Prediction, Bootstrapping and Monte Carlo Analyses Based on Linear Mixed Models with QAPE 2.0 Package

by Alicja Wolny–Dominiak and Tomasz Żądło

Abstract The paper presents a new R package **qape** for prediction, accuracy estimation of various predictors and Monte Carlo simulation studies of properties of both predictors and estimators of accuracy measures. It allows to predict any population and subpopulation characteristics of the response variable based on the Linear Mixed Model (LMM). The response variable can be transformed, e.g. to logarithm and the data can be in the cross-sectional or longitudinal framework. Three bootstrap algorithms are developed: parametric, residual and double, allowing to estimate the prediction accuracy. Analyses can also include Monte Carlo simulation studies of properties of the methods used. Unlike other packages, in the prediction process the user can flexibly define the predictor, the model, the transformation function of the response variable, the predicted characteristics and the method of accuracy estimation.

1 Introduction

One of the tasks in application of mixed models in the real-life problems is the prediction of random effects. Then, the predicted values give the possibility for further prediction, e.g. characteristics of interest such as sum, mean or quantiles or the future value of the response variable for cross-sectional or longitudinal data.

Three main predictors of these characteristics are proposed in the literature: Empirical Best Linear Unbiased Predictors - EBLUPs (see e.g. Henderson (1950) and Royall (1976)), PLUG-IN predictors (see e.g. Boubeta et al. (2016), Chwila and Żądło (2019), Hobza and Morales (2016)) and Empirical Best Predictors - EBPs (see e.g. Molina and Rao (2010)). Each assumes the LMM to model the response variable.

The numerous successful applications of these three predictors for cross-sectional and longitudinal data can be found in the model approach in survey sampling, including the small area estimation. In paper Fay III and Herriot (1979) the Authors introduce the prediction of the mean income for small places based on the special case of the LMM model called Fay-Herriot model and the EBLUP. The analysis of poverty is extended in many works, e.g. in Molina and Rao (2010) and Christiaensen et al. (2012). In turn, in Battese et al. (1988) the Authors analyse the total crop areas based on survey and satellite data using EBLUPs. The proposed LMM model is known as the Battese-Harter-Fuller model. The predictors are also exploited in the subject of experience rating in non-life insurance, see Frees et al. (1999) and Bühlmann and Gisler (2005), where the longitudinal data are under consideration. The insurance premium for the next period for every policy in the insurance portfolio is predicted.

A major challenge in this type of prediction is the estimation of the prediction accuracy measure. Most often it is the Root Mean Squared Error (RMSE), which is given in analytical form or can be e.g. estimated using bootstrap. A feature of the distribution of the squared prediction error is usually a very strong positive asymmetry. Because the mean is not recommended as the appropriate measure of the central tendency in such distributions, the alternative prediction accuracy measure called the Quantile of Absolute Prediction Errors (QAPE), proposed by Żądło (2013) and Wolny-Dominiak and Żądło (2020), can be applied.

There is a variety of R packages to calculate the considered predictors together with the accuracy measure of prediction, usually the RMSE. The package **sae**, see Molina and Marhuenda (2015), provides EBLUPs based on Fay-Herriot and Battese-Harter-Fuller models. In turn, the multivariate EBLUP for Fay-Herriot models is implemented in **msae**, see Permatasari and Ubaidillah (2021). Several EBLUPs introduced in Rao and Yu (1994) are implemented in package **saery** introduced by Lefler et al. (2014), likewise in **JoSAE**, see Breidenbach (2018), but with additional heteroscedasticity analysis. The EBP is provided in the package **emdi** described in Kreutzmann et al. (2019).

A new package in this area is our proposed package **qape**. It allows the prediction of flexibly defined characteristics of the response variable using the above three predictors, assuming an appropriate LMM. A novel feature of the package **qape**, compared to those already in place, is the ability of bootstrap estimation of the prediction accuracy measures, both the RMSE and QAPE. Three types of bootstrap procedures are provided: parametric, residual and double.

There are three groups of functions in this package: predictors values calculation, bootstrap estimation of RMSE and QAPE measures, and Monte Carlo (MC) analysis of properties of predictors and prediction accuracy estimators. The prediction is based on a LMM model defined by the user and allows to predict the population characteristics of the response variable, which can be defined by a linear combination (in the case of EBLUP), by any R function (e.g. `sum`) or any function defined by the user (in the case of the EBP and PLUG-IN predictors). The package allows for full flexibility in defining: the model, the predicted characteristic, and the transformation of the response variable.

This paper is organized as follows. Firstly, the background of the LMM is presented together with the theoretical foundations of the prediction including prediction accuracy measures. Then, the package functionality in the area of prediction is presented and illustrated. A short application based on `radon` data, a cross-sectional dataset available in **HLMdiag** package, to predict three subpopulation characteristics is shown. Subsequently, the theoretical background of the prediction accuracy measures estimation based on bootstrap is presented. Implementations of bootstrap algorithms in **qape** are briefly introduced. Finally, the procedure of the model-based Monte Carlo simulation study is discussed. The paper ends with a conclusion.

2 Prediction accuracy measures

We consider the problem of prediction of any given function of the population vector $\mathbf{Y}$ of the response variable:

$$\theta = f_{\theta}(\mathbf{Y}) \tag{1}$$

under the LMM. It covers linear combinations of $\mathbf{Y}$ (such as one future realization of the response variable or population and subpopulation means and totals) but also other population and subpopulation characteristics such quantiles and variability measures.

To assess the accuracy of the particular predictor $\hat{\theta}$, firstly, the prediction error is defined as $U = \hat{\theta} - \theta$. Therefore, the well-known RMSE has the following formula:

$$RMSE(\hat{\theta}) = \sqrt{E(\hat{\theta} - \theta)^2} = \sqrt{E(U^2)}. \tag{2}$$

The alternative to the RMSE based on the mean could be the QAPE based on quantiles. It represents the pth quantile of the absolute prediction error $|U|$, see Żądło (2013) and Wolny-Dominiak and Żądło (2020), and it is given by:

$$QAPE_p(\hat{\theta}) = \inf\left\{x : P\left(\left|\hat{\theta} - \theta\right| \leq x\right) \geq p\right\} = \inf\left\{x : P\left(|U| \leq x\right) \geq p\right\} \tag{3}$$

This measure informs that at least $p100\%$ of observed absolute prediction errors are smaller than or equal to $QAPE_p(\hat{\theta})$, while at least $(1-p)100\%$ of them are higher than or equal to $QAPE_p(\hat{\theta})$. Quantiles reflect the relation between the magnitude of the error and the probability of its realization. It means that using the QAPE, it is possible to make a full description of the distribution of prediction errors instead of using the average (reflected by the RMSE). Furthermore, the MSE is the mean of positively (usually very strongly) skewed squared prediction errors, where the mean should not be used as a measure of the central tendency of positively skewed distributions.

The above described accuracy prediction measures RMSE and QAPE can be estimated using the bootstrap techniques. Their estimators as well as the bootstrap distributions of the prediction errors based on any (assumed or misspecified) model are provided in **qape** package, including algorithms where the parallel computing is used.

In the **qape** package, the whole prediction process has its own specific procedure, which can be presented in the following steps.

Procedure 1 The process of prediction, accuracy measures estimation and Monte Carlo simulation analyses in **qape**

1. Define the characteristics of the response variable to predict,
2. provide the information on sample and population values,
3. define the LMM,
4. estimate parameters of the LMM,
5. predict the random variable θ using the chosen class of predictors,
6. estimate the prediction accuracy measures RMSE and QAPE using one of the developed bootstrap algorithms,
7. conduct simulation analyses of properties of predictors and accuracy measures estimators under any (also misspecified) LMM model.

3 The prediction under LMM

The main functions of the **qape** package provide the bootstrap estimation of prediction accuracy measures. However, it must be preceded by the prediction process, including the choice of the LMM and the predictor.

3.1 The model

Let $\mathbf{Y}$ denote the vector of response variables $Y_1, Y_2, ..., Y_N$. Assuming, without a loss of generality, that only the first n realizations of Y_i are observed, $\mathbf{Y}$ can be decomposed as $\mathbf{Y} = \begin{bmatrix}\mathbf{Y}_s^T & \mathbf{Y}_r^T\end{bmatrix}^T$, where $\mathbf{Y}_s$ and $\mathbf{Y}_r$ are of dimension $n \times 1$ and $(N-n) \times 1$, respectively. In all notations, the subscript "s" is used for observed realizations of the variable of interest and "r" for the unobserved ones. Two known matrices of auxiliary variables are also considered, denoted by $\mathbf{X}$ and $\mathbf{Z}$, which are associated with fixed and random effects, respectively. The $\mathbf{X}$ matrix is of dimension $N \times p$, and it consists of p regression variables. It can be decomposed like $\mathbf{Y}$ as follows: $\mathbf{X} = \begin{bmatrix}\mathbf{X}_s^T & \mathbf{X}_r^T\end{bmatrix}^T$, where matrices $\mathbf{X}_s$ and $\mathbf{X}_r$, both known, are of dimension $n \times p$ and $(N-n) \times p$, respectively. Similarly, the $\mathbf{Z}$ matrix of dimension $N \times h$ can be written as follows: $\mathbf{Z} = \begin{bmatrix}\mathbf{Z}_s^T & \mathbf{Z}_r^T\end{bmatrix}^T$, where matrices $\mathbf{Z}_s$ and $\mathbf{Z}_r$, both known, are of dimension $n \times h$ and $(N-n) \times h$, respectively.

Then, let $LMM(\mathbf{X}, \mathbf{Z}, \boldsymbol{\psi})$ denotes the LMM of the following form (e.g. Rao and Molina (2015), p. 98):

$$\begin{cases} \mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{v} + \mathbf{e} \\ E(\mathbf{e}) = \mathbf{0}, E(\mathbf{v}) = \mathbf{0} \\ Var(\mathbf{e}) = \mathbf{R}(\boldsymbol{\delta}), Var(\mathbf{v}) = \mathbf{G}(\boldsymbol{\delta}) \end{cases} \tag{4}$$

The vector of parameters in model (4) is then $\boldsymbol{\psi} = \begin{bmatrix}\boldsymbol{\beta}^T & \boldsymbol{\delta}^T\end{bmatrix}^T$, where $\boldsymbol{\beta}$ is a vector of fixed effects of dimension $p \times 1$ and $\boldsymbol{\delta}$ is a vector of variance components. The random part of the model is described by the known matrix $\mathbf{Z}$, a vector $\mathbf{v}$ of random effects of dimension $h \times 1$ and a vector $\mathbf{e}$ of random components of dimension $N \times 1$, where $\mathbf{e}$ and $\mathbf{v}$ are assumed to be independent. The vector of random components $\mathbf{e}$ will be decomposed similarly to the vector $\mathbf{Y}$, i.e. $\mathbf{e} = \begin{bmatrix}\mathbf{e}_s^T & \mathbf{e}_r^T\end{bmatrix}^T$.

In the residual bootstrap implemented in **qape**, there is a need to re-write the LMM model to take account of the specific structure of data, i.e. the grouping variables taken into account in the random part of the model. In this case, without a loss of the generality, the LMM model can be written as follows:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}_1\mathbf{v}_1 + ... + \mathbf{Z}_l\mathbf{v}_l + ... + \mathbf{Z}_L\mathbf{v}_L + \mathbf{e}, \tag{5}$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_l, \ldots, \mathbf{v}_L$ are independent vectors of random effects assumed for different divisions of the $\mathbf{Y}$ vector (under different grouping of the data) and $\mathbf{Z}_1, \ldots, \mathbf{Z}_l, \ldots, \mathbf{Z}_L$ are known matrices of auxiliary variables associated with random effects. Writing in (5): $\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 & \ldots & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \ddots & & & \vdots \\ \mathbf{0} & \ldots & \mathbf{Z}_l & \ldots & \mathbf{0} \\ \vdots & & & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{Z}_L \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix}\mathbf{v}_1^T & \ldots & \mathbf{v}_l^T & \ldots & \mathbf{v}_L^T\end{bmatrix}^T$ the LMM model is obtained. Let

$$\mathbf{v}_l = \left[\mathbf{v}_{l1}^T \ldots \mathbf{v}_{lk}^T \ldots \mathbf{v}_{lK_l}^T\right]^T \tag{6}$$

be of dimension $K_l J_l \times 1$, where $\mathbf{v}_{lk}$ is of dimension $J_l \times 1$ for all $k = 1, ..., K_l$ and K_l is the number of random effects at the lth level of grouping. Hence, $\mathbf{Z}_l$ is $N \times K_l J_l$. For example, if the random regression coefficient model is considered with two random coefficients where both random effects are subpopulation-specific, where D is the number of subpopulations, then $L = 1$, $K_1 = 2$ and $J_1 = D$.

3.2 Predictors

In the **qape** package, in the general case the predicted characteristic is given by any function of response variables:

$$\theta = f_\theta(\mathbf{Y}). \tag{7}$$

Under the $LMM(\mathbf{X}, \mathbf{Z}, \boldsymbol{\psi})$ model it could be predicted using one of three predictors:

1. Empirical Best Linear Unbiased Predictor (EBLUP),
2. Empirical Best Predictor (EBP) under nested error LMM,

3. PLUG-IN predictor under the LMM.

The first predictor (EBLUP) allows to predict the linear combination of the response variables:

$$\theta = f_\theta(\mathbf{Y}) = \gamma^T\mathbf{Y} = \gamma_s^T\mathbf{Y}_s + \gamma_r^T\mathbf{Y}_r, \tag{8}$$

where γ is a vector of weights. In this case, the predicted characteristic θ is basically the linear combination of the response variable. For example, if one of the elements of γ equals 1 and the rest of the elements equals 0, then one realization of the response variable is predicted. If all elements in γ vector equal 1, then θ becomes the sum of all Y_i's in the whole considered population dataset. The two-stage EBLUP corresponds to the Best Linear Unbiased Predictor (BLUP) introduced in Henderson (1950) and Royall (1976) as:

$$\hat{\theta}^{BLUP}(\boldsymbol{\delta}) = \gamma_s^T\mathbf{Y}_s + \hat{\theta}_r(\boldsymbol{\delta}), \tag{9}$$

where the predictor of the linear combination $\gamma_r^T\mathbf{Y}_r$ of unobserved random variables is given by $\hat{\theta}_r(\boldsymbol{\delta}) = \gamma_r^T\mathbf{X}_r\tilde{\boldsymbol{\beta}}(\boldsymbol{\delta}) + \gamma_r^T\mathbf{Z}_r\tilde{\mathbf{v}}(\boldsymbol{\delta})$, where $\tilde{\boldsymbol{\beta}}(\boldsymbol{\delta})$ is the Best Linear Unbiased Estimator of $\boldsymbol{\beta}$ and $\tilde{\mathbf{v}}(\boldsymbol{\delta})$ is the Best Linear Unbiased Predictor of $\mathbf{v}$, both presented in (4). As shown by Żądło (2017) p. 8094, if $Cov(\mathbf{e}_r, \mathbf{e}_s) = \mathbf{0}$, then the predictor (9) is the BLUP of θ defined as the linear combination (8). Even if $Cov(\mathbf{e}_r, \mathbf{e}_s) \neq \mathbf{0}$, the predictor $\hat{\theta}_r(\boldsymbol{\delta})$ is the Best Linear Unbiased Predictor of the following linear combination of $\boldsymbol{\beta}$ and $\mathbf{v}$: $\gamma_r^T\mathbf{X}_r\boldsymbol{\beta} + \gamma_r^T\mathbf{Z}_r\mathbf{v}$. The EBLUP $\hat{\theta}^{EBLUP}$ is obtained by replacing the vector of variance components $\boldsymbol{\delta}$ in BLUP (9) with the estimator $\hat{\boldsymbol{\delta}}$. If (a) the expectation of the predictor is finite, (b) $\hat{\boldsymbol{\delta}}$ is any even, translation-invariant estimator of $\boldsymbol{\delta}$, (c) the distributions of both random effects and random components are symmetric around $\mathbf{0}$ (not necessarily normal), the EBLUP remains unbiased, as proved by Kackar and Harville (1981).

To introduce the second predictor, called EBP, considered e.g. by Molina and Rao (2010), firstly, the Best Predictor (BP) $\hat{\theta}^{BP}$ of characteristic $\theta(\mathbf{Y})$ has to be defined. It is computed by minimizing the Mean Squared Error $MSE(\hat{\theta}) = E(\hat{\theta} - \theta)^2$ and can be written as $\hat{\theta}^{BP} = E(\theta|\mathbf{Y}_s)$. It means that the conditional distribution of $\mathbf{Y}_r|\mathbf{Y}_s$ must be known to compute its value while at least the parameters of this distribution, denoted by $\boldsymbol{\psi}$ in (4), are unknown. The EBP $\hat{\theta}^{EBP}$ is obtained by replacing these parameters with estimators $\hat{\boldsymbol{\psi}}$. Its value can be computed according to the Monte Carlo procedure presented in the supplementary document for this paper.

The last predictor is the PLUG-IN predictor defined as (e.g. Chwila and Żądło (2019)):

$$\hat{\theta}^{PLUG-IN} = \theta(\begin{bmatrix}\mathbf{Y}_s^T & \hat{\mathbf{Y}}_r^T\end{bmatrix}^T), \tag{10}$$

where $\hat{\mathbf{Y}}_r$ is the vector of fitted values of unobserved random variables under the assumed model (any model specified by the statistician). Under the LMM and if the linear combination of $\mathbf{Y}$ is predicted, the PLUG-IN predictor is the EBLUP, but generally, it is not optimal. However, it was shown in simulation studies that it can have similar or even higher accuracy compared to empirical (estimated) best predictors, where the best predictors minimize the prediction mean squared errors (cf. e.g. Boubeta et al. (2016), Chwila and Żądło (2019), Hobza and Morales (2016)). Moreover, the PLUG-IN predictor is less computationally demanding than the EBP.

3.3 Predictors in qape

To deal with the LMM model, the **qape** package uses the `lmer()` function from the **lme4** package, see Bates et al. (2015). Assuming (4) and based on $\mathbf{Y}_s$, the vector of model parameters $\boldsymbol{\psi} = [\boldsymbol{\beta}^T, \boldsymbol{\delta}^T]^T$ is estimated using the Restricted Maximum Likelihood Method (REML), known to be robust on non-normality, see e.g Jiang (1996), and $\hat{\boldsymbol{\psi}}$ is obtained.

In order to obtain the predictor of θ, one of the three **qape** functions can be applied: `EBLUP()`, `ebpLMMne()` or `plugInLMM()`. Firstly, the characteristic of response variables of interest has to be defined. It is actually obvious for EBLUP, which can be used only to predict the population/subpopulation linear combination (e.g. the sum) by using the argument `gamma` equivalent to the population vector of weights γ in (8). For other two predictors, the EBP and the PLUG-IN, the input argument called `thetaFun` has to be given (see $f_\theta(.)$ in (7)). Function `thetaFun` could define one characteristic or a vector of characteristics, for example:

```
> thetaFun1 <- function(x) median(x)
> thetaFun2 <- function(x) c(sum(x), mean(x), sd(x))
```

Secondly, two groups of input arguments, common to all three predictors, has to be provided:

- group 1 - arguments defining the sample and the population
  - `YS` - values of the dependent variable in the sample ($\mathbf{Y}_s$),

  - `reg` - the population matrix of auxiliary variables named in `fixed.part`, `random.part` and `division`,
  - `con` - the population $0-1$ vector with 1s for elements in the sample and 0s for elements which are not in the sample,
- group 2 - arguments defining the model
  - `fixed.part` - fixed-effects terms declared as in `lm4::lmer` function,
  - `random.part` - random-effects terms declared as in `lm4::lmer` function,
  - `weights` - the population vector of weights.

The weights make it possible to include heteroscedasticity of random components in the LMM.

In `EBLUP()` and `plugInLMM()` the random-effects terms of the LMM have to be declared as the input argument `random.part`. The form of the `ebpLMMne` predictor, in turn, requires defining in the `ebpLMMne()` function the so-called `division` argument instead of `random.part`. This input represents the variable dividing the population dataset into subsets, which are taken into account in the nested error linear mixed model with ’`division`’-specific random components (presented in supplementary document for this paper).

In the process of prediction, it is often necessary to perform data transformation before estimating the model parameters. An example is the logarithmic scaling of the variable of interest. The **qape** package offers the possibility for declaring the argument `backTrans` to conduct the data back-transformation. Hence, a very flexible solution is used which allows to use any transformation of the response variable such that the back-transformation can be defined. This argument (available in R or defined by the user function) should be the back-transformation function of the already transformed dependent variable used to define the model, e.g. for log-transformed `YS` used as the response variable:

```
> backTrans <- function(x) exp(x)
```

The main output is the value of predictor `thetaP`. For each class of predictors, there are two S3 methods registered for existing generic functions `print` and `summary`. The full list of output arguments is presented in detail in the `qape-manual` file, cf. Wolny-Dominiak and Żądło (2023).

3.4 Radon data and the model

In order to demonstrate the functionality of the package’s main functions, in the following examples the `radon` dataset available in **HLMdiag** package (Loy and Hofmann (2014)) is analyzed. It contains the results of a survey measuring radon concentrations in 919 owner-occupied homes in 85 counties of Minnesota (see Figure 1). A study was conducted in 1987-1988 by the Minnesota Department of Health, showing that indoor radon levels are higher in Minnesota compared to typical levels in the U.S. In the data, the response variable `log.radon` (denoted in (11) by $log(Y_{ic})$) is the radon measurement in logarithms of picoCurie per liter. The independent variables, on the other hand, are: `uranium` (x_{1ic}) the average county-level soil uranium content, `basement` (x_{2ic}) the 0-1 variable indicating the level of the home at which the radon measurement was taken - 0 for basement, 1 for the first floor, and `county` (denoted by subscript c in (11)) is county ID.

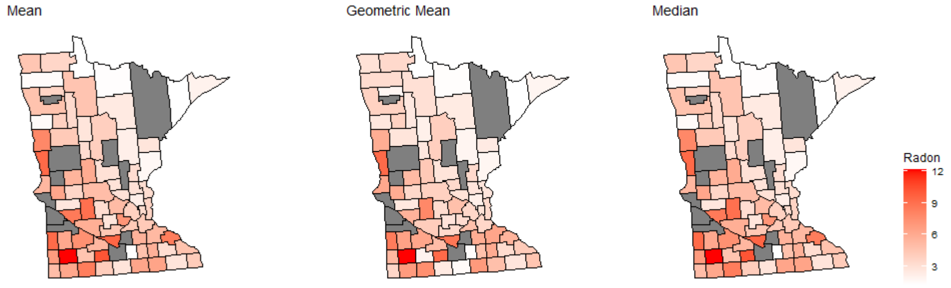

Figure 1: The maps of characteristics of radon concentration in counties in picoCurie per liter. The gray colour means that the value is NA (Not Available)

In all considered examples, the prediction for the county no. 26 (`county == 26`) is conducted and it is assumed that the observations in this county from the first floor (`basement == 1`) are not available (see Figure 2).

The `radon` dataset is widely discussed in the literature. In the paper Nero et al. (1994), the Authors used an ordinary regression model to predict county geometric means of radon concentration using

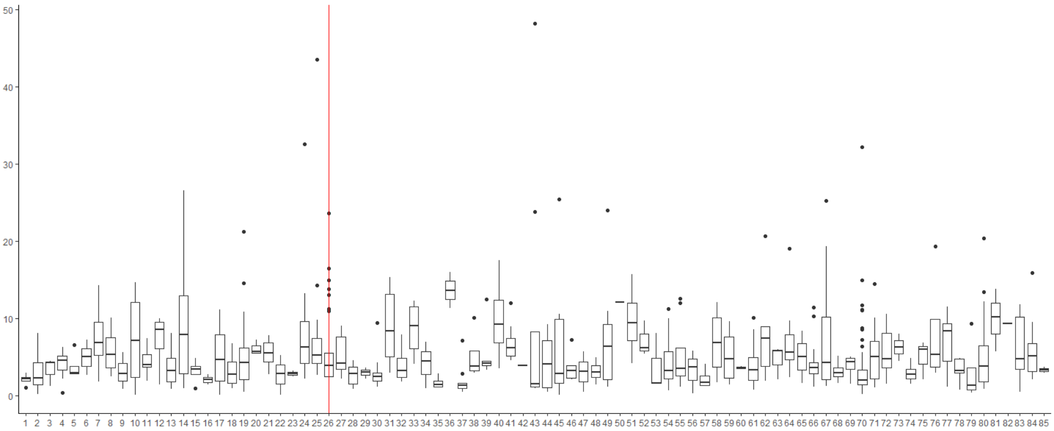

Figure 2: The distributions of radon concentration in picoCurie per liter in counties. The red line indicates county no. 26

surficial soil radium data from the National Uranium Resource Evaluation. In turn, the paper Price et al. (1996) focuses on the prediction of the geometric mean of radon for each county, but using a Bayesian approach. For the radon data we use the following model

$$log(Y_{ic}) = \beta_1 x_{1ic} + (\beta_2 + v_{1c})x_{2ic} + \beta_0 + v_{2c} + e_{ic}, \tag{11}$$

where $i = 1, 2, \dots, N$, $c = 1, 2, \dots, C$, $N = 919$ observations, $C = 85$ counties, β_1, β_2 and β_0 are unknown fixed effects, v_{1c} and v_{2c} are random effects, e_{ic} are random components, v_{1c}, and e_{ic} are mutually independent, v_{2c} and e_{ic} are mutually independent too, $Cor(v_{1c}, v_{2c}) = \rho$, $v_{1c} \sim (0, \sigma^2_{v_1})$, $v_{2c} \sim (0, \sigma^2_{v_2})$ and $e_{ic} \sim (0, \sigma^2_e)$. As can easily be seen, the considered model is the random coefficient model with two correlated county-specific random effects. Its syntax written using the package **lme4** notation is as follows:

```
radon.model <- lmer(log.radon ~ basement + uranium + (basement | county), data = radon)
```

This and similar LMMs are considered, analyzed, and used for the considered dataset in many publications, with a good overview presented in Gelman and Hill (2006). In Gelman and Pardoe (2006), based on their preceding research Price et al. (1996), Lin et al. (1999), Price and Gelman (2005), a very similar model but with additional multivariate normality assumptions is studied, verified and chosen as fitting well to the data within a Bayesian framework. The same model as in Gelman and Pardoe (2006) with its special cases is considered in Cantoni et al. (2021) but within the frequentist approach. Based on 25 measures of explained variation and model selection, the Authors conclude that the same model as considered in our paper (with additional normality assumption, however, which is not used in all cases considered in that paper), "seems the best" (Cantoni et al., 2021, p. 10) for the radon data. Further tests of the model are presented by Loy (2013), Loy and Hofmann (2015) and Loy et al. (2017) (see also Cook et al. (2007) for the introduction of the methodology) showing among others: the normality and homescedasticity of random components, the normality of the distribution of the random slope but – what is important for our further considerations – the lack of the normality of the random intercept. Since the problem of choosing and verifying a model for the considered dataset is widely discussed in the literature, we will focus on the issues that are new in this case, namely the problem of prediction and estimation of the prediction accuracy as well as the Monte Carlo analysis of predictors' properties.

3.5 Example 1

This example shows the prediction procedure in the package **qape**. In the first step, it is needed to define all the input arguments that will then be passed to the prediction functions.

```
> Ypop <- radon$log.radon # the population vector of the dependent variable
> # It is assumed that observations from the first floor
> # in county no. 26 are not available:
> con <- rep(1, nrow(radon))
> con[radon$county == 26 & radon$basement == 1] <- 0
> YS <- Ypop[con == 1] # sample vector of the dependent variable
> reg <- dplyr::select(radon, -log.radon) # the population matrix of auxiliary variables
> fixed.part <- 'basement + uranium' # the fixed part of the considered model
> random.part <- '(basement|county)' # the random part of the considered model
> # The vector of weights to define
> # the predicted linear combination -  the mean for county == 26:
> gamma <-
+   (1 / sum((radon$county == 26))) * ifelse((radon$county == 26), 1, 0)
> estMSE <- TRUE # to include the naive MSE estimator of the EBLUP in the output
```

Then the functions corresponding to each predictor can be used. First, the EBLUP prediction in the package **qape** is presented. As the EBLUP is limited to the linear combination of random variables, the predicted characteristic is simply the arithmetic mean. To be precise, it is the mean of logarithms of measurements (instead of the mean of measurements), because the EBLUP can be used only under the linear (linearized) models. As in the LMM the homescedasticity of random components is assumed, the input argument `weights = NULL` is set up.

```
> myeblup <- EBLUP(YS, fixed.part, random.part, reg, con, gamma,  weights = NULL, estMSE)
> # the value of the predictor of the arithmetic mean
> # of logarithms of radon measurements:
> myeblup$thetaP
[1] 1.306916
> myeblup$neMSE # the value of the naive MSE estimator
[1] 0.002292732
```

Hence, the predicted value of the arithmetic mean of logarithms of radon measurements equals 1.306916 log picoCurie per liter. The estimated root of prediction MSE equals $\sqrt{0.002292732} \approx 0.048$ log picoCurie per liter, but – what is important – it is the value of the naive RMSE estimator (as defined by Rao and Molina, 2015, p. 106), which means that it ignores the decrease of accuracy due to the estimation of model parameters.

The second part of this example shows the prediction of the arithmetic mean, geometric mean and median of radon measurements (not logarithm of radon measurements) in county no. 26 with the use of the PLUG-IN predictor. It requires the setting of two input arguments: `thetaFun` and `backTrans`.

```
> thetaFun <- function(x) {
+   c(mean(x[radon$county == 26]), psych::geometric.mean(x[radon$county == 26]),
+     median(x[radon$county == 26]))
+   }
> backTransExp <- function(x) exp(x) # back-transformation
> myplugin <- plugInLMM(YS, fixed.part, random.part, reg, con, weights = NULL,
+                       backTrans = backTransExp, thetaFun)
> # values of the predictor of arithmetic mean, geometric mean
> # and median of radon measurements:
> myplugin$thetaP
[1] 3.694761 4.553745 3.900000
```

In this case we can conclude that the predicted values of the aritmethmic mean, geometric mean and median in county no. 26 equal: 3.694761, 4.553745 and 3.9 picoCurie per liter, respectively. The problem of prediction accuracy estimation will be discussed in the next sections of the paper.

The **qape** package allows to use the Empirical Best Predictor (EBP) (see the supplementary document for this paper) as well. It provides predicted values of any function of the variable of interest, as the PLUG-IN predictor. However, this requires stronger assumptions to be met. The EBP procedure available in **qape** package is prepared under the assumption of the normality of the variable of interest after any transformation. However, in the case of the considered model for logarithms of radon measurements, the assumption is not met as we mentioned before based on the results presented in the literature. It can also be verified using `normCholTest` function (available in **qape** package) as follows:

```
> normCholTest(radon.model, shapiro.test)$p.value
[1] 2.589407e-08
```

Moreover, due to the fact of very time-consuming iterative procedure used to compute the EBP for the general case, in the **qape** package the function `ebpLMMne` uses a very fast procedure working only for nested error Linear Mixed Models (see Molina and Rao (2010)).

The prediction of any function of the random variables based on cross-sectional data has been considered. Its special case, not presented above but widely discussed in the econometric literature, is the prediction of one random variable, in this case a radon measurement for one non-observed owner-occupied home. Furthermore, the **qape** package is also designed for prediction based on longitudinal data for current or future periods as shown in examples for the `EBLUP`, `plugInLMM` and `ebpLMMne` functions in the `qape-manual` file, cf. Wolny-Dominiak and Żądło (2023).

4 Bootstrap procedures

The **qape** package provides three main types of bootstrap algorithms: the parametric bootstrap, the residual bootstrap and the double-bootstrap.

The parametric bootstrap procedure is implemented according to González-Manteiga et al. (2007) and González-Manteiga et al. (2008) and could be described in the following steps:

1. based on n observations of the dependent and independent variables ($\mathbf{Y}_s$, $\mathbf{X}_s$ and $\mathbf{Z}_s$) estimate $\boldsymbol{\psi}$ to obtain the vector of estimates $\hat{\boldsymbol{\psi}}$,
2. generate B realizations $y_i^{*(b)}$ of Y_i, under the $LMM(\mathbf{X}, \mathbf{Z}, \hat{\boldsymbol{\psi}})$ and multivariate normality of random effects and random components obtaining $\mathbf{y}^{*(b)} = \begin{bmatrix} y_1^{*(b)} & \ldots & y_i^{*(b)} & \ldots & y_N^{*(b)} \end{bmatrix}^T$, where $i = 1, 2, ..., N$ and $b = 1, 2, ..., B$,
3. decompose the vector $\mathbf{y}^{*(b)}$ as follows $\begin{bmatrix} \mathbf{y}_s^{*(b)T} & \mathbf{y}_r^{*(b)T} \end{bmatrix}^T$,
4. in the bth iteration ($b = 1, 2, ..., B$)
   (a) compute the bootstrap realization $\theta^{*(b)} = \theta^{*(b)}(\mathbf{y}^{*(b)}, \hat{\boldsymbol{\psi}})$ of random variable θ,
   (b) obtain the vector of estimates $\hat{\boldsymbol{\psi}}^{*(b)}$ using $\mathbf{y}_s^{*(b)}$ and compute the bootstrap realization of predictor $\hat{\theta}$ denoted by $\hat{\theta}^{*(b)}(\mathbf{y}_s^{*(b)}, \hat{\boldsymbol{\psi}}^{*(b)})$ based on $LMM(\mathbf{X}, \mathbf{Z}, \hat{\boldsymbol{\psi}}^{*(b)})$,
   (c) compute bootstrap realizations of prediction error U^* denoted by u^* and for the bth iteration given by:
   $$u^{*(b)} = \hat{\theta}^{*(b)}(\mathbf{y}_s^{*(b)}, \hat{\boldsymbol{\psi}}^{*(b)}) - \theta^{*(b)}(\mathbf{y}^{*(b)}, \hat{\boldsymbol{\psi}}) = \hat{\theta}^{*(b)} - \theta^{*(b)}, \tag{12}$$
5. compute the parametric bootstrap estimators of prediction accuracy measures: RMSE and QAPE replacing prediction errors U in (2) and (3) by their bootstrap realizations.

Another possible method to estimate the prediction accuracy measures is the residual bootstrap. In what follows, we use the notation $srswr(\mathbf{A}, m)$ to indicate the outcome of taking a simple random sample with replacement of size m of rows of matrix $\mathbf{A}$. If $\mathbf{A}$ is a vector, it simplifies to a simple random sample with replacement of size m of elements of $\mathbf{A}$.

To obtain the algorithm of the residual bootstrap, it is enough to replace step 2 of the parametric bootstrap procedure presented above with the following procedure of the population data generation based on (5):

- generate B population vectors of the variable of interest, denoted by $\mathbf{y}^{*(b)}$ as
  $$\mathbf{y}^{*(b)} = \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{Z}_1\mathbf{v}_1^{*(b)} + ... + \mathbf{Z}_l\mathbf{v}_l^{*(b)} + ... + \mathbf{Z}_L\mathbf{v}_L^{*(b)} + \mathbf{e}^{*(b)}, \tag{13}$$
  where $\hat{\boldsymbol{\beta}}$ is an estimator (e.g. REML) of $\boldsymbol{\beta}$, $\mathbf{e}^{*(b)}$ is a vector of dimension $N \times 1$ defined as $srswr(col_{1 \le i \le n}\hat{e}_i, N)$, where $\hat{e}_i$ ($i = 1, 2, ..., n$) are residuals, $\mathbf{v}_l^{*(b)}$ (for $1, 2, ..., L$) is the vector of dimension $K_l J_l \times 1$ built from the columns of the matrix: $srswr\left(\begin{bmatrix} \hat{\mathbf{v}}_{l1} & \ldots & \hat{\mathbf{v}}_{lk} & \ldots & \hat{\mathbf{v}}_{lK_l} \end{bmatrix}, J_l\right)$ of dimension $J_l \times K_l$, where $\hat{\mathbf{v}}_{lk}$ are estimates of elements of random effects vector (6).

The next 3–5 steps in this procedure are analogous to steps in the parametric bootstrap procedure.

In the above-described step, it can be seen that if more than one vector of random effect is assumed at the lth level of grouping, then the elements are not sampled with replacement independently. In this case, rows of the matrix formed by these vectors are sampled with replacement.

The residual bootstrap algorithm can also be performed with so-called "correction procedure". This procedure, which can improve the properties of the residual bootstrap estimators due to the underdispersion of the uncorrected residual bootstrap distributions, is presented in the supplementary document for this paper.

5 Bootstrap in qape

Two bootstrap procedures are implemented in separate functions: bootPar() (the parametric bootstrap) and bootRes() (the residual bootstrap). According to the general Procedure 1, the step preceding the bootstrap procedure in both functions is the definition of the predictor object. It must be one of the following: EBLUP, ebpLMMne or plugInLMM. This object has to be passed to bootPar() or bootRes() as the input parameter predictor. The other input parameters are intuitive: B - the number of bootstrap iterations and p - order of quantiles in the estimated QAPEs.

The additional input parameter in bootRes() is a logical condition called correction, which makes it possible to include an additional correction term for both random effects and random components, presented in the supplementary document for this paper, to avoid the problem of underdispersion of residual bootstrap distributions.

The main output values in both functions are basically the measures: estRMSE and estQAPE computed based on (2) and (3), respectively, where prediction errors are replaced by their bootstrap realizations. There is also the output error being the vector of bootstrap realizations of prediction errors, which is useful e.g. in in-depth analysis of the prediction accuracy and for graphical presentation of results. To estimate these accuracy measures, we use below the residual bootstrap with the correction procedure.

As previously stated, our package utilizes the lmer() function from the lme4 package for estimating model parameters. However, this function has been known to generate convergence warnings in certain situations, listed for example by Bates et al. (2015) p. 25, when the estimated variances of random effects are close to zero. Such scenarios may occur when models are estimated for smaller or medium-sized datasets, when complex variance-covariance structures are assumed, or when the grouping variable considered for random effects has only a few levels. Although we have not observed such issues estimating model parameters based on the original dataset required to compute values of the predictors in previous sections, bootstrapping or Monte Carlo simulations are more complex cases. This is because, based on the estimates of model parameters, the values of the dependent variables are generated B times, and then model parameters are estimated in each out of B iterations. Therefore, in at least some iterations, dependent variable values may be randomly generated giving realizations, where the variance of the random effect is relatively close to zero. As a result, estimates of model parameters can be obtained; however, convergence issues implying warnings may occur. In such cases, there are at least two possible solutions. The first option is to discard iterations with warnings, which would imply that the dependent variable would not follow the assumed model as required, but instead only its conditional version with relatively high values of variances of random effects. It will imply overdispersed bootstrap distribution of random effects, which will affect the bias of the bootstrap estimators of accuracy measures. The second option is to consider all generated realizations, despite convergence warnings, as long as the parameters can be estimated for all iterations. We opted for the latter solution, as argued in Bates et al. (2015) p. 25, who noted that "being able to fit a singular model is an advantage: when the best fitting model lies on the boundary of a constrained space".

5.1 Example 2

The analyses presented in Example 1 are continued. We extend the previous results to include the issue of estimating the prediction accuracy of the considered predictors. The use of functions for this estimation primarily requires an object of class predictor, here "myplugin".

```
> class(myplugin)
[1] "plugInLMM"
```

The short chunk of the R code presents the residual bootstrap estimators of the RMSE (estRMSE) and the QAPE (estQAPE) of the PLUG-IN predictors (plugin) of previously analyzed three characteristics of radon measurements in county no. 26: the arithmetic mean, geometric mean and median. In this and subsequent examples we make the computations for relatively high number of iterations allowing, in our opinion, to get reliable results. These results are also used to prepare Figure 3. However, the computations are time-consuming. The supplementary R file contains the same chunks of the code but the number of iterations applied is smaller in order to execute the code swiftly.

```
> # accuracy measures estimates based on
> # the residual bootstrap with the correction:
> B <- 500 # number of bootstrap iterations
> p <- c(0.75, 0.9) # orders of Quantiles of Absolute Prediction Error
> set.seed(1056)
> residBoot <- bootRes(myplugin, B, p, correction = TRUE)
> # values of estimated RMSEs of the predictor of three characteristics:
> # the arithmetic mean, geometric mean and median of radon measurements, respectively:
> residBoot$estRMSE
[1] 0.1848028 0.2003681 0.2824359
> # values of estimated QAPEs
> # (of order 0.75 in the first row, and of order 0.9 in the second row)
> # of the predictor of three characteristics:
> # the arithmetic mean, geometric mean and median of radon measurements,
> # in the 1st, 2nd and 3rd column, respectively:
> residBoot$estQAPE
         [,1]      [,2]      [,3]
75% 0.1533405 0.2135476 0.2908988
90% 0.2813886 0.3397411 0.4374534
```

Let us concentrate on interpretations of estimators of accuracy measures for the predictor of the geometric mean, i.e. the second value of `residBoot$estRMSE`, and values in the second column of `residBoot$estQAPE`. It is estimated that the average difference between predicted values of the geometric mean and their unknown realizations equals 0.2003681 picoCurie per liter. Furthermore, it is estimated that at least 75% of absolute prediction errors of the predictor of the geometric mean are smaller or equal to 0.2135476 picoCurie per liter and at least 25% of absolute prediction errors of the predictor are higher or equal to 0.2135476 picoCurie per liter. Finally, it is estimated that at least 90% of absolute prediction errors of the predictor of the geometric mean are smaller or equal to 0.3397411 picoCurie per liter and at least 10% of absolute prediction errors of the predictor are higher or equal to 0.3397411 picoCurie per liter. The distributions of bootstrap absolute prediction errors with values of estimated RMSEs and QAPEs for the considered three prediction problems are presented in Figure 3.

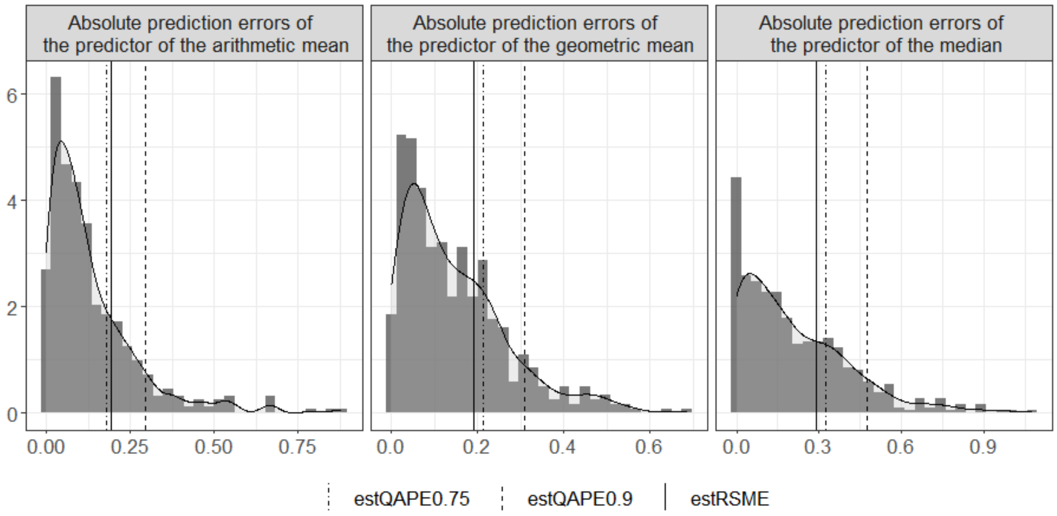

Figure 3: The histograms of bootstrap absolute prediction errors for `myplugin` (for PLUG-IN predictors of the arithmetic mean, geometric mean and median) for $B = 500$

Since the assumption of normality is not met, the parametric bootstrap should not be used in this case. For this reason, we do not present the results for this method below, although – but for illustrative purposes only – they are presented in the supplementary R file. Moreover, these analyses can also be conducted using `bootParFuture()` and `bootResFuture()` functions where parallel computing algorithms are applied. The input arguments and the output of these functions are the same as in `bootPar()` and `bootRes()`. Examples based on these functions are also included in the supplementary R file.

6 Bootstrap under the misspecified model in qape

The qape package also allows to use predictors under a model different from the assumed one (e.g. a simpler or more robust model), but estimate its accuracy under the assumed model. In this case, the parametric and residual bootstrap procedures are implemented in `bootParMis()` and `bootResMis()` functions. These functions allow to estimate the accuracy of two predictors under the model correctly specified for the first of them. Of course, it is expected that the estimated accuracy of the first predictor will be better than of the second one, but the key issue can be the difference between estimates of accuracy measures. A small difference, even to the second predictor's disadvantage, may be treated by the user as an argument for using the second predictor due to its properties, such as robustness or simplicity.

The considered functions allow to estimate the accuracy of two predictors, which belong to the class `plugInLMM`, under the model used to define the first of them. The remaining arguments are the same as in `bootPar()` and `bootRes()` functions: `B` - the number of bootstrap iterations, and `p` - orders of QAPE estimates to be taken into account.

The output results of `bootParMis()` and `bootResMis()` include – similarly to `bootPar()` and `bootRes()` functions – estimates of the RMSEs and QAPEs of both predictors (denoted here by: `estRMSElmm`, `estRMSElmmMis`, `estQAPElmm` and `estQAPElmmMis`), and boostrap realizations of their prediction errors (`errorLMM` and `errorLMMmis`).

6.1 Example 3

In this example, we study the same accuracy measures as in Example 2, but the aim is to compare the predictor `myplugin` and other predictor defined under the misspecified LMM. First, the misspecified model has to be defined, and a relevant predictor has to be computed.

```
> fixed.part.mis <- '1'
> random.part.mis <- '(1|county)'
> myplugin.mis <- plugInLMM(YS, fixed.part.mis, random.part.mis, reg, con,
+                         weights = NULL, backTrans = backTransExp, thetaFun)
```

Having two objects: `myplugin` and `myplugin.mis`, one can proceed to a comparison by estimating bootstrap prediction accuracy performed using the residual bootstrap with correction procedure. In this case, we estimate the prediction accuracy of these two predictors under the model used to define the first of them.

```
> set.seed(1056)
> residBootMis <- bootResMis(myplugin, myplugin.mis, B, p, correction = TRUE)
> # residual bootstrap with the correction RMSE estimators
> # of 'plugin' of: arithmetic mean, geometric mean and median
> # of radon measurements in county 26:
> residBootMis$estRMSElmm
[1] 0.1848028 0.2003681 0.2824359
> # residual bootstrap with the correction RMSE estimators
> # of 'plugin.mis' of: arithmetic mean, geometric mean and median
> # of radon measurements in county 26:
> residBootMis$estRMSElmmMis
[1] 0.1919184 0.3192304 0.2762137
> # residual bootstrap with the correction QAPE estimators of order 0.75 and 0.9
> # of 'plugin' of: arithmetic mean, geometric mean and median
> # of radon measurements in county 26:
> residBootMis$estQAPElmm
         [,1]      [,2]      [,3]
75% 0.1533405 0.2135476 0.2908988
90% 0.2813886 0.3397411 0.4374534
> # residual bootstrap with the correction QAPE estimators of order 0.75 and 0.9
> # of 'plugin.mis' of: arithmetic mean, geometric mean and median
> # of radon measurements in county 26:
> residBootMis$estQAPElmmMis
         [,1]      [,2]      [,3]
75% 0.2267062 0.3802836 0.3255197
90% 0.2813787 0.4970726 0.4489399
```

The results, presented above, were obtained for the same number of bootstrap iterations as in Example 2 ($B = 500$). If we compare, under the model defined in `plugin`, estimated RMSEs of `plugin` and `plugin.mis` predictors of the geometric mean given by 0.2003681 and 0.3192304 picoCurie per liter, respectively, we can state that the estimated accuracy (measured by RMSE estimators) of the first predictor is better comparing with the second one. If we are not interested in the average accuracy measures but in the right tail of the distribution of prediction errors, we can use estimates of QAPE of order 0.9 to compare the accuracy. The result for the `plugin.mis` of the geometric mean equals to

0.4970726 picoCurie per liter, and it is higher comparing with 0.3397411 picoCurie per liter obtained for `plugin` for the same prediction problem. Hence, in this case, the accuracy comparison based both on the RMSE and QAPE leads to the same finding.

In the previous paragraph, we have focused on the results for the case of prediction of the geometric mean. If the comparison is made for the case of prediction of the arithmetic mean (the first column of output results) or the median (the third column of output results), we will come to the same conclusion regarding the estimated accuracy of `plugin` and `plugin.mis` as in the case of prediction of the geometric mean.

Similarly to the residual bootstrap, the parametric bootstrap procedure `paramBootMis` available in **qape** package can be performed. However, in the considered case the normality assumption is not met (as discussed above) and the procedure is not recommended. The appropriate chunk of the R code is presented in the supplementary R file, but it is solely intended for illustrative purposes.

7 Monte Carlo simulation analyses

In the previous section, our aim was to estimate the prediction accuracy under correctly specified or misspecified model. In this section, we do not estimate the accuracy, but we approximate the true prediction accuracy under the specified model in the Monte Carlo simulation study. The crucial difference is that in this case, the model parameters used are obtained based on the whole population dataset, not the sample. If the number of iterations is large enough, we can treat the computed values of the measures as their true values, which are unknown in practice.

The last step of the analysis in **qape** package presented in Procedure 1 is the Monte Carlo (MC) simulation analysis of:

- properties of predictors
- and properties of parametric, residual and double bootstrap estimators of accuracy measures.

The whole Monte Carlo procedure is as follows.

Procedure 2 Model-based Monte Carlo simulation analyses in **qape**

```
1. define the population vector of the dependent variable and the population matrix of auxiliary variables,
2. provide the information on the division of the population into the sampled and non-sampled part,
3. define $\theta$ - the characteristics of the response variable to be predicted,
4. define the predictors $\hat{\theta}$ and accuracy measures estimators which properties are to be assessed,
5. define the model to be used to generate realizations of the values of the dependent variable and estimate its parameters based on population data,
6. For k=1, 2, ..., K
    6.1. generate the population vector of the response variable based on the assumed model,
    6.2. based on population data, compute the characteristics $\theta$, denoted by $\theta_k$,
    6.3. based on sample data, estimate the parameters of the LMM,
    6.4. based on sample data, compute values of predictors $\hat{\theta}$, denoted by $\hat{\theta}_k$,
    6.5. based on sample data, estimate the accuracy of $\hat{\theta}$ using bootstrap methods,
7. End For
8. compute accuracy measures of predictors using $\hat{\theta}_k$ and $\theta_k$ (for $k = 1, 2, ..., K$),
9. compute accuracy measures of estimators of prediction accuracy measures.
```

8 Monte Carlo analyses in qape

In order to perform a Monte Carlo (MC) analysis on the properties of predictors, it is necessary to have access to the entire population data for both dependent and independent variables. The function `mcLMMmis()` can be used with the following arguments. Firstly, the population values of the dependent variable (after a necessary transformation) should be declared as `Ypop`. By using the `Ypop` values, we can estimate the model parameters based on the entire population data (assuming that they are known). This allows us to generate values of the dependent variable in the simulation study that can mimic its distribution in the entire population, not just in the sample. This approach ensures that our simulation study can be an accurate representation of the random process in the entire population, resembling the real-world scenario. Secondly, three predictors: predictorLMMmis, predictorLMM, predictorLMM2, which belong to the class plugInLMM, are to be defined. The first one is used only to define the (possibly misspecified) model used to generate population values of the response variables. Accuracy of predictorLMM and predictorLMM2 is assessed in the simulation study. The next two arguments include the number of MC iterations K and orders p of QAPEs used to assess the prediction accuracy. Finally, it should be noted that it is possible to modify covariance matrices of random components and random effects based on the model defined in predictorLMMmis, which are used tThiso generate values of the dependent variable. It is possible by declaring values of ratioR and ratioG arguments, which the diagonal elements of covariance matrices of random components and random effects, respectively, are divided by.

The output of this function covers the following statistics of both predictors computed in the simulation study: relative biases (rBlmm and rBlmm2), relative RMSEs (rRMSElmm and rRMSElmm2) and QAPEs (QAPElmm and QAPElmm2). Simulation-based prediction errors of both predictors (errorLMM and errorLMM2) are also taken into account.

8.1 Example 4

In the example, an MC simulation is carried out assuming the myplugin predictor. The goal is to approximate the true accuracy of the prediction assuming model (11). Hence, in the package **qape**, all input predictor objects in the function mcLMMmis have to be defined as myplugin.

```
> # input arguments:
predictorLMMmis <- myplugin # to define the model
predictorLMM <- myplugin # which properties are assessed in the simulation study
predictorLMM2 <- myplugin  # which properties are assessed in the sim. study
```

Except that no modification of covariance matrices has to be used.

```
# diag. elements of the covariance matrix of random components are divided by:
ratioR <- 1
# diag. elements of the covariance matrix of random effects are divided by:
ratioG <- 1
```

We specify the number of Monte Carlo iterations.

```
K <- 500 # the number of MC iterations
```

The analysis is conducted in the object MC.

```
> set.seed(1086)
> MC <- mcLMMmis(Ypop, predictorLMMmis, predictorLMM, predictorLMM2,
+                        K, p, ratioR, ratioG)
> # relative bias of 'predictorLMM'
> # of the arithmetic mean, geometric mean and median in county 26 (in %):
> MC$rBlmm
[1] -1.73208393 -0.04053178 -5.22355236
```

Results of the relative biases are obtained. It is seen, that under the assumed model the values of the considered predictor of the geometric mean (the second value of MC$rBlmm) are smaller than possible realizations of the geometric mean on average by 0.04053178%. In turn, the relative RMSEs are as follows.

```
> # relative RMSE of 'predictorLMM'
> # of the arithmetic mean, geometric mean and median in county 26 (in %):
> MC$rRMSElmm
[1] 3.429465 4.665810 7.146678
```

In the considered case, the average difference between predicted values of the geometric mean and its possible realizations (the second value of MC$rRMSElmm) equals 4.665810%. It should be noted that this value can be treated as the true value of the relative RMSE (if the number of iterations is large enough), not the estimated value obtained in Examples 2 and 3.

Finally, QAPEs of orders 0.75 and 0.9 are considered.

```
> # QAPE of order 0.75 and 0.9 of 'predictorLMM'
> # of the arithmetic mean, geometric mean and median in county 26:
> MC$QAPElmm
         [,1]      [,2]      [,3]
75% 0.1491262 0.1989504 0.2919221
90% 0.2895684 0.2959457 0.4728064
```

Let us interpret the results presented in the second column of MC$QAPElmm. At least 75% (90%) of absolute prediction errors of the predictor of the geometric mean are smaller or equal to 0.1989504 (0.2959457) picoCurie per liter and at least 25% (10%) of absolute prediction errors of the predictor are higher or equal to 0.1989504 (0.2959457) picoCurie per liter. Similar to the values of the rRMSEs in the previous code chunk, the values can be considered to be true QAPE values, not the estimates presented in Examples 2 and 3.

In Example 4, the accuracy of one predictor under the model used to define this predictor was presented. A more complex version of the simulation study, where the properties of two predictors are studied under the model defined by the third predictor, is presented in the supplementary R file. What is more, the **qape** package also allows to use mcBootMis() function to conduct MC analyses of properties of accuracy measure estimators (estimators of MSEs and QAPEs) of two predictors (which belong to the class plugInLMM) declared as arguments. The model used in the simulation study is declared in the first predictor, but the properties of accuracy measures estimators of both predictors are studied. Output results of mcBootMis() covers simulation results on properties of different accuracy measures estimators, including the relative biases and relative RMSEs of the parametric bootstrap MSE estimators of both predictors. The same simulation-based statistics but for parametric bootstrap QAPE estimators are also included. Other bootstrap methods, including the residual bootstrap with and without the correction procedure, are also taken into account. The full list of output arguments of mcBootMis() function are presented in qape-manual file, cf. Wolny-Dominiak and Żądło (2023).

9 Conclusions

The package enables R users to make predictions and assess the accuracy under linear mixed models based on different methods in a fast and intuitive manner – not only based on the RMSE but also based on Quantiles of Absolute Prediction Errors. It also covers functions which allow to conduct Monte Carlo simulation analyses of properties of the methods of users interest. Its main advantage, compared to other packages, is the considerable flexibility in terms of defining the model (as in the **lme4** package) and the predicted characteristic, but also the transformation of the response variable.

In our opinion, the package is useful for scientists, practitioners and decision-makers in all areas of research where accurate estimates and forecasts for different types of data (including cross-sectional and longitudinal data) and for different characteristics play the crucial role. We believe that it will be of special interest to survey statisticians interested in the prediction for subpopulations with small or even zero sample sizes, called small areas.

References

D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. doi: 10.18637/jss.v067.i01. [p70, 75]

G. E. Battese, R. M. Harter, and W. A. Fuller. 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, 1988. [p67]

M. Boubeta, M. J. Lombardía, and D. Morales. Empirical best prediction under area-level poisson mixed models. *Test*, 25(3):548–569, 2016. [p67, 70]

J. Breidenbach. *JoSAE: Unit-Level and Area-Level Small Area Estimation*, 2018. URL https://CRAN.R-project.org/package=JoSAE. R package version 0.3.0. [p67]

H. Bühlmann and A. Gisler. *A course in credibility theory and its applications*. Springer, 2005. [p67]

E. Cantoni, N. Jacot, and P. Ghisletta. Review and comparison of measures of explained variation and model selection in linear mixed-effects models. *Econometrics and Statistics*, 2021. [p72]

L. Christiaensen, P. Lanjouw, J. Luoto, and D. Stifel. Small area estimation-based prediction methods to track poverty: validation and applications. *The Journal of Economic Inequality*, 10(2):267–297, 2012. [p67]

A. Chwila and T. Żądło. On properties of empirical best predictors. *Communications in Statistics-Simulation and Computation*, pages 1–34, 2019. [p67, 70]

D. Cook, D. F. Swayne, and A. Buja. *Interactive and dynamic graphics for data analysis: with R and GGobi*, volume 1. Springer, 2007. [p72]

R. E. Fay III and R. A. Herriot. Estimates of income for small places: an application of james-stein procedures to census data. *Journal of the American Statistical Association*, 74(366a):269–277, 1979. [p67]

E. W. Frees, V. R. Young, and Y. Luo. A longitudinal data analysis interpretation of credibility models. *Insurance: Mathematics and Economics*, 24(3):229–247, 1999. [p67]

A. Gelman and J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, Cambridge ; New York, 1st edition edition, Dec. 2006. ISBN 978-0-521-68689-1. [p72]

A. Gelman and I. Pardoe. Bayesian measures of explained variance and pooling in multilevel (hierarchical) models. *Technometrics*, 48(2):241–251, 2006. [p72]

W. González-Manteiga, M. J. Lombardía, I. Molina, D. Morales, and L. Santamaría. Estimation of the mean squared error of predictors of small area linear parameters under a logistic mixed model. *Computational Statistics & Data Analysis*, 51:2720–2733, 2007. [p74]

W. González-Manteiga, M. J. Lombardía, I. Molina, D. Morales, and L. Santamaría. Bootstrap mean squared error of small-area eblup. *Journal of Statistical Computation and Simulation*, 78:443–462, 2008. [p74]

C. R. Henderson. Estimation of genetic parameters. *Biometrics*, 6(2):186–187, 1950. [p67, 70]

T. Hobza and D. Morales. Empirical best prediction under unit-level logit mixed models. *Journal of official statistics*, 32(3):661–692, 2016. [p67, 70]

J. Jiang. Reml estimation: asymptotic behavior and related topics. *The Annals of Statistics*, 24(1):255–286, 1996. [p70]

R. N. Kackar and D. A. Harville. Unbiasedness of two-stage estimation and prediction procedures for mixed linear models. *Communications in statistics-theory and methods*, 10(13):1249–1261, 1981. [p70]

A.-K. Kreutzmann, S. Pannier, N. Rojas-Perilla, T. Schmid, M. Templ, and N. Tzavidis. The r package emdi for estimating and mapping regionally disaggregated indicators. *Journal of Statistical Software*, 91, 2019. [p67]

M. D. E. Lefler, D. M. Gonzalez, and A. P. Martin. *saery: Small Area Estimation for Rao and Yu Model*, 2014. URL `https://CRAN.R-project.org/package=saery`. R package version 1.0. [p67]

C. Lin, A. Gelman, P. N. Price, and D. H. Krantz. Analysis of local decisions using hierarchical modeling, applied to home radon measurement and remediation. *Statistical Science*, 14(3):305–337, 1999. [p72]

A. Loy. *Diagnostics for mixed/hierarchical linear models*. PhD thesis, Iowa State University, 2013. [p72]

A. Loy and H. Hofmann. HLMdiag: A suite of diagnostics for hierarchical linear models in R. *Journal of Statistical Software*, 56(5):1–28, 2014. URL `https://www.jstatsoft.org/article/view/v056i05`. [p71]

A. Loy and H. Hofmann. Are you normal? the problem of confounded residual structures in hierarchical linear models. *Journal of Computational and Graphical Statistics*, 24(4):1191–1209, 2015. [p72]

A. Loy, H. Hofmann, and D. Cook. Model choice and diagnostics for linear mixed-effects models using statistics on street corners. *Journal of Computational and Graphical Statistics*, 26(3):478–492, 2017. [p72]

I. Molina and Y. Marhuenda. sae: An R package for small area estimation. *The R Journal*, 7(1):81–98, jun 2015. URL `https://journal.r-project.org/archive/2015/RJ-2015-007/RJ-2015-007.pdf`. [p67]

I. Molina and J. Rao. Small area estimation of poverty indicators. *Canadian Journal of Statistics*, 38(3): 369–385, 2010. [p67, 70, 73]

A. Nero, S. Leiden, D. Nolan, P. Price, S. Rein, K. Revzan, H. Woolenberg, and A. Gadgil. Statistically based methodologies for mapping of radon'actual'concentrations: the case of minnesota. *Radiation Protection Dosimetry*, 56(1-4):215–219, 1994. [p71]

N. Permatasari and A. Ubaidillah. *msae: Multivariate Fay Herriot Models for Small Area Estimation*, 2021. URL https://CRAN.R-project.org/package=msae. R package version 0.1.4. [p67]

P. N. Price and A. Gelman. Should you measure the radon concentration in your home? In *Statistics: A Guide to the Unknown*, pages 149–170. Duxbury Press, Belmont, CA, 4th edition edition, Mar. 2005. ISBN 978-0-534-37282-8. [p72]

P. N. Price, A. V. Nero, and A. Gelman. Bayesian prediction of mean indoor radon concentrations for minnesota counties. *Health Physics*, 71(6):922–936, 1996. [p72]

J. N. Rao and I. Molina. *Small area estimation*. John Wiley & Sons, 2015. [p69, 73]

J. N. Rao and M. Yu. Small-area estimation by combining time-series and cross-sectional data. *Canadian Journal of Statistics*, 22(4):511–528, 1994. [p67]

R. M. Royall. The linear least-squares prediction approach to two-stage sampling. *Journal of the American Statistical Association*, 71(355):657–664, 1976. [p67, 70]

A. Wolny-Dominiak and T. Żądło. On bootstrap estimators of some prediction accuracy measures of loss reserves in a non-life insurance company. *Communications in Statistics-Simulation and Computation*, pages 1–16, 2020. [p67, 68]

A. Wolny-Dominiak and T. Żądło. *qape: Quantile of Absolute Prediction Errors*, 2023. URL https://CRAN.R-project.org/package=qape. R package version 2.0. [p71, 74, 80]

T. Żądło. On parametric bootstrap and alternatives of mse. In *Proceedings of 31st International Conference Mathematical Methods in Economics*, pages 1081–1086, 2013. [p67, 68]

T. Żądło. On prediction of population and subpopulation characteristics for future periods. *Communications in Statistics-Simulation and Computation*, 461(10):8086–8104, 2017. [p70]

Alicja Wolny–Dominiak
Department of Statistical and Mathematical Methods in Economics
University of Economics in Katowice
50, 1 Maja Street
40–287 Katowice
Poland
alicja.wolny-dominiak@uekat.pl
web.ue.katowice.pl/woali/

Tomasz Żądło
Department of Statistics, Econometrics and Mathematics
University of Economics in Katowice
50, 1 Maja Street
40–287 Katowice
Poland
tomasz.zadlo@uekat.pl
web.ue.katowice.pl/zadlo/