonal covariance matrices (Prasad and Rao 1990; Datta and Lahiri 2000; González-Manteiga *et al.* 2008; Rao and Molina 2015). Elaborate bootstrap-schemes for the estimation of MSEs are an established alternative (Hall and Maiti 2006; González-Manteiga *et al.* 2008; Chambers and Chandra 2013) and the preferred choice under our general mixed model.

We propose a nonparametric random effect block (REB) bootstrap for estimating the MSE of area-level means of sampled and unsampled domains. The major aim is the correct reproduction of dependence-structures of data and an incorporation of uncertainty introduced through the estimation of the MERF. The nonparametric generation and resampling of random components was originally introduced by Chambers and Chandra (2013). Krennmair and Schmid (2022) postulate the importance to resample centred and scaled empirical error components by a bias-adjusted residual variance introduced by Mendez and Lohr (2011) before constructing a bootstrap population. In short, the estimator of the residual variance under the MERF from Equation 2, ($\hat{\sigma}^2_\epsilon$) is positively biased as it includes excess uncertainty concerning the estimation of function $\hat{f}$. Further methodological and performance details are found in Krennmair and Schmid (2022). For cases of existing unit-level auxiliary covariates, we imitate the sampling process by random draws from the simulated bootstrap populations. In the presence of aggregated census-level data, we generate (pseudo-) true values by resampling error components only. This idea follows methodological principles of the bootstrap for finite populations introduced by González-Manteiga *et al.* (2008). For details, model-based simulations and examples, please see Krennmair *et al.* (2022b).

2.3. Non-Linear Indicators

The analysis of distributional aspects of consumption and income (in-) equality based on statistical indicators build on a long tradition in statistical research (Atkinson 1987; Cowell 2011). In contrast to the estimation of domain-specific means, the model-based estimation of

quantiles and (non-linear) poverty indicators requires information on the area-specific CDF of y_{ij}. Chambers and Dunstan (1986)(CD) combine a model for a finite-population CDF of y_{ij} with a smearing-argument (Duan 1983) to develop a model-consistent estimator for a finite-population CDF from survey-sample data. Tzavidis *et al.* (2010) introduce the CD-method within a general unit-level framework for SAE with a focus on the estimation of SAE means and quantiles in the context of a bias-adjusted alternative to the EBLUP and outlier-robust M-quantile estimators. Extensions towards poverty (Marchetti *et al.* 2012) and inequality indicators (Marchetti and Tzavidis 2021) were investigated.

Rooted within the general unit-level framework of Tzavidis *et al.* (2010), Krennmair *et al.* (2022a) propose an estimator $F_i^*(t)$ for the area-specific CDF of y_{ij} using MERFs. Essentially, we extend the smearing method to $\hat{\mu}_{ij}$ as given by Estimator 2 using OOB-residuals $e_{ij}^* = y_{ij} - \hat{\mu}_{ij}^{\text{OOB}}$, where $\hat{\mu}_{ij}^{\text{OOB}} = \hat{f}^{\text{OOB}}(\mathbf{x}_{ij}) + \hat{u}_i$. OOB-residuals are a genuine choice for achieving more robust estimates of the CDF of MERFs, ensuring that these model-residuals e_{ij}^* mirror the estimated variance properties under Model 1. The estimator for $F_i^*(t)$ is given by:

$$\hat{F}_i^*(t) = N_i^{-1} \left[\sum_{j \in s_i} \mathbf{I}(y_{ij} \leq t) + n_i^{-1} \sum_{j \in s_i} \sum_{k \in r_i} \mathbf{I} \left(\hat{\mu}_{ik} + \underbrace{(y_{ij} - \hat{\mu}_{ij}^{\text{OOB}})}_{e_{ij}^*} \leq t \right) \right] \tag{5}$$

Smearing is computationally intensive and a Monte Carlo (MC) approximation to the area-specific CDF of y_{ij} provides an alternative. The MC-based approach draws conceptual parallels to the EBP (Molina and Rao 2010), however, lacks theoretical foundation (Marchetti *et al.* 2012). Nevertheless, the MC approximation to Equation 5 is time-efficient and given a sufficiently high number of iterations (e.g. `B_MC = 200`) no obviously identifiable differences between point estimates for various indicators are observable. **SAEforest** provides both methods and recommends the use of the theoretically supported smearing approach as default.

Estimates for indicators δ_i are calculated from $\hat{F}_i^*(t)$ using a known function $h()$. Default indicators and corresponding functions $h()$ are defined in Table 1. Package **SAEforest** includes the $(10\%, 25\%, 50\%, 75\%, 90\%)$ quantiles as default indicators characterizing the distribution of y_{ij}. We additionally include common economic measures of poverty such as the head count ratio (Hcr) and the poverty gap (Pgap) (Foster *et al.* 1984) and inequality measures such as the Gini-coefficient (Gini 1912) and the Quintile share ratio (Qsr) (Eurostat 2004). The Hcr defines the rate of being at risk of poverty, while the Pgap ratios the mean income shortfall of the poor to its respective poverty line. Both poverty indicators require a poverty threshold (z), which can be defined in absolute terms (e.g. numerical values of national poverty lines) or relative terms (e.g. defining a function depending on y_{ij}). Package **SAEforest** allows for both options. Focussing on distributional aspects, the Gini is a common measure summarizing inequality between 0 (absolute) equality and 1 (absolute inequality). While the Gini bundles information on the whole distribution, the Qsr focusses on the relation between joint income (or consumption) of the 80 and 20 percent quantile. Additionally, users can use a custom function for arbitrary statistical indicators relying on input `Y` and threshold `z`. The example in Section 4.1 will discuss customizable features in detail.

Following the work of Krennmair *et al.* (2022a), the package provides two bootstrap schemes (`nonparametric` and `wild`), each applicable for the smearing and the MC-based versions. The major difference between the two bootstrap schemes is the generation of the bootstrap

Table 1: List of predefined population indicators in **SAEforest**. F_i is the empirical distribution function in domain i.

Indicator	Definition $h()$	Range
Mean_i	$\frac{\sum_{j=1}^{N_i} y_{ij}}{N_i}$	$\mathbb{R}$
$\text{Q}_{i,q}$	$F_i^{-1}(q) = \inf\{y_{ij} \in \mathbb{R} : F_i(y_{ij}) \geq q\}$	$\mathbb{R}$
Hcr_i	$\frac{1}{N_i}\sum_{j=1}^{N_i} \mathbf{I}(y_{ij} \leq z)$	$[0,1]$
Pgap_i	$\frac{1}{N_i}\sum_{j=1}^{N_i}(\frac{z-y_{ij}}{z})\mathbf{I}(y_{ij} \leq z)$	$[0,1]$
Gini_i	$\frac{2\sum_{j=1}^{N_i} j y_{ij}}{N_i \sum_{j=1}^{N_i} y_{ij}} - \frac{N_i+1}{N_i}$	$[0,1]$
Qsr_i	$\frac{\sum_{j=1}^{N_i} \mathbf{I}(y_{ij} > Q_{i,0.8} y_{ij})}{\sum_{j=1}^{N_i} \mathbf{I}(y_{ij} > Q_{i,0.2} y_{ij})}$	$[0,1]$
custom_i	$g(y_{ij}, z)$	$\mathbb{R}$

population. The nonparametric bootstrap prepares and resamples random components for its bootstrap population in the same way as described in Section 2.2 and subsequently calculates (non-linear) indicators from the simulated data. The wild bootstrap (`wild`) exclusively relies on centred OOB-residuals and a specific matching scheme between sampled and synthetic observations building the bootstrap population. Details and performance specifics for both procedures are found in Krennmair *et al.* (2022a).

3. Data set description

Typical applications of SAE comprise survey sample data on target variable y_{ij} and predictive variables $\mathbf{x}_{ij}$. Since, existing auxiliary data sources (census or administrative/register data) do not include information on the target variable, auxiliary data sources strengthen estimates on disaggregated metrics of y_{ij} through a predictive model. As discussed in Section 2, we provide models, which handle auxiliary covariates of domain-specific individual observations or domain-level aggregates (e.g means). The exemplary datasets in this package include both types of information for illustrative purpose.

In general, this package uses data examples provided by package **emdi** (Kreutzmann *et al.* 2019). In short, the datasets comprise simulated synthetic data from the European Union Statistics on Income and Living Conditions (EU-SILC) for Austria from 2006. Although, no conclusions regarding the official levels of inequality and poverty in Austrian districts must be inferred, the simulated population micro-dataset `eusilcA_pop` exhibits realistic distributional characteristics. Originally, the `eusilcA_pop` data is a modification of the `eusilcP` data used in package **simFrame** (Alfons and Templ 2013), which reports micro-data on the nine states as lowest geographical level. Kreutzmann *et al.* (2019) use publicly available sources, such as population sizes or income rankings of districts, to assign households to one of the 94 districts. Further details on the process of data synthetization can be found in Kreutzmann *et al.* (2019).

Focussing on social and economic inequality indicators, the target variable is the equivalized household income (`eqIncome`). For the construction of `eqIncome`, total household disposable income is divided by the equivalized household size (Hagenaars *et al.* 1994). Apart from domain-level identifiers for states (`state`) and the districts (`districts`), auxiliary variables are socio-demographic characteristics, such as gender or the receipt of state benefits. An overview of model covariates is provided in Table 4 in the Appendix. The dataset `eusilcA_popAGG` comprises aggregated district-level means and is used for the illustration of Method 4 in Section 2.2. For the production of uncertainty estimates, Method 4 requires information on population-level domain sizes (Krennmair *et al.* 2022b). Synthetic population sizes for Austrian districts are provided by `popNsize`.

The unit-level sample `eusilcA_smp` is drawn by stratified random sampling from the population dataset, where districts are defined as stratas. The resulting dataset comprises 1945 observations with domain-specific sample sizes ranging from 14 ("Lienz") to 200 for the Austrian capital ("Wien"). About 25 percent of domains are not covered by the survey dataset, additionally motivating the use of model-based SAE approaches. For the illustration of the mapping function `map_indicators`, we use a shape file for the Austrian districts of class `SpatialPolygonDataFrame` (Bivand *et al.* 2013), obtainable from package **emdi** (Kreutzmann *et al.* 2019).

4. Core Functionality: The package

The statistical methods for point and MSE estimates from Section 2 are implemented in the main function `SAEforest_model`. The functionality of the package mirrors the proposed methodological flexibility of tree-based machine learning methods: firstly, depending on the available auxiliary data sources (aggregated or unit-level covariates) and the indicators of interest (means or non-linear indicators), domain-specific estimates are produced using `SAEforest_model`. Users must specify corresponding scenarios with options `meanOnly = TRUE` and/or `aggData = TRUE`. Resulting model objects can be checked by summary statistics and visual model diagnostics using the generic functions `summary` and `plot`. Function `tune_parameters` asses potential improvements of the model by tuning model hyper-parameters. Finally, function `summarize_indicators` extracts final domain-specific estimates and function `map_indicators` visualizes and maps indicators upon request. Detailed examples on the functionality of proposed methods follow in the subsections below.

Generic functions of the package rely on S3 objects of class `SAEforest` (Chambers and Hastie 1992). The main function `SAEforest_model` wraps the basis function `MERFranger`. The implementation of the MERF algorithm is done by a composite model of a random forest fitted by the package **ranger** (Wright and Ziegler 2017) and random intercepts and corresponding variance components obtained by the package **lme4** (Bates *et al.* 2015). Thus, users benefit from the full functionality of both package environments including generic functions of respective classes `ranger` and `merMod`. Moreover, users can directly pass hyper-parameters to the function `ranger` or choose alternative splitrules for trees. Although the basis function `MERFranger` is only addressed through wrapper functions for the average package user, we additionally provide the function to enable unit-level predictions under more advanced correlation and dependency structures. By this, we aim to facilitate further research and development using MERFs for SAE. For details see `help(MERFranger)` or the methodology discussed in Krennmair and Schmid (2022).

Table 2: Details on inputs for main function `SAEforest_model`.

Input	Description	`meanOnly = T`	`meanOnly = F`
`Y`	Continuous target variable.	✓	✓
`X`	Matrix or data.frame of predictive covariates.	✓	✓
`dName`	Character of domain identifier.	✓	✓
`smp_data`	data.frame of survey sample data.	✓	✓
`pop_data`	data.frame of population-level covariates `X`.	✓	✓
`MSE`	Specification of uncertainty estimates. Currently available options are: `none`, `nonparametric` and for `meanOnly = F` additionally `wild`.	✓	✓
`importance`	Variable importance processed by **ranger**. Must be one of the following: "impurity", "impurity_corrected" or "permutation".	✓	✓
`initialRandomEffects`	Initial estimate of random effects. Defaults to 0.	✓	✓
`ErrorTolerance`	Value monitoring MERF algorithm's convergence. Defaults to 1e-04.	✓	✓
`MaxIterations`	Value specifying maximal amount of iterations for MERF algorithm. Defaults to 25.	✓	✓
`B`	Bootstrap replications for MSE estimation. Defaults to 100.	✓	✓
`B_adj`	Bootstrap replications for adjustment of residual variance. Defaults to 100.	✓	✓
`na.rm`	Logical. Whether missing values should be removed.	✓	✓
`...`	Additional parameters passed to `ranger`. Most important parameters are `mtry` (number of variables to possibly split at in each node), or `num.trees` (number of trees).	✓	✓
`aggData`	Logical. Whether aggregated covariate information is used.	✓	
`popnsize`	Information of population size of domains. Only needed if `aggData = TRUE` and MSE is requested.	✓	
`OOsample_obs`	Out-of-sample observations taken from the closest area. Only needed if `aggData = TRUE` with default set to 25.	✓	
`ADDsamp_obs`	Out-of-sample observations taken from the closest area if first iteration for the calculation of calibration weights fails. Only needed if `aggData = TRUE` with default set to 0.	✓	
`w_min`	Minimal number of covariates from which informative weights are calculated. Only needed if `aggData = TRUE`. Defaults to 3.	✓	
threshold	Set a custom threshold for indicators. The threshold can be a known numeric value or function of `Y`. Defaults to `NULL` resulting in 60% of median of `Y`.		✓
`custom_indicator`	A list of additional functions containing the indicators to be calculated. These functions must only depend on the target variable `Y` and the `threshold`. Defaults to `NULL`.		✓
`smearing`	Logical input indicating whether a smearing based approach or a MC-based version for point estimates is obtained. Defaults to `TRUE`.		✓
`B_MC`	Bootstrap populations to be generated for the MC version. Defaults to 100.		✓

4.1. Estimation of Domain-level indicators

The following examples use the synthetic Austrian EU-SILC data discussed in Section 3. Firstly, we focus on the most ideal case including unit-level survey sample data and access to unit-level covariate data from a census to estimate the area-level mean. The information on the equivalized income is only measured in the survey data, but covariates `X_covar` are measured on survey and census level.

```
R> #Loading data
data("eusilcA_pop")
data("eusilcA_smp")

income <- eusilcA_smp$eqIncome
X_covar <- eusilcA_smp[,-c(1,16,17,18)]
```

This data-scenario corresponds to Method 2. As we are only interested in the area-level mean, we specify option `meanOnly = TRUE` and define target variable `Y` and corresponding covariates in the sample `X = X_covar`. Input values for covariates `X` must be predictors only and we remove columns containing area-level codes and the target variable for the assignment `X = X_covar`. We explicitly denote `dName` to indicate separate areas for random intercepts and assign the survey dataset `smp_data` and the dataset comprising population-level information `pop_data`. For the current example, point estimates are sufficient and we specify `MSE = "none"`. As discussed in Section 2, the current implementation has an option to produce uncertainty estimates of area-level means with option `nonparametric` referring to the MSE procedures discussed in Krennmair and Schmid (2022). Dealing with unit-level population data, we keep the default of `aggData = FALSE`. Note that this option must be replaced by `TRUE` in the case of limited covariate information.

```
R> MERFmodel1 <- SAEforest_model(Y = income, X = X_covar, dName = "district",
+         smp_data = eusilcA_smp, pop_data= eusilcA_pop, MSE = "none",
+         meanOnly = TRUE, aggData = FALSE)
```

Before we discuss model components and respective results, we focus on inputs for estimating more complex area-level indicators, such as quantiles or inequality indicators. Function `SAEforest_model` with option `meanOnly = FALSE` corresponds to the methodology explained in Section 2.3 and allows for further scenario-dependent inputs. The option `smearing` determines whether we want to construct a full smearing CDF or choose a Monte-Carlo simulated marginal distribution of y_{ij}. Depending on computational feasibility, we advice the general use of smearing-based estimates due to its theoretical corroboration compared to the MC version. For MSE estimates, we have options `none, wild` or `nonparametric` as described in Krennmair *et al.* (2022a). The default indicators returned by `SAEforest_model` with option `meanOnly = FALSE` include the mean, median, quantiles (10%, 25%, 75% and 90%), Hcr, Pgap, Gini, and the Qsr. Users specify a custom threshold by passing a known numeric value or a function of `Y`. If the threshold is `NULL`, 60 % of the median of `Y` is taken as threshold. Additionally, `SAEforest_model` allows for custom indicators. In the following example we constructed a new indicator, defining area-level maximum incomes. The input for `custom_indicator` must be a list of functions depending only on inputs `Y` and `threshold`.

```
R> MERFmodel2 <- SAEforest_model(Y = income, X = X_covar, dName = "district",
+        smp_data = eusilcA_smp, pop_data = eusilcA_pop, smearing = FALSE,
+        meanOnly = FALSE, MSE = "nonparametric", B = 100, mtry=5,
+        num.trees = 500, threshold = function(Y){0.5 * median(Y)},
+        custom_indicator = list(my_max = function(Y, threshold){max(Y)}))
```

Function `SAEforest_model` allows to pass arguments directly to the function `ranger` using the generic three-dotted option (`...`). Most important inputs to specify a random forest are the number of randomized variables for each node split decision (`mtry`) or the overall number of trees (`num.trees`). Any option available for `ranger` (such as alternative split criteria) can be directly passed to the function. For details see Wright and Ziegler (2017) and our discussion on tuning parameters in Section 4.3. Table 2 in the Appendix summarizes and explains the inputs for `SAEforest_model`.

Function `SAEforest_model` produces an output object of class `SAEforest`, which always includes at least four elements: (i) point estimates of specified regionally disaggregated indicators; (ii) a `MERFmodel` object including information on the model fit for fixed effects and random effects; (iii) MSE estimates if requested and `NULL` otherwise; (iv) the value of the adjusted standard deviation used in the MSE bootstrap or `NULL` otherwise. In the case of domain-level means under aggregated covariate information (`aggData = TRUE`), the object additionally includes an element, capturing the number of variables used in the weighting process from aggregated covariate information. Table 3 summarizes and explains individual components of `SAEforest` objects. Several generic functions are applicable and we firstly focus on model diagnostics produced by `summary` and `plot` in the following section.

4.2. Summary function and diagnostic plots

Function `summary` is an important generic function to obtain essential information on a fitted model object. An exemplary output from `summary` of a fitted model object of class `SAEforest` is given below:

```
R> summary(MERFmodel1)

Call:
SAEforest_model(Y = income, X = X_covar, dName = "district",
smp_data = eusilcA_smp, pop_data = eusilcA_pop, MSE = "none",
aggData = FALSE, importance = "impurity")

Domains
In-sample Out-of-sample Total
70            24    94
Totals:
Units in sample: 1945
Units in population: 25000
                   Min. 1st Qu. Median      Mean 3rd Qu. Max.
Sample_domains       14    17.0   22.5  27.78571   29.00  200
Population_domains    5   126.5  181.5 265.95745  265.75 5857

Random forest component:
Type:                             Regression
```

```
Number of trees:                          500
Number of independent variables:          14
Mtry:                                     3
Minimal node size:                        5
Variable importance mode:                 impurity
Splitrule:                                variance
Rsquared (OOB):                           0.62036

Structural component of random effects:
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: Target ~ -1 + (1 | district)
Data: data
Offset: forest_preds
        AIC      BIC   logLik deviance df.resid
39225.2  39236.3 -19610.6  39221.2     1943
Scaled residuals:
        Min      1Q  Median      3Q     Max
-3.1425 -0.5243 -0.0577  0.4433 11.6832

Random effects:
Groups   Name        Variance Std.Dev.
district (Intercept) 12132734 3483
Residual             30771664 5547
Number of obs: 1945, groups:  district, 70
ICC:  0.2827853

Convergence of MERF algorithm:
Convergence achieved after 8 iterations.
A maximum of 25 iterations used and tolerance set to: 1e-04
Monitored Log-Likelihood:
0 -19546.21 -19572.14 -19588.23 -19592.72 -19604.67 -19599.86 -19609.86 -19610.59
```

The `summary` output provides preliminary insights into SAE characteristics such as domain-specific sample sizes, information on sampled and unsampled domains and the total amount of observations. In this example, we face domain-specific sample sizes with a median of 22.5 households, motivating the use of model-based SAE. Moreover, for 24 out of 94 domains, no direct estimates are obtainable. The second essential insight from the output reports model-specific metrics. Starting with the random forest part, we find values such as tuning parameters and R^2 on fixed effects. The R^2 of around 0.62 substantiates the model's predictive capability. The information on the fit of the structural component of our MERF model describes the variance for the area-level random intercept and the individual residuals as well as the intra-class-correlation coefficient (ICC). The ICC of about 0.29 justifies the need for an area-level random effect. The last block of our summary-output highlights convergence properties of the MERF algorithm, such as the amount of needed iterations and the monitored level of likelihood.

As discussed in Section 2, the MERF model is a composite model of a random forest and a structural model. This structure is not only mirrored in the output of `summary`, but also within each fitted model object. Thus, users can address elements directly from the fitted model object and use the generic functions from **ranger** (Wright and Ziegler 2017) and **lme4** (Bates *et al.* 2015) respectively. Corresponding objects are stored in `ForestModel` and `Effectmodel`.

Table 3: Details on an object of class `SAEforest`.

Object of class `SAEforest`

Component	Short description
`MERFmodel`	The `MERFmodel` object comprises information on the model fit, details on the algorithm and variance components.
`Indicators`	Element comprising area-level identifiers and estimates.
`MSE_estimates`	Includes area-level identifiers and uncertainty estimates if requested and `NULL` otherwise.
`AdjustedSD`	If MSE results are requested residual variance proposed by Mendez and Lohr (2011) is reported and `NULL` otherwise.
`NrCovar`	Exists only if `meanOnly = TRUE`. Set to `NULL` except `aggData = TRUE` for which it includes a list of variable names of covariates used for the calculation of calibration weights. See Krennmair *et al.* (2022b) for details.

Details on `MERFmodel`

Component	Short description
`Forest`	Random forest of class `ranger` modelling fixed effects of the model.
`EffectModel`	Model of random effects of class `merMod` capturing structural components of MERFs.
`RandomEffects`	List element containing the values of random intercepts from `EffectModel`.
`RanEffSD`	Standard deviation of random intercepts.
`ErrorSD`	Standard deviation of unit-level errors.
`VarianceCovariance`	VarCorr matrix from `EffectModel`.
`LogLik`	Vector of log-likelihood of the MERF algorithm.
`IterationsUsed`	Iterations used until convergence of the MERF algorithm is reached.
`OOBresiduals`	Vector of OOB-residuals.
`Random`	Character specifying the random intercept in the random effects model.
`ErrorTolerance`	Value monitoring the MERF algorithm's convergence.
`initialRandomEffects`	Vector of initial specification of random effects.
`MaxIterations`	Value specifying the maximal amount of iterations for the MERF algorithm.
`call`	The summarized function call for the object.
`data_specs`	Data characteristics such as domain-specific sample sizes or number of out-of-sample areas.
`data`	The survey sample data.

Especially for objects of class `merMod`, there exist advantageous generics to extract model components. The following functions are directly applicable: `getData, VarCorr, sigma, residuals, ranef, fixef`. For instance, `ranef` obtains random effects and `VarCorr` directly accesses the variance-covariance matrix:

```
R> ranef(MERFmodel1)
R> VarCorr(MERFmodel1)
```

An major complement of summaries and descriptive statistics are diagnostic plots. The generic `plot` function in the package **SAEforest**, produces random forest specific diagnostic tools, like variable importance plots (vip) and partial dependence plots (dpd). A variable importance plot ranks the importance of predictive covariates in the estimation process of the model. Figure 1 reports the mean decrease in impurity (variance) calculated for each predictor as the sum over the number of splits across all trees that include the predictor. For the variable importance plot, arguments are passed internally to the function `vip` (Greenwell *et al.* 2020).

The additional partial dependence plot (pdp) depicts the estimated marginal effect for a given number of influential covariates on the target variable. The pdp plot is produced using the package **pdp** (Greenwell 2017).

The function `plot` offers several options of customization: most importantly, users can decide whether they want both plots or just the vip plot by specifying `pdp_plot = FALSE`. The plotting engine is **ggplot2** (Wickham 2011) and several graphical arguments, such as colours or themes can be directly specified. Additionally, the generic function `plot` provides the possibility to export a list including requested plots, which allows for modifications based on the additivity of layers for `ggplot`-objects.

```
R> plot(MERFmodel1, num_features = 6, col = "darkgreen",
+   fill = "darkgreen", alpha = 0.8,horizontal = TRUE,
+   gg_theme = theme_minimal(), lsize = 1.5, lty = "solid",
+   grid_row = 2, out_list = FALSE,
+   pdp_plot = TRUE)
```

Figure 1 shows the first plot on the fitted object `MERFmodel1`. Most influential variables in the estimation process of fixed effects are net cash income (cash), age-related benefits (age_ben), whether a person is self-employed (self_empl), obtains income from rent (rent), profits from capital investment (cap_inv) or receives family related allowances (fam_allow). Importance plots do not allow for inferences on predictive relations between our target variable of equivalized household income and the covariates. A scrutiny of the pdp plot in Figure 1 highlights potential non-linear relations for instance for cash, where the average marginal effect flattens with cash values over 50000. A similar pattern is observable for self-employed income. Another non-linear peculiarity is the discontinuity for fam_allow around 20000.

4.3. Model-tuning and important parameters

Random forests are nonparametric procedures, which performance depend on tuning parameters. Function `tune_parameters` assists in fine-tuning of parameters for the implemented MERF method. Essentially, this function is a modified wrapper for `train` from the package **caret** (Kuhn 2022), treating MERFs as a custom method. Tuning can be performed on the following four parameters: `num.trees` (the number of trees for a forest), `mtry` (number of variables as split candidates at in each node), `min.node.size` (minimal individual node size) and `splitrule` (general splitting rule of individual trees).

Necessary inputs for `tune_parameters` are control parameters for function `train` from the package **caret** (Kuhn 2022), such as the type of cross validation (`method = "repeatedcv"`), the numner of folds (`number = 5`), and corresponding repetitions (`repeats = 1`). Moreover, the input of potential tuning parameters must be defined by a grid of parametrization candidates. Data-specific inputs, such as the defined target variable, covariates and the survey dataset resemble the input for the wrapper function `SAEforest_model` discussed in Section 4.1.

```
R> fitControl <- caret::trainControl(method = "repeatedcv", number = 5,
+        repeats = 3)

# Define a tuning-grid
```

```
R> merfGrid <- expand.grid(num.trees = 500, mtry = c(3,7,9),
+        min.node.size = c(10), splitrule = "variance")

R> tune_parameters(Y = income, X = X_covar, data = eusilcA_smp, dName =
+        "district", trControl = fitControl, tuneGrid = merfGrid. plot_res =
+        FALSE)

1945 samples
15 predictor
No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 1557, 1557, 1556, 1556, 1554, 1556, ...
Resampling results across tuning parameters:

mtry  RMSE      Rsquared   MAE
3     5769.200  0.7126250  3832.716
7     5496.742  0.7333739  3565.051
9     5514.225  0.7306313  3556.285

Tuning parameter 'num.trees' was held constant at a value of 500
Tuning parameter 'min.node.size' was held constant at a value of 10
Tuning parameter 'splitrule' was held constant at a value of variance
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were num.trees = 500, mtry = 7,
min.node.size = 10 and splitrule = variance.
```

The output of `tune_parameters` coincides with output from `train` in the package **caret** (Kuhn 2022). Users can specify whether the summarized information should be accompanied by visualized diagnostics based on **ggplot2** (Wickham 2011). Most important metrics for fine-tuning decisions are cross-validated results of the RMSE, MAE or the conditional R^2. Following the default specifcation using RMSE as most important criterion for regression, the optimal tuning parameter on the number of randomized split candidates at each node (`mtry`) is 7.

4.4. Mapping of results and presentation of indicators

The previous functions focussed on the estimation of indicators and the diagnosis of model quality as well as improvements using optimized tuning parameters. Equally important to the package **SAEforest**, however, is the clear and intuitive presentation of results. Function `summarize_indicators` reports point and MSE estimates as well as calculated coefficients of variation (CV) from a fitted SAEforest object. The CV is an established indicator for national statistical offices to assess associated uncertainty and quality of estimates and is defined as:

$$\mathrm{CV}(\hat{\delta}_i) = \frac{\sqrt{\widehat{\mathrm{MSE}}(\hat{\delta}_i)}}{\hat{\delta}_i}.$$

Users can optionally include a character vector specifying indicators to be reported, referring to all calculated indicators (`all`); each default indicator's name (`Mean, Quant10,`

`Quant25, Median, Quant75, Quant90, Gini, Hcr, Pgap, Qsr` or the function name/s of `custom_indicator`/s) or a vector of multiple indicator names. If the object is estimated by `SAEforest_model` under option `meanOnly = TRUE`, all indicator arguments are ignored and only the `Mean` is returned.

The output object of class `summarize_indicators.SAEforest` allows for generic functions for `data.frames` such as `head, tail, as.matrix, as.data.frame` and `subset`. In the following example, we provide a summary on the Mean, Gini and our customized indicator, identifying the area-level maximum income and respective CVs.

```
R> head(summarize_indicators(MERFmodel2, MSE = FALSE, CV =TRUE, indicator =
+        c("Mean", "Gini", "my_max")))
```

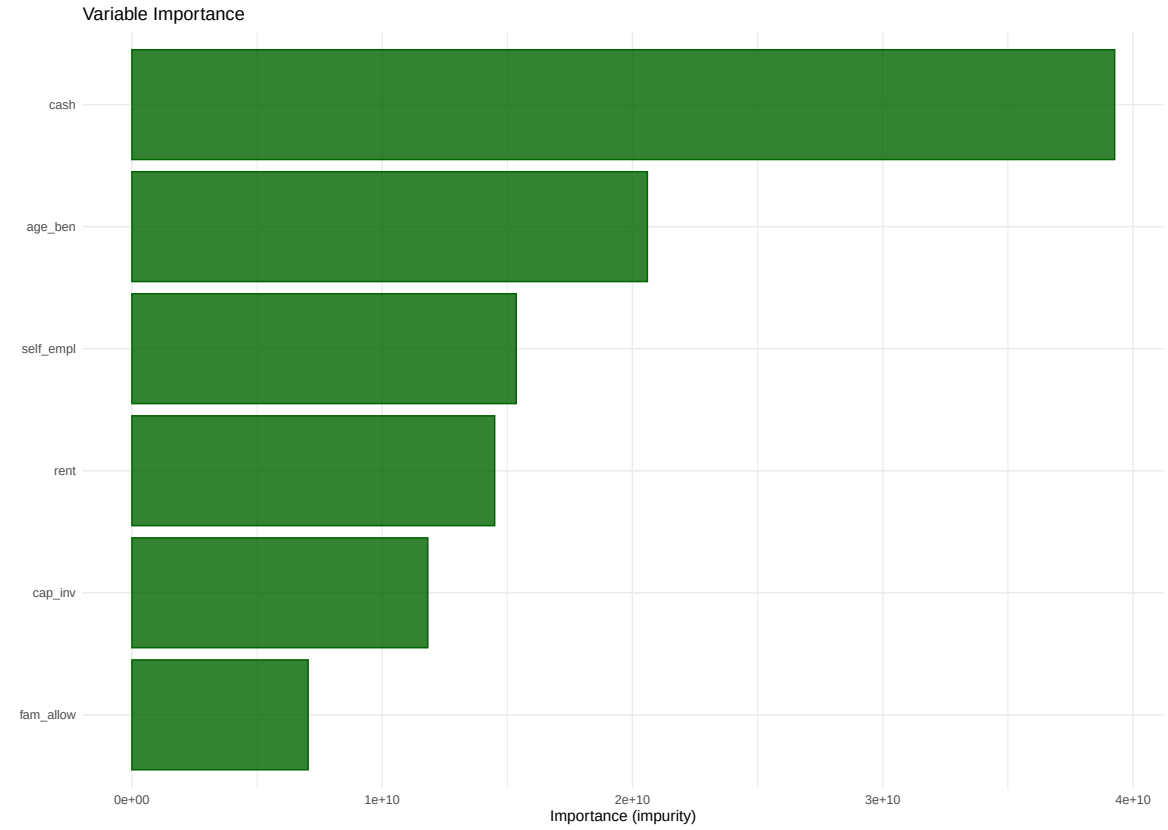
```
       district     Mean    Mean_CV      Gini    Gini_CV   my_max my_max_CV
     Amstetten 14249.76 0.05492730 0.2476006 0.07018764 56579.45 0.3340302
         Baden 22648.20 0.02940555 0.1767620 0.06594121 69621.40 0.2957758
       Bludenz 12411.98 0.09589232 0.2772565 0.09071811 45723.53 0.4555028
 Braunau am Inn 12046.12 0.06895787 0.2770546 0.06957077 53530.96 0.3863748
       Bregenz 32554.19 0.03074062 0.1559062 0.11456712 77513.46 0.2358645
```

Revealing spatial patterns of inequality and poverty, necessitates the presentation of results with maps. Function `map_indicators` visualizes estimates from a fitted model object of class `SAEforest` on a specified map. Essential inputs for `map_indicators` are the fitted model object, the `map_object` of class `SpatialPolygonsDataFrame` (Bivand *et al.* 2013) and the domain-level identifier from the `map_object`. In case of differing area-level identifiers between the model object of class `SAEforest` and the `SpatialPolygonsDataFrame` object, `map_tab` provides a possibility to enter a `data.frame` linking areas effectively. Comparably to `summarize_indicators`, users can choose specific indicators and whether MSE or CV results should be mapped. For further details we refer to the help page of function `map_indicators` or Bivand *et al.* (2013) for a concise overview on the handling of spatial data in R.

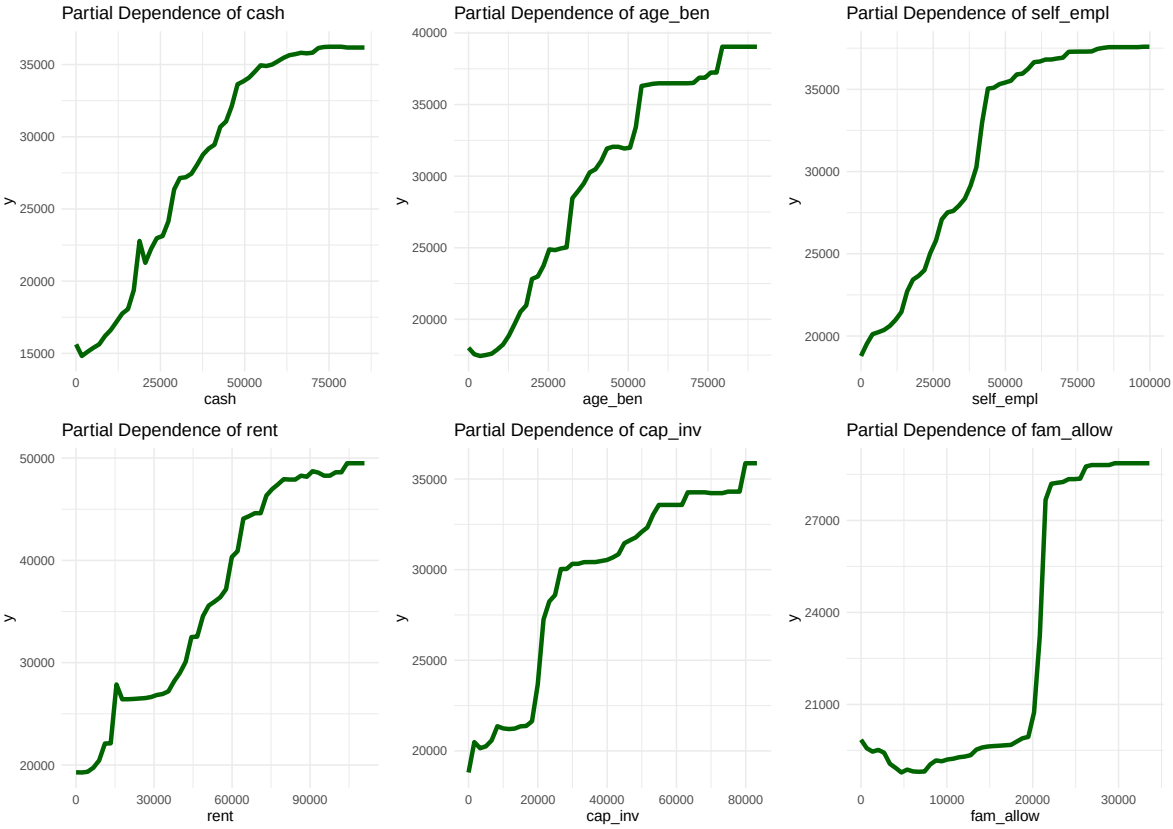
Emphasis lies on the flexibility to customize and adapt produced maps. Users can choose colours and themes of the plot based on the plot engine **ggplot2** (Wickham 2011) and export a list of `ggplot`-elements for further customization if `return_plot = TRUE`. Additionally, users can export a fortified data frame comprising map data and the chosen indicators to produce customized maps using preferred alternative mapping and plotting procedures.

Continuing on our example, we load the shape file on 94 Austrian districts and map results from the fitted object `MERFmodel2` for the Mean and the Gini. The map of mean equivalized household income shown by Figure 2 indicates differences across Austrian districts, where "Mödling" reports the highest value, which is in accordance to official statistics of income in Austria (Statistik Austria 2021). Also inequality measured by the Gini is not equally distributed ranging from 0.141 ("Urfahr-Umgebung") to a maximum of 0.301 ("Zell am See"). The majority of CVs for domain-specific values of mean and Gini estimates lies below the 20% threshold, which meets the reliability criterion of Eurostat (2019).

```
R> data("shape_Aut")
R> map_indicators(object = MERFmodel2, MSE = FALSE, CV = TRUE,
+        map_obj = shape_Aut, indicator = c("Mean", "Gini"),
+        map_dom_id = "PB")
```



(a) Variable importance plot (vip).



(b) Partial dependence plots (pdp) for 6 most influential variables.

Figure 1: Output from function `plot`.

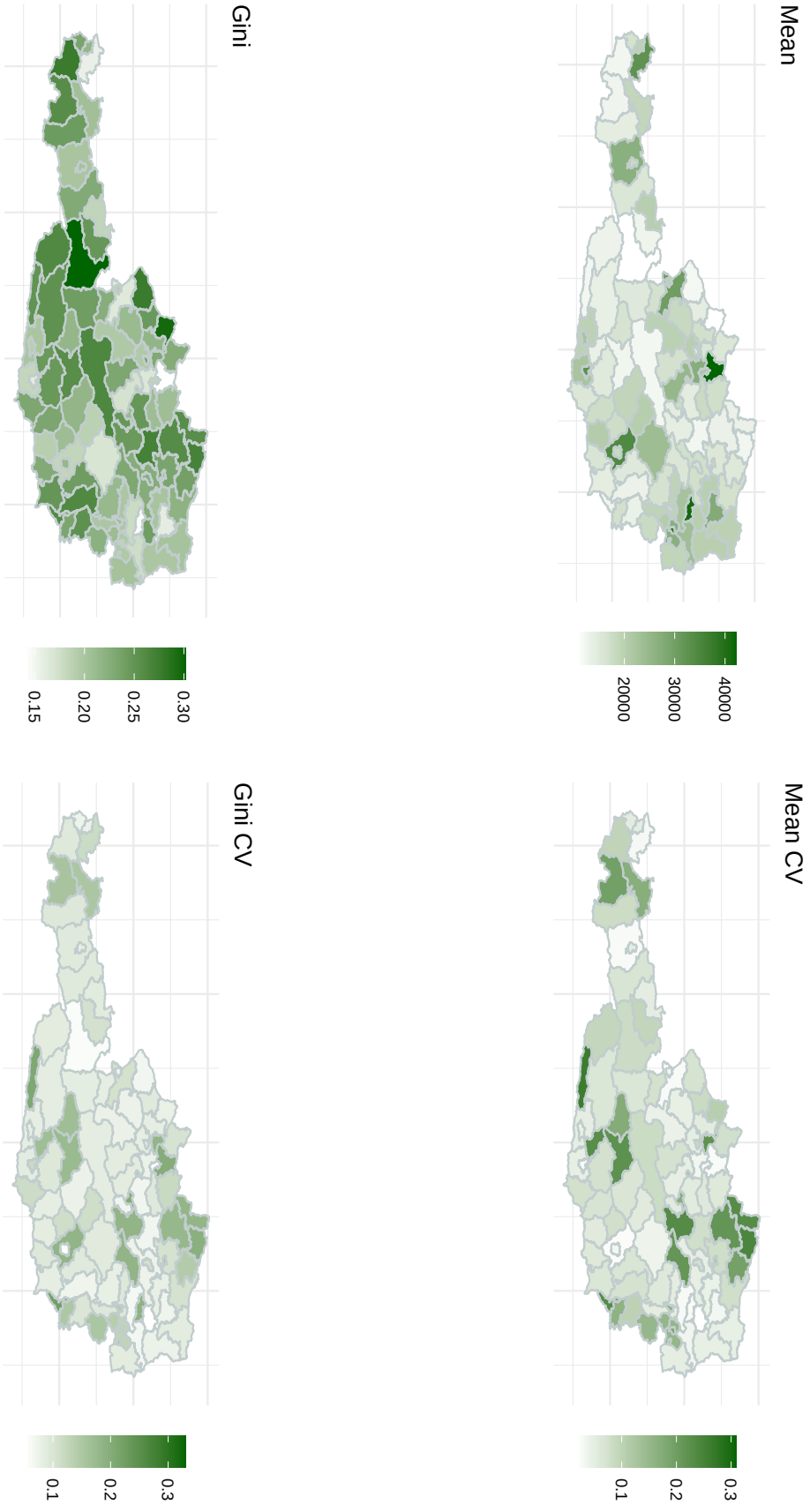


Figure 2: District-level estimates for Mean and Gini-coefficient including CVs mapped on Austrian territory. Resulting plots are produced from function `map_indicators`.

5. Discussion and Outlook

This package aims to bridge concepts of machine learning methods and 'traditional' perspectives of SAE. From a methodological perspective, the estimation of point and uncertainty estimates for domain-level indicators is performed under unit-level and aggregated covariates and dependency structures of observations are modelled using a semi-parametric framework of MERFs. Benefits of random forests align with the proclaimed focus on robustification of SAE models against model-failure (e.g. providing insurances against model-misspecification, valid variable selection including complex and potentially non-linear interactions between covariates and the effective handling of outliers) (Jiang and Rao 2020). Moreover, random forests handle high-dimensional ($p > n$) datasets enabling additional perspectives on research concerning Big Data sources (Marchetti *et al.* 2015; Schmid *et al.* 2017).

The package **SAEforest** adds valuable insights and advantages to the existing repertoire of SAE methods and yet remains within the methodological tradition of SAE. This includes efforts to provide solutions within the context of domain-level indicators, dependent data structures and in the broader context of survey methodology. We acknowledge that compared to LMMs, benefits of flexibility serve at cost of explainability and attribution, however, this is mitigated by the package's emphasis on informative summary diagnostics and plots (e.g. vip and pdp plots). In addition, the package functionality is characterized by an intuitive workflow and functions to facilitate the visualization of geospatial data. Future versions of the package will ideally include a generalization of our framework to binary and count data. Additionally, the extension towards other machine learning approaches, such as Support Vector Machines, Gradient Boosting and Bayesian Additive Regression Trees is a thought-provoking goal for further research.

6. Appendix

Table 4: Details on the predictive covariates in the survey and population-level datasets.

Variable	Explanation
`eqIncome`	numeric; a simplified version of the equivalized household income. Only available in the survey sample.
`eqsize`	numeric; the equivalized household size according to the modified OECD scale.
`gender`	factor; the person's gender (levels: male and female).
`cash`	numeric; employee cash or near cash income (net).
`self_empl`	numeric; cash benefits or losses from self-employment (net).
`unempl_ben`	numeric; unemployment benefits (net).
`age_ben`	numeric; old-age benefits (net).
`surv_ben`	numeric; survivor's benefits (net).
`sick_ben`	numeric; sickness benefits (net).
`dis_ben`	numeric; disability benefits (net).
`rent`	numeric; income from rental of a property or land (net).
`fam_allow`	numeric; family/children related allowances (net).
`house_allow`	numeric; housing allowances (net).
`cap_inv`	numeric; interest, dividends, profit from capital investments in unincorporated business (net).
`tax_adj`	numeric; repayments/receipts for tax adjustment (net).
`state`	factor; state (nine levels).
`district`	factor; districts (94 levels).
`weight`	numeric; constant weight.

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.

Atkinson AB (1987). "On the measurement of poverty." *Econometrica*, **55**(4), 749–764.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software*, **67**(1), 1–48.

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

Biau G, Scornet E (2016). "A Random Forest Guided Tour." *Test*, **25**(2), 197–227.

Bivand RS, Pebesma E, Gomez-Rubio V (2013). *Applied spatial data analysis with R, Second edition*. Springer, NY. URL https://asdar-book.org/.

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

Breidenbach J (2018). ***JoSAE**: Unit-Level and Area-Level Small Area Estimation*. R package version 0.3.0, URL https://CRAN.R-project.org/package=JoSAE.

Breiman L (2001). "Random Forests." *Machine Learning*, **45**(1), 5–32.

Capitaine L (2020). ***LongituRF**: Random Forests for Longitudinal Data*. R package version 0.9, URL https://CRAN.R-project.org/package=LongituRF.

Chambers J, Hastie T (1992). *Statistical Models in S*. Chapman & Hall, London.

Chambers R, Chandra H (2013). "A Random Effect Block Bootstrap for Clustered Data." *Journal of Computational and Graphical Statistics*, **22**(2), 452–470.

Chambers R, Dunstan R (1986). "Estimating Distribution Functions from Survey Data." *Biometrika*, **73**(3), 597–604.

Chambers R, Tzavidis N (2006). "M-quantile models for small area estimation." *Biometrika*, **93**(2), 255–268.

Cowell FA (2011). *Measuring inequality*. 3rd edition. Oxford University Press, New York.

Datta GS, Lahiri P (2000). "A Unified Measure of Uncertainty of Estimated Best Linear Unbiased Predictors in Small Area Estimation Problems." *Statistica Sinica*, **10**(2), 613–627.

Diallo MS, Rao JNK (2018). "Small Area Estimation of Complex Parameters Under Unit-Level Models with Skew-Normal Errors." *Scandinavian Journal of Statistics*, **45**(4), 1092–1116.

Duan N (1983). "Smearing Estimate: A Nonparametric Retransformation Method." *Journal of the American Statistical Association*, **78**(383), 605–610.

Efron B (2020). "Prediction, Estimation, and Attribution." *Journal of the American Statistical Association*, **115**(530), 636–655.

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

Eurostat (2019). "DataCollection: precision level DCF. Eurostat, Luxembourg." (Available from https://datacollection.jrc.ec.europa.eu/wordef/precision-level-dcf).

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

Ghosh M, Myung J, Moura F (2016). ***robustsae**: Robust Bayesian Small Area Estimation*. R package version 0.1.0, URL https://CRAN.R-project.org/package=robustsae.

Gini C (1912). *Variabilit'a e Mutabilit'a. Contributo allo Studio delle Distribuzioni e delle Relazioni Statistiche*. P. Cuppini, Bologna.

González-Manteiga W, Lombardía MJ, Molina I, Morales D, Santamaría L (2008). "Bootstrap Mean Squared Error of a Small-Area EBLUP." *Journal of Statistical Computation and Simulation*, **78**(5), 443–462.

Graf M, Marín JM, Molina I (2019). "A Generalized Mixed Model for Skewed Distributions Applied to Small Area Estimation." *Test*, **28**(2), 565–597.

Greenwell BM (2017). "**pdp**: An R Package for Constructing Partial Dependence Plots." *The R Journal*, **9**(1), 421–436.

Greenwell BM, Boehmke B, Gray B (2020). "Variable Importance Plots—An Introduction to the vip Package." *The R Journal*, **12**(1), 343–366.

Hagenaars AJ, De Vos K, Asghar Zaidi M, *et al.* (1994). "Poverty statistics in the late 1980s: Research based on micro-data." *Technical report*, Office for Official Publications of the European, Luxembourg (Luxembourg).

Hajjem A, Bellavance F, Larocque D (2014). "Mixed-Effects Random Forest for Clustered Data." *Journal of Statistical Computation and Simulation*, **84**(6), 1313–1328.

Hall P, Maiti T (2006). "On Parametric Bootstrap Methods for Small Area Prediction." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68**(2), 221–238.

Hastie T, Tibshirani R, Friedman J (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd edition. Springer Science & Business Media, New York.

Jiang J, Rao JS (2020). "Robust Small Area Estimation: An Overview." *Annual Review of Statistics and its Application*, **7**(1), 337–360.

Krennmair P, Schmid T (2022). "Flexible domain prediction using mixed effects random forests." *Journal of Royal Statistical Society: Series C (Applied Statistics)*. Forthcoming.

Krennmair P, Schmid T, Tzavidis N (2022a). "The Estimation of Poverty Indicators Using Mixed Effects Random Forests." Workig Paper.

Krennmair P, Würz N, Schmid T (2022b). "Analysing Opportunity Cost of Care Work using Mixed Effects Random Forests under Aggregated Census Data." URL https://arxiv.org/abs/2204.10736.

Kreutzmann AK, Pannier S, Rojas-Perilla N, Schmid T, Templ M, Tzavidis N (2019). "The R Package emdi for Estimating and Mapping Regionally Disaggregated Indicators." *Journal of Statistical Software*, **91**(7).

Kuhn M (2022). ***caret**: Classification and Regression Training*. R package version 6.0-92, URL https://CRAN.R-project.org/package=caret.

Li ZR, Martin BD, Hsiao Y, Godwin J, Paige J, Wakefield J, Clark SJ, Fuglstad GA, Riebler A (2021). *SUMMER: Small-Area-Estimation Unit/Area Models and Methods for Estimation in R*. R package version 1.2.0, URL https://CRAN.R-project.org/package=SUMMER.

Marchetti S, Giusti C, Pratesi M, Salvati N, Giannotti F, Pedreschi D, Rinzivillo S, Pappalardo L, Gabrielli L (2015). "Small Area Model-Based Estimators Using Big Data Sources." *Journal of Official Statistics*, **31**(2), 263–281.

Marchetti S, Tzavidis N (2021). "Robust estimation of the Theil index and the Gini coefficient for small areas." *Journal of Official Statistics*, **37**(4), 955–979.

Marchetti S, Tzavidis N, Pratesi M (2012). "Non-parametric bootstrap mean squared error estimation for M-quantile estimators of small area averages, quantiles and poverty indicators." *Computational Statistics & Data Analysis*, **56**(10), 2889–2902.

Mendez G, Lohr S (2011). "Estimating Residual Variance in Random Forest Regression." *Computational Statistics & Data Analysis*, **55**(11), 2937–2950.

Molina I, Marhuenda Y (2015). "sae: An R Package for Small Area Estimation." *The R Journal*, **7**(1), 81–98.

Molina I, Martín N (2018). "Empirical best prediction under a nested error model with log transformation." *The Annals of Statistics*, **46**(5), 1961–1993.

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

Neufeld A, Heggeseth B (2019). ***splinetree**: Longitudinal Regression Trees and Forests*. R package version 0.2.0, URL https://CRAN.R-project.org/package=splinetree.

Opsomer JD, Claeskens G, Ranalli MG, Kauermann G, Breidt F (2008). "Non-Parametric Small Area Estimation Using Penalized Spline Regression." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **70**(1), 265–286.

Owen A (1990). "Empirical likelihood ratio confidence regions." *The Annals of Statistics*, **18**(1), 90–120.

Owen A (2001). *Empirical likelihood*. Chapman and Hall, New York.

Pfeffermann D (2013). "New important developments in small area estimation." *Statistical Science*, **28**(1), 40–68.

Prasad NGN, Rao JNK (1990). "The Estimation of the Mean Squared Error of Small-Area Estimators." *Journal of the American Statistical Association*, **85**(409), 163–171.

Rao JNK, Molina I (2015). *Small Area Estimation*. 2.nd edition. Wiley series in survey methodology, New Jersey: Wiley.

Rojas-Perilla N, Pannier S, Schmid T, Tzavidis N (2020). "Data-Driven Transformations in Small Area Estimation." *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **183**(1), 121–148.

Saha A, Basu S, Datta A (2021). ***RandomForestsGLS**: Random Forests for Dependent Data*. R package version 0.1.3, URL https://CRAN.R-project.org/package=RandomForestsGLS.

Schmid T, Bruckschen F, Salvati N, Zbiranski T (2017). "Constructing sociodemographic indicators for national statistical institutes by using mobile phone data: estimating literacy rates in Senegal." *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **180**(4), 1163–1190.

Schoch T (2014). ***rsae**: Robust Small Area Estimation*. R package version 0.1-5, URL https://CRAN.R-project.org/package=rsae.

Sinha SK, Rao JNK (2009). "Robust small area estimation." *Canadian Journal of Statistics*, **37**(3), 381–399.

Statistik Austria (2021). "Statistik der Lohnsteuer 2020." *Technical report*, Statistik Austria, Wien.

Sugasawa S, Kubokawa T (2019). "Adaptively Transformed Mixed-Model Prediction of General Finite-Population Parameters." *Scandinavian Journal of Statistics*, **46**(4), 1025–1046.

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

Tzavidis N, Zhang LC, Luna A, Schmid T, Rojas-Perilla N (2018). "From Start to Finish: A Framework for the Production of Small Area Official Statistics." *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **181**(4), 927–979.

Varian HR (2014). "Big Data: New Tricks for Econometrics." *Journal of Economic Perspectives*, **28**(2), 3–28.

Wang J, Chen LS (2016). *MixRF: A Random-Forest-Based Approach for Imputing Clustered Incomplete Data*. R package version 1.0, URL https://CRAN.R-project.org/package=MixRF.

Warnholz S (2018). *saeRobust: Robust Small Area Estimation*. R package version 0.2.0, URL https://CRAN.R-project.org/package=saeRobust.

Wickham H (2011). "ggplot2." *Wiley interdisciplinary reviews: computational statistics*, **3**(2), 180–185.

Wright MN, Ziegler A (2017). "ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R." *Journal of Statistical Software*, **77**(1), 1–17.

Affiliation:

Patrick Krennmair
Institute for Statistics and Econometrics
School of Business & Economics
Freie Universität Berlin
14195 Berlin, Germany
E-mail: patrick.krennmair@fu-berlin.de