Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

nonprobsvy – An R package for modern methods for non-probability surveys

Łukasz Chrostowski
Adam Mickiewicz University

Piotr Chlebicki
Stockholm University

Maciej Beręsewicz
Poznań University of Economics and Business
Statistical Office in Poznań

Abstract

The paper presents the **nonprobsvy** package which implements the state-of-the-art statistical inference methods for non-probability samples. The package implements various approaches that can be categorized into three groups: prediction-based approach, inverse probability weighting and doubly robust approach. On the contrary to the existing packages **nonprobsvy** assumes existance of either full population or probability-based population information and laverage the **survey** package for the inference. The package implements both analytical and bootstrap variance estimation for all of the proposed estimators. In the paper we present the theory behind the package, its functionalities and case study that showcases the usage of the package. The package is aimed at official statisticans, public opinion or market researchers who whould like to use non-probability samples (e.g. big data, opt-in web panels, social media) to accurately estimate popualtion characteristics.

Keywords: data integration, doubly robust estimation, propensity score estimation, mass imputation, R, Python, **survey**.

1. Introduction

In official statistics, information about the target population and its characteristics is mainly collected through probability surveys, census or is obtained from administrative registers, which covers all (or nearly all) units of the population. However, owing to increasing non-response rates, particularly unit non-response and non-contact, resulting from the growing

respondent burden, as well as rising costs of surveys conducted by National Statistical Institutes (NSIs), non-probability data sources are becoming more popular (Beręsewicz 2017; Beaumont 2020). Non-probability surveys, such as opt-in web panels, social media, scanner data, mobile phone data or voluntary register data, are currently being explored for use in the production of official statistics (Citro 2014; Daas, Puts, Buelens, and Hurk 2015), public opinion studies or market research. Since the selection mechanism in these sources is unknown, standard design-based inference methods cannot be directly applied.

Table 1 that compares the basic characteristics of probability and non-probability samples. In particular, what are the advantages and disadvantages of each type of sample with respect to population coverage, bias, variance, costs, and the selection mechanism for observations into the samples. In general, non-probability samples suffers from unknown selection mechanism (i.e. unknown probabilities of inclusion into sample) and under-coverage of certain groups from the population. As a result direct estimation based on these samples are characterised with bias and, in most cases, small variance (for large non-probability surveys) which leads to so called big data paradox. Certainly, cost and timeliness of these surveys is significantly smaller than for non-probability samples.

Factor	**Probability sample**	**Non-probability sample**
Selection	Known probabilities	Unknown self-selection
Coverage	Complete	Incomplete
Estimation bias	Unbiased under design	Potential systematic bias
Variance of estimates	Typically high	Typically low
Cost	High	Low
Availability	Long	Rapid

Table 1: Comparison of probability and non-probability samples and its characteristics

To address this problem, several approaches based on inverse probability weighting (IPW), model-based prediction / mass imputation (MI) and doubly robust (DR) estimators have been proposed for two main scenarios: 1) population-level data are available, either in the form of unit-level data (e.g. from a~register covering the whole population) or known population totals/means, and 2) only survey data are available as a~source of information about the target population (cf. Elliott and Valliant 2017). Wu (2022) classified these approaches into three groups that require a~joint randomization framework involving p (probability sampling design) and one of the outcome regression model ξ or propensity score model q. In this approach the IPW estimator is under the qp framework, the MI estimator is under the ξp framework, DR is under the qp or ξp framework.

Most approaches assume that population data are used to reduce the bias of non-probability sampling by a proper reweighting to reproduce known population totals/means; by modelling $E(Y|\boldsymbol{X})$ using various techniques; or combining both approaches (for instance doubly robust estimators, cf. Chen, Li, and Wu (2020); Multilevel Regression and Post-stratification (MRP) also called *Mister-P*, cf. Gelman (1997)). Majority of these methods rely on a limited number of moments of continuous or count data, with some exceptions. For example, non-parametric approaches based on nearest neighbours (NN), such as those discussed by Yang, Kim, and Hwang (2021), kernel density estimation (KDE) described by Chen, Yang and Kim Chen, Yang, and Kim (2022) or quantile balanced IPW by Beręsewicz, Szymkowiak, and Chlebicki (2025) have also been proposed. It should be highlighted that, on contrary to probability

samples, there is no single method that can be used for non-probability samples. Literature, and thus statistical software, offers various methods as presented in the next section.

1.1. Software for non-probability samples

Table 2 presents comparison of availability of various inference methods and functionalities of selected packages. We focused on packages available through CRAN or PyPI (for non-CRAN packages see Cobo, Ferri-García, Rueda-Sánchez, and Rueda (2024)). In the comparison we included four packages that focuses on non-probability samples: **NonProbEst** (Rueda, Ferri-García, and Castro 2020), **balance** (Sarig, Galili, and Eilat 2023), **inps** (Castro Martín 2024) and our **nonprobsvy** as well as two packages that implements specific, methods: **rstanarm** (MRP; Goodrich, Gabry, Ali, and Brilleman (2024)) and **GJRM** (sample selection models from econometrics; Marra and Rodicw (2023)). The **NonProbEst** implements IPW under logistic regression as well as machine learning approaches (e.g. CART, GBM or Neural Networks) however does not provide theoretical justification of using the ML approaches. The **balance** focuses solely on the IPW and uses standard and covariate balancing propensity score.

Functionalities	**NonProbEst**	**rstanarm**	**GJRM**	**balance**	**inps**	**nonprobsvy**
IPW	✓	–	+/-	✓	✓	✓
Calibrated IPW	–	–	–	–	–	✓
MI (population)	✓	–	–	–	–	✓
MI (sample)	–	–	–	–	–	✓
DR	–	–	–	–	✓	✓
MRP	–	✓	–	–	–	–
Sample selection	–	–	✓	–	–	–
Variable selection	✓	–	–	✓	–	✓
Analytical variance	–	–	–	–	–	✓
Bootstrap variance	✓	–	–	–	–	✓
Integration with **survey** or **samplics**	–	–	–	–	–	✓

Table 2: Comparison of inference methods an of different packages

It should be however noted that implementation of specific approaches may vary significantly as the documentation of specific modules is limited. For instance, **NonProbEst** allows for calibration of IPW weights *after* they are estimated, while **nonprobsvy** does this in one step. On the other hand **NonProbEst** implements various weights based on the estimated propensity scores while **nonprobsvy** supports only two types of IPW weights. Furthermore, it should be noted that **NonProbEst**, **balance** and **inps** implements survey calibration for non-probability sample but it is not theoretically justified as the inclusion probabilities are unknown (on the contrary to probability samples). Only the **nonprobsvy** package leveraged the use of the **survey** package to estimate analytic as well as bootstrap variance of the non-probability estimators. To our knowledge the **nonprobsvy** is the solely software (open or close) that implements state-of-the-art methods for non-probability samples.

The remaining part of the paper is as follows. In Section 2 theory of the statistical inference based on non-probability samples is presented. We provide basic set-up and introduce methods in separate subsections. Section 3 describes the dependencies of the package and the main function with its arguments. Section 4 presents a case study of integration of the

Polish Job Vacancy Survey with a voluntary admin data: Central Job Offers Database with an aim on estimating number of companies with at least vacancy offered on a single shift. Section 5 presents classes and the `S3Methods` implemented in the package. Paper finishes with summary and plans for the future works. In the Appendix we present codes for specific approaches discussed in the paper, algorithms and detailed derivations of the implemented methods.

2. Methods for non-probability samples

2.1. Basic setup

Let $U = \{1, ..., N\}$ denote the target population consisting of N labelled units. Each unit i has an associated vector of auxiliary variables $\boldsymbol{x}_i$ (a realisation of the random vector $\boldsymbol{X}_i$ in the super-population) and the study variable y_i (a realisation of the random variable Y_i in the super-population). Let $\{(y_i, \boldsymbol{x}_i), i \in S_A\}$ be a dataset of a non-probability sample S_A of size n_A and let $\{(\boldsymbol{x}_i, \pi_i), i \in S_B\}$ be a dataset of a probability sample S_B of size n_B, where only information about variables $\boldsymbol{X}$ and inclusion probabilities π (which in the super population model are also considered to be random variables) are available. Let R_i be an indicator of inclusion into non-probability sample S_A. Each unit in the sample S_B has been assigned a design-based weight given by $d_i = 1/\pi_i$. The above description of the data is presented in a more concise form in Table 3.

Sample	ID	Sample weight $d = \pi^{-1}$	Covariates $\boldsymbol{x}$	Study variable y
Non-probability sample (S_A)	1	?	✓	✓
	⋮	?	⋮	⋮
	n_A	?	✓	✓
Probability sample (S_B)	1	✓	✓	?
	⋮	⋮	⋮	?
	n_B	✓	✓	?

Table 3: Two sample setting.

The goal is to estimate a finite population mean $\mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i$ of the target variable Y. As values of y_i are not observed in the probability sample, it cannot be used to estimate the target quantity. Instead, one could try combining the non-probability and probability samples to estimate μ_y. As there is no single method how to achieve this assumptions vary significantly as summarised by Wu (2022). The main assumptions that apply to all methods are: no overlap between samples and no measurement error in Y_i and $\boldsymbol{X}_i$ is observed. In the next sections we briefly present methods that are implemented in the package.

2.2. Prediction-based approach

Mass imputation

Imputation refers to the process of replacing missing or incomplete data with substituted

values. The goal of imputation is to allow for more complete data analysis, as many statistical methods require complete datasets.

Mass imputation is the application of imputation techniques to an entire dataset where many observations have missing values for the given variable. Kim, Park, Chen, and Wu (2021), Yang *et al.* (2021), Chlebicki, Chrostowski, and Beręsewicz (2024) propose the following imputation strategies as:

- Model based approach (GLM),
- Nearest neighbour imputation (NN),
- Predictive mean matching (PMM).

Mass imputation is particularly useful in large datasets where missing data can be widespread, and it seeks to preserve the relationships between variables, thus improving the overall integrity of the data.

As presented in (Table 3), we do not know the value of the dependent variable Y for the units in the probability sample. In this case, the method will be to impute the values of the explanatory variable for all units in the probability sample. We therefore treat the non-probability sample as a training set that is used to build the imputation model. In this subsection, we distinguish three main methods of mass imputation based on linear models and the k-nearest neighbours algorithm. Other popular methods for estimating the variable Y from the variable $\boldsymbol{X}$ can also be considered, e.g. machine learning models such as random forests or neural networks.

We can obtain an estimate of the population mean based on known design weights and an imputation model for units from the probability sample:

$$\hat{\mu}_{MI} = \frac{1}{\hat{N}_{\mathrm{B}}} \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} \hat{y}, \tag{1}$$

$\hat{N}_{\mathrm{B}} = \sum_{i \in S_B} d_i^B$ and $\hat{y}$ is the estimated value of y for units from probability samples based on mass imputation model.

This estimator can be understood as a version of the Horvitz–Thompson estimator, which are used to estimate mean or total values in the population (based on probability sampling and inclusion probabilities). The only difference is that in our case, instead of the known values of the Y variable, we use its estimated equivalents.

Generalized Linear Models

Let us assume the following parametric model for the sample S_A based on the conditional expected value of the variable Y. Let

$$\mathbb{E}\left(y_i \mid \boldsymbol{x}_i\right) = m\left(\boldsymbol{x}_i, \boldsymbol{\beta}_0\right) \tag{2}$$

for a certain p–dimensional vector $\boldsymbol{\beta}_0$ and a known m function from a given class of mean functions for generalized linear models.

According to the model described, we have

$$y_i = m\left(\boldsymbol{x}_i\right) + \varepsilon_i, \quad i = 1, 2, \ldots, N.$$

We also assume that the random variables ε_i are independent with $\mathbb{E}(\varepsilon_i) = 0$ and $\sigma^2(\varepsilon_i) = \mathbf{v}(\boldsymbol{x})\sigma^2$. It is assumed that $\mathbf{v}(\boldsymbol{x})$ has a known value and is homogeneous, i.e. homogeneous, regardless of the sample under study. Let us represent the process of mass imputation of a linear model to a sample S_B. Finally we are interested in finding a vector $\boldsymbol{\beta}$ solves the following equation:

$$U(\boldsymbol{\beta}) = \frac{1}{n_A} \sum_{i \in S_A} \{y_i - m(\boldsymbol{x}_i; \boldsymbol{\beta})\} h(\boldsymbol{x}_i; \boldsymbol{\beta}) = \mathbf{0}, \tag{3}$$

for some p-dimensional vector of function $h(\boldsymbol{x}_i; \boldsymbol{\beta})$, where $h(\boldsymbol{x}_i; \boldsymbol{\beta}) = \boldsymbol{x}_i$ might be used for certain applications. The mass imputation process is described in Algorithm 1.

Nearest Neighbour Algorithm

On the other hand, it is also possible to consider a non-parametric model for the problem described, i.e. for each individual from sample S_B, the k-nearest neighbours from sample S_A are found based on the values of the auxiliary vector $\boldsymbol{X}$ and the corresponding metric. Then, the missing values of the variable Y from sample S_B are replaced by the values (or their mean if more than one neighbour is considered) of this variable for the corresponding neighbours from sample S_A. The algorithm is as follows

Note that the algorithm differs depending on the number of nearest neighbours chosen. In case $k = 1$ the nearest neighbour value is imputed according to the chosen metric, for example the Euclidean metric. In case $k > 2$ the average of the nearest neighbours values is imputed. The literature indicates that this method suffers from the so-called curse of multidimensionality, i.e. for samples with several explanatory variables, imputation can lead to a large variance in the estimator. On the other hand, the algorithm is easy to interpret and simple to implement.

Predictive Mean Matching

Predictive mean-matching imputation is a particularly well-known way of dealing with non-response among respondents, and is favoured by statistical offices for compiling a country's official population statistics. It is a version of the k-nearest neighbour algorithm, but instead of looking at the distances between the vectors of the auxiliary variables, it looks at the distance between the functions of the mean vectors. This helps to reduce the curse of multidimensionality and, at the same time, allows the observed values of the explanatory variable or their mean to be calculated. Let us therefore present two algorithms that describe the steps to follow to perform a mass imputation using the mean matching method.

As can be seen, the difference between the two algorithms is due to step 2. In the first approach, we compare $\hat{y}$ from samples S_A and S_B. The second, on the other hand, compares $\hat{y}$ from sample S_B with the known y from sample S_A. It is worth noting that proof of the consistency of these estimators can be found in Chlebicki *et al.* (2024).

2.3. Inverse Probability Weighting

The main disadvantage of non-probability sampling is the unknown selection mechanism for a unit to be included in the sample. This is why we talk about the so-called biased sample problem. The inverse probability approach is based on the assumption that a reference probability sample is available and therefore we can estimate the propensity score of the selection

mechanism. In recent years, a number of articles have addressed this issue. Chen *et al.* (2020) propose maximum likelihood estimation approach for estimating propensity scores for selection mechanism. Wu (2022) present the approach based on generalized estimating equations, this method is also mentioned in Yang, Kim, and Song (2020). On the other hand calibration approach for quantiles was explained Beręsewicz and Szymkowiak (2024) and Sant'Anna, Song, and Xu (2022) present the approach based on maximize the covariate distribution balance among different treatment groups.

In the formal framework, let us introduce the following assumptions for propensity score model, which will imply a number of properties derived in the thesis.

(A1) The selection indicator R_i^A and explanatory variable y_i are independent.

(A2) All units have a so-called non-probability sample propensity score, which is non-zero, i.e. $\pi_i^{\mathrm{A}} > 0$, where $\pi_i^{\mathrm{A}} = P_q\left(R_i^A = 1 \mid \boldsymbol{x}_i, y_i\right)$, where q refers to the model for the selection mechanism for the non-probability sample (propensity score model).

(A3) Indicator variables R_i^A and R_j^A are independent with $i \neq j$.

The estimated propensity score is used to construct an inverse probability weighting estimator of the population mean of the form

$$\hat{\mu}_{IPW} = \frac{1}{\hat{N}^A} \sum_{i \in S_A} \frac{y_i}{\hat{\pi}_i^A}. \tag{4}$$

where $\hat{N}^A = \sum_{i \in S_A} \hat{d}_i^A = \sum_{i \in S_A} \frac{1}{\hat{\pi}_i^A}$.

Maximum Likelihood Estimation

Consider the following likelihood function

$$\begin{aligned} \ell(\boldsymbol{\theta}) &= \sum_{i=1}^{N} \left\{ R_i^A \log \pi_i^{\mathrm{A}} + \left(1 - R_i^A\right) \log\left(1 - \pi_i^{\mathrm{A}}\right) \right\} \\ &= \sum_{i \in S_{\mathrm{A}}} \log\left\{ \frac{\pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)}{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)} \right\} + \sum_{i=1}^{N} \log\left\{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\right\} \end{aligned} \tag{5}$$

In practice, a function of this form cannot be used because we do not observe all units from the population. Hence, the second component of the function is replaced by the Horvitz-Thompson estimator, which is used when having access to the design weights for the units in the sample. In our case, these will be the weights d_i^B for the units in the sample S_B. We then have

$$\ell^*(\boldsymbol{\theta}) = \sum_{i \in S_{\mathrm{A}}} \log\left\{ \frac{\pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)}{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)} \right\} + \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} \log\left\{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\right\}. \tag{6}$$

Our objective is to find the maximum likelihood estimator $\hat{\pi}_i^A = \pi(\boldsymbol{x}_i, \hat{\boldsymbol{\theta}})$, such that $\hat{\boldsymbol{\theta}}$ maximises the function defined above.

Generalized Estimating Equations

Equations of the type $\mathrm{U}(\boldsymbol{\theta}) = \mathbf{0}$, where $\mathrm{U}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} l^*(\boldsymbol{\theta})$, obtained from the maximum likelihood estimation can be replaced by a system of generalized estimating equations of the form

$$\mathbf{G}(\boldsymbol{\theta}) = \sum_{i \in S_A} h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) - \sum_{i \in S_B} d_i^B \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \mathbf{0}, \tag{7}$$

where $h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)$ is a certain continuous function. In the literature, the most commonly considered functions are $h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \boldsymbol{x}_i$ and $h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \boldsymbol{x}_i \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^{-1}$. Note that if the function h is equal to the vector of observed characteristics $\boldsymbol{x}$, then $\mathbf{G}$ is reduced to

$$\mathbf{G}(\boldsymbol{\theta}) = \sum_{i \in S_A} \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) \boldsymbol{x}_i.$$

In the next subsection we will proof that this is disorted version of MLE approach with π_i^A modelling by logistic regression. If we use the second form of the function h we get the following form of the function $\mathbf{G}$

$$\mathbf{G}(\boldsymbol{\theta}) = \sum_{i \in S_A} \frac{\boldsymbol{x}_i}{\pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)} - \sum_{i \in S_B} d_i^B \boldsymbol{x}_i.$$

The advantage of this method is the ability to estimate with global values of the variables (e.g. from external sources) instead of a probability sample. Note that this is allowed by the second form of the $\mathbf{G}$ function. Its second term is nothing more than the estimated sums of the $\boldsymbol{x}$ variables. On the other hand, empirical studies suggest that the process of solving this type of equation may be less stable than the maximum likelihood method. In other words, the iterative algorithm for finding zeros that satisfy equation (7) may not converge.

In the `nonprobsvy` package the propensity scores can modelled using three different link functions: logistic, complementary log-log and probit. The logistic regression is the most commonly used for modelling probabilities. This method is based on the so-called sigmoidal function and for our settign it has the form It satisfies certain properties to model the probability. For our scheme, this will be the probability of belonging to the non-probability sample. Another approach to modelling binary variables and also probabilities is probit regression. It is based on the standard normal distribution and as logit is also symmetric around $p = 0.5$. On the other hand regression with the cloglog model is particularly useful if we are modelling rare phenomena, i.e. the probabilities will oscillate around the values 1 and 0. Compared to the sigmoidal function and the distribution, the cloglog function is more asymmetric towards the value 0.5.

2.4. Doubly Robust approach

The inverse probability weighting and mass imputation estimators are sensible on misspecified models for propensity score and outcome variable respectively. For this purpose so called doubly-robust methods, which take into account these problems, are presented.

The proposed estimation procedure addresses the challenge of combining data from nonprobability and probability survey samples. Traditional semiparametric models, often applied to such problems, are not directly usable in this context due to the distinct nature of the two samples. Instead, a joint randomization framework is employed, integrating semiparametric models for propensity scores with outcome regression for the nonprobability sample and

design-based inference from the probability sample. This framework leads to a doubly robust (DR) estimation approach, which is effective in the presence of model misspecifications.

Inverse Probability Weighted (IPW) estimators are sensitive to misspecified propensity score models, particularly when propensity scores are very small. To improve robustness and efficiency, the doubly robust method incorporates a prediction model for the response variable. Moreover, even if one of the models is misspecified, the DR estimator remains consistent, showcasing the "double robustness'' property.

Joint Randomization Approach

The joint randomization approach combines two processes: the selection mechanism of a non-probability sample, modelled by propensity scores, and the design-based inference from a probability sample.

The response y_i is predicted using a regression model $m(\boldsymbol{x}_i, \boldsymbol{\beta})$ (or NN/PMM methods), where $\boldsymbol{\beta}$ is estimated from the non-probability sample. With known design weights d_i^B for $i \in S_B$ we can define the DR estimator as

$$\hat{\mu}_{\mathrm{DR}} = \frac{1}{\hat{N}^{\mathrm{A}}} \sum_{i \in S_{\mathrm{A}}} d_i^{\mathrm{A}} \left\{ y_i - m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right) \right\} + \frac{1}{\hat{N}^{\mathrm{B}}} \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right), \tag{8}$$

where $d_i^A = \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^{-1}, \hat{N}^A = \sum_{i \in S_A} d_i^A$ and $\hat{N}^B = \sum_{i \in S_B} d_i^B$.

It remains consistent if either the propensity score model $\pi(\boldsymbol{x}_i, \boldsymbol{\theta})$ or the outcome regression model $m(\boldsymbol{x}_i, \boldsymbol{\beta})$ is correctly specified.

The joint randomization approach ensures robustness by accounting for randomness in both the non-probability sample through $\pi(\boldsymbol{x}_i, \boldsymbol{\theta})$ and the probability sample through design-based inference.

Minimization of the bias for doubly robust methods

By reducing the variance of the estimators, for example by variable selection, we cannot control the bias of the estimator, which may increase. Therefore, according to Yang *et al.* (2020), the idea is to determine the equations leading to the estimation of the $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ parameters based on the bias of the population mean estimator. In contrast to the joint randomization approach, this method allows for the estimation of the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ in a single step, rather than in two separate steps.

We will first present the bias of the doubly robust estimator and then, using optimisation techniques, discuss the equations leading to its minimization. Thus we have

$$\begin{aligned} \operatorname{bias}\left(\hat{\mu}_{DR}\right) =& \mid \hat{\mu}_{DR} - \mu \mid \\ &= \frac{1}{N} \sum_{i=1}^{N} \left\{ \frac{R_i^A}{\pi_i^A\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\theta}\right)} - 1 \right\} \left\{ y_i - m\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta}\right) \right\} \\ &+ \frac{1}{N} \sum_{i=1}^{N} \left(R_i^B d_i^B - 1 \right) m\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta}\right) \end{aligned} \tag{9}$$

To minimize $\operatorname{bias}\left(\hat{\mu}_{DR}\right)^2$ let us calculate the gradient of the square of the bias at $(\boldsymbol{\beta}, \boldsymbol{\theta})$. We then have

$$\frac{\partial \operatorname{bias}\left(\hat{\mu}_{DR}\right)^2}{\partial \left(\boldsymbol{\beta}^{\mathrm{T}}, \boldsymbol{\theta}^{\mathrm{T}}\right)^{\mathrm{T}}} = 2 \operatorname{bias}\left(\hat{\mu}_{DR}\right) J(\theta, \beta),$$

where

$$J(\theta, \beta) = \begin{pmatrix} J_1(\theta, \beta) \\ J_2(\theta, \beta) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N} R_i^A \left\{ \frac{1}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} - 1 \right\} \left\{ y_i - m\left(\boldsymbol{x}_i, \boldsymbol{\beta}\right) \right\} \boldsymbol{x}_i \\ \sum_{i=1}^{N} \frac{R_i^A}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} \frac{\partial m(\boldsymbol{x}_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} - \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} \frac{\partial m(\boldsymbol{x}_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \end{pmatrix},$$

which leads to the problem of solving the following system of equations

$$\begin{pmatrix} \sum_{i=1}^{N} R_i^A \left\{ \frac{1}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} - 1 \right\} \left\{ y_i - m\left(\boldsymbol{x}_i, \boldsymbol{\beta}\right) \right\} \boldsymbol{x}_i \\ \sum_{i=1}^{N} \frac{R_i^A}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} \frac{\partial m(\boldsymbol{x}_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} - \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} \frac{\partial m(\boldsymbol{x}_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \end{pmatrix} = \mathbf{0}, \tag{10}$$

which can be solved using Newton–Raphson optimization method.

2.5. Variable selection algorithms

When dealing with multivariate data with a large number of features, it is recommended to select variables for estimation that are statistically significant for the model under consideration. Yang and Kim (2020) point that using variable selection techniques during estimation is crucial, especially when dealing with high-dimensional data. Variable selection not only improves model stability and computational feasibility, but also reduces variance, which can increase when irrelevant auxiliary variables are included. Including irrelevant variables increases the complexity of the model and makes the estimation process more error-prone and unstable. Therefore, variable selection is key to ensure robust and efficient estimation. Very popular in the statistical literature are variable selection methods such as *Least Absolute Shrinkage and Selection Operator* (**LASSO**), *Smoothly Clipped Absolute Deviation* (**SCAD**) or *Minimax Concave Penalty* (**MCP**), which, thanks to appropriate loss functions, degenerate the coefficients in variables that have no significant effect on the dependent variable. In this way, the result obtained, for example, a linear regression equation, is based only on the features selected by the model. The selection procedure works in a similar way for non-probability methods using external data sources such as sample or population totals or averages. In particular, the technique is divided into two steps. In the first, we select the relevant variables using an appropriately constructed loss function and the estimating equations used. In the second case, we construct the equations on the basis of the derived biases of the relevant estimator, whose value we calculate only on the basis of the selected characteristics.

In the case of data integration based on probability and nonprobability samples, the selection of variables is part of a two-step process leading to the estimation of the mean, where in the first step statistically significant variables are selected and in the second step the model is rebuilt. For the first step, a penalized logistic regression model has been proposed to estimate propensity scores (Yang *et al.* (2020)), but this approach can be extended to other linking models such as clog-log and probit functions. For a parametric model-based mass imputation, penalized OLS (*Ordinary Least Squared*) is considered. It is worth mentioning that Yang and Kim (2020), in their article on this topic, used the SCAD method (*Smoothly Clipped Absolute Deviation*), but we extend it on other selection techniques such as LASSO and MCP.

3. The main function

3.1. The nonprob function

The **nonprobsvy** package is built around the `nonprob` function. The main design objective was to make using `nonprob` as similar as possible to standard R functions for fitting statistical models, such as `stats::glm`, while incorporating survey design features from the **survey** package. The most important arguments are given in Table 4 and the obligatory ones are `data`, while `selection`, `outcome`, or `target` must be specified depending on the chosen method.

Argument	Description
`data`	`data.frame` with data from the non-probability sample
`selection`	`formula` for the selection (propensity) equation
`outcome`	`formula` for the outcome equation
`target`	`formula` with target variables
`svydesign`	Optional `svydesign2` object
`pop_totals`, `pop_means`, `pop_size`	Optional named `vector` with population totals or means of the covariates and population size
`method_selection`	Link function for the IPW approach (`"logit"`, `"probit"`, `"cloglog"`)
`method_outcome`	Specification of the MI approach (one of `c("glm", "nn", "pmm")`)
`family_outcome`	The GLM family for the MI approach (one of `c("gaussian", "binomial", "poisson")`)
`subset`	Optional `vector` specifying a subset of observations to be used in the fitting process
`strata`	Optional `vector` specifying strata
`weights`	Optional `vector` of prior weights to be used in the fitting process
`na_action`	`function` indicating what should happen when the data contain `NA`'s
`control_selection`, `control_outcome`, `control_inference`	Control parameters for selection and outcome model and the variance estimation via the `controlSel`, `controlOut` and `controlInf` functions respectively
`start_selection`, `start_outcome`	Optional `vector` with starting values for the parameters of the selection and the outcome equation
`verbose`	Logical value indicating if verbose output should be printed
`se`	Logical value indicating whether to calculate and return the standard error of the estimated mean
`...`	Additional optional arguments

Table 4: `nonprob` function arguments description

The `nonprob` function is used specify inference methods through specifying `selection` and `outcome` arguments. If out of these two `selection` is specified than the IPW estimators are used, if only the `outcome` then the MI approach is used and if both are specified the DR approach is applied. The package allows to provide either reference population data (via the `pop_totals`, or `pop_means` and `pop_size`) or a probability sample declared by the `svydesign` argument (`svydesign2` class of from the **survey** package). Selection of the speficic

inference method is done through `method_selection`, `method_outcome`, `family_outcome`, `control_selection` and `control_outcome` arguments. Specification of variance estimation method is done via the `control_inference` argument.

In addition to using the survey package for design-based inference when probability samples are available, it also supports the various methods for estimating propensity scores and outcome models described in this thesis, such as logistic regression, complementary log-log models, probit models, generalized linear models, nearest neighbour algorithms and predictive mean matching.

Resulting object of class `nonprobsvy` is a list that contains the following (most important) elements:

- `data` – an `data.frame` containing non-probability sample.
- `X` – a `matrix` containing both samples,
- `y` – a list containing all variables declared in either `target` or `outcome` arguments,
- `R` – a numeric `vector` informing about inclusion into non-probability sample,
- `weights` – propensity score weights or NULL (for the MI estimators),
- `output` - a `data.frame` containing point and standard error estimates,
- `outcome` – a `list` of results for each `outcome` models,
- `selection` – a `list` of results for the `selection` model.
- `svydesign` – a `svydesign2` object passed by the `svydesign` argument.

After this neat description of the main functionality of the package, we will move on to some examples of its use. We will show how to define the given arguments in order to obtain estimates of interest as a result. We will be less interested in the results than in the way they are presented. There will be room in the following chapters for an analysis of simulations and applications of the package to the real world. We will focus on the three main estimators, as function calls for other functionalities such as variable selection, other linking functions or mass imputation methods.

3.2. Controlling inference methods and the variance estimation

We provide three control function that allow users to specify the exact inference methods and the variance estimation. The `controlSel` function provides essential control parameters for fitting the selection model in the `nonprob` function. It allows users to select between the MLE or GEE (calibrated) approach through `est_method_sel` (and the type with the `h` argument), specify the optimizer (`optimizer`) and the which variable selection should be applied (using different penalty functions like SCAD, lasso, and MCP through `penalty`) along with parameters (e.g. number of folds via the `nfolds` argument). The package uses the **nleqslv** package and fitting parameters (arguments starting with the `nleqslv*`).

The `controlOut` to fine-tune various aspects of the estimation process, including the variable selection methods (through different penalty options like SCAD, LASSO, and MCP with

their respective tuning parameters), and detailed configuration for NN and PMM approaches (using parameters like `predictive_match`, `pmm_weights`, and `pmm_k_choice`).

Finally, the `controlInf` function configures the parameters for variance estimation in the `nonprob` function. It allows user to specify whether the analytical or bootstrap approach should be used (the `var_method` argument), whether the variable selection should be applied (the `vars_selection` argument) and what type of boostrap should be applied for the probability sample (the `rep_type` argument). This function also allow to specify the how the inference for the DR approach should be taken: if a union or a division of variables after variable selection was applied (the `bias_inf` argument) and if the bias correction should be applied (the `bias_correction` argument).

In the next sections we present a case study on integration of non-probability sample with a reference probability sample. We will present various estimators and compare them. Finally, we present more advanced options of the package.

4. Data analysis example

4.1. Description of the data

Before we explain the case study let's first load the package.

```
R> library(nonprobsvy) ## for estimation
R> library(ggplot2) ## for visualisation
```

The goal of the case study to integrate administrative (`admin`) and survey (`jvs`) data about job vacancies in Poland.

The first source is the Job Vacancy Survey (JVS) with a sample of 6,523 units. The survey is based on a probability sample drawn according to proportional-to-size stratified sampling design. The details regarding the survey can be found in Statistics Poland (2021). The dataset contains about The Nomenclature of Economic Activities (NACE; 14 levels, `nace` column), `region` (16 levels), sector (2 levels, `private` column), size of the entity (3 levels: Small, Medium and Large) and the final weight (i.e. design weight corrected for non-contact and non-response).

```
R> data(jvs)
R> head(jvs)
```

```
   id private size nace region weight
1 j_1       0    L    0     14      1
2 j_2       0    L    0     24      6
3 j_3       0    L  R.S     14      1
4 j_4       0    L  R.S     14      1
5 j_5       0    L  R.S     22      1
6 j_6       0    M  R.S     26      1
```

As the package leverage the **survey** package functionalities we need to define the `svydesign2` object via the `svydesign` function as presented below. The dataset does not contain the true

stratification variable we use a simplified version by specifying `~ size + nace + region` and we do not know have information on the non-response and its correction we simply assume that the `weight` is the calibrated weight that sums up to the population size.

```
R> jvs_svy <- svydesign(ids = ~ 1,
+                       weights = ~ weight,
+                       strata = ~ size + nace + region,
+                       data = jvs)
```

The second source is the Central Job Offers Database (CBOP), which is a register of all vacancies submitted to Public Employment Offices (see https://oferty.praca.gov.pl). We treat this as the *non-probability sample* because is voluntary administrative data and inclusion mechanism is unknown. This dataset was prepared in such way that the records out of scope (either by the definition of vacancy or population of entities) were excluded. The dataset contains the same variables as JVS with one additional `single_shift` which is our target variable defined as: `whether a company seeks at least one employee for a single-shift job`. The goal of this case study is to estimate *the share of companies that seeks employees for a single-shift job* in Poland in a given quarter.

```
R> data(admin)
R> head(admin)
```

```
   id private size nace region single_shift
1 j_1       0    L    P     30        FALSE
2 j_2       0    L    O     14         TRUE
3 j_3       0    L    O     04         TRUE
4 j_4       0    L    O     24         TRUE
5 j_5       0    L    O     04         TRUE
6 j_6       1    L    C     28        FALSE
```

Please note that, this paper does not aim to provide full tutorial on using non-probability samples for statistical inference. Thus, we skipped the part of aligning variables to meet the same definitions, assessing how strong is the relation between auxiliary variables, target variable and selection mechanism and distribution mis-matches between both samples. In the examples below we assume that there is no overlap between two sources and the nai"ve, reference estimate, given by a simple mean of the `single_shift` column of `admin` equals to 66.1%.

4.2. Estimation

Propensity score approach

First, we start with the IPW approach with to possible estimation methods MLE (standard) and GEE (calibrated to the estimated survey totals). We start by calling the `nonprob` function where we define the `selection` argument responsible for the formulae for the inclusion variables, the `target` argument which specifies the variable of interest `single_shift`. The rest refer to the `svydesign` object, dataset and specification of the link function (`method_selection`).

```
R> ipw_est1 <- nonprob(
+   selection = ~ region + private + nace + size,
+   target = ~ single_shift,
+   svydesign = jvs_svy,
+   data = admin,
+   method_selection = "logit" ## this is the default
+ )
```

In order to get the basic information about the estimated target quantity we can use the `print` method the object. It provides the call and the estimated mean, standard error (`SE`) and 95% confidence interval (`lower_bound` and `upper_bound`).

```
R> ipw_est1
```

```
Call:
nonprob(data = admin, selection = ~region + private + nace +
    size, target = ~single_shift, svydesign = jvs_svy, method_selection = "logit")

Estimated population mean with overall std.err and confidence interval:

                  mean          SE lower_bound upper_bound
single_shift 0.7083228 0.009436907   0.6898268   0.7268188
```

If we are interested in a detailed information about the model we can use the `summary` method.

```
R> summary(ipw_est1)
```

```
Call:
nonprob(data = admin, selection = ~region + private + nace +
    size, target = ~single_shift, svydesign = jvs_svy, method_selection = "logit")

------------------------
Estimated population mean: 0.7083 with overall std.err of: 0.009437
And std.err for nonprobability and probability samples being respectively:
0.003958 and 0.008567

95% Confidence inverval for popualtion mean:
             lower_bound upper_bound
single_shift   0.6898268   0.7268188


Based on: Inverse probability weighted method
For a population of estimate size: 52898.13
Obtained on a nonprobability sample of size: 9344
```

```
With an auxiliary probability sample of size: 6523
-------------------------

Regression coefficients:
----------------------
For glm regression on selection variable:
            Estimate Std. Error z value  P(>|z|)
(Intercept) -0.65278    0.07498  -8.706  < 2e-16 ***
region04     0.83780    0.07121  11.765  < 2e-16 ***
region06     0.19954    0.07245   2.754  0.00589 **
region08     0.10481    0.08911   1.176  0.23950
region10    -0.15756    0.06408  -2.459  0.01393 *
region12    -0.60987    0.06029 -10.115  < 2e-16 ***
region14    -0.84150    0.05419 -15.530  < 2e-16 ***
region16     0.76386    0.08660   8.821  < 2e-16 ***
region18     1.17811    0.07142  16.495  < 2e-16 ***
region20     0.22252    0.09261   2.403  0.01627 *
region22    -0.03753    0.06039  -0.621  0.53438
region24    -0.40670    0.05474  -7.430 1.09e-13 ***
region26     0.20287    0.08489   2.390  0.01685 *
region28     0.57863    0.06797   8.513  < 2e-16 ***
region30    -0.61021    0.05908 -10.328  < 2e-16 ***
region32     0.32744    0.06957   4.706 2.52e-06 ***
private      0.05899    0.05880   1.003  0.31571
naceD.E      0.77274    0.10033   7.702 1.34e-14 ***
naceF       -0.37783    0.04271  -8.847  < 2e-16 ***
naceG       -0.33370    0.03788  -8.809  < 2e-16 ***
naceH       -0.65175    0.05977 -10.904  < 2e-16 ***
naceI        0.41179    0.05726   7.191 6.41e-13 ***
naceJ       -1.42639    0.13622 -10.471  < 2e-16 ***
naceK.L      0.06171    0.07981   0.773  0.43941
naceM       -0.40678    0.06741  -6.034 1.60e-09 ***
naceN        0.80035    0.06733  11.888  < 2e-16 ***
naceO       -0.69355    0.09460  -7.331 2.28e-13 ***
naceP        1.25095    0.07647  16.359  < 2e-16 ***
naceQ        0.30287    0.06799   4.455 8.41e-06 ***
naceR.S      0.22228    0.06975   3.187  0.00144 **
sizeM       -0.36413    0.03444 -10.574  < 2e-16 ***
sizeS       -1.02916    0.03504 -29.369  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
-------------------------

Weights:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.169   2.673   4.333   5.661   7.178  49.951
-------------------------
```

```
Residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.8552 -0.2308  0.5393  0.3080  0.7987  0.9800

AIC: 43894.82
BIC: 44140.32
Log-Likelihood: -21915.41 on 15835 Degrees of freedom
```

This displays information on the datasets used for estimation (probability and non-probability sample). The estimated regression coefficients are also shown, in this case for the logit model for propensity score model (section `Regression coefficients`). In addition, for diagnostic purposes, we have access to the distribution of the weights calculated from the inclusion probabilities (section `Weights`), the distribution of the residuals from the model (section `Residuals`), as well as the values of the AIC, BIC statistics in the case of models based on MLE.

If we are interested in the calibrated IPW, one needs to define a `controlSel` function in the `control_selection` argument with the `est_method_sel` argument equal to `gee` (the default is `mle`) and set the value of `h` (as dicussed in the 2.3.2 section).

```
R> ipw_est2 <- nonprob(
+   selection = ~ region + private + nace + size,
+   target = ~ single_shift,
+   svydesign = jvs_svy,
+   data = admin,
+   method_selection = "logit",
+   control_selection = controlSel(h = 1, est_method_sel = "gee")
+ )
```

Results are comparable to the standard IPW point estimate (70.4 vs 70.8) while the standard error is lower higher.

```
R> ipw_est2


Call:
nonprob(data = admin, selection = ~region + private + nace +
    size, target = ~single_shift, svydesign = jvs_svy, method_selection = "logit",
    control_selection = controlSel(h = 1, est_method_sel = "gee"))

Estimated population mean with overall std.err and confidence interval:

                  mean         SE lower_bound upper_bound
single_shift 0.7041796 0.01169878   0.6812504   0.7271088
```

The calibrated IPW signiciantly improves the balance as can be accessed by the `nonprobsvycheck` function:

```
R> data.frame(ipw_mle=nonprobsvycheck(~size, ipw_est1, 1)$balance,
+             ipw_gee=nonprobsvycheck(~size, ipw_est2, 1)$balance)

      ipw_mle ipw_gee
sizeL  -367.6       0
sizeM  -228.4       0
sizeS  1624.1       0
```

Notice that, neither in the package nor this paper we focus a detailed description of the post-hoc results, such as covariate balance. This can be done via existing CRAN packages, for instance using the `bal.tab` function from the **cobalt** package (Greifer 2024).

Prediction-based approach

If a user is interested in a prediction-based approach, in particular mass imputation estimators, then should specify the argument `outcome` as a formulae (similarly as in the `glm` function). We allow single outcome (specified as `y ~ x1 + x2 + ... + xk`) or multiple outcomes (as `y1 + y2 + y2 ~ x1 + x2 + ... + xk`). Note that if the `outcome` argument is specified then there is no need to specify `target` argument. By default GLM type of the MI estimator is assumed (i.e. `method_outcome="glm"`). In the code below we present possible way to declare this type of the MI estimator.

```
R> mi_est1 <- nonprob(
+   outcome = single_shift ~ region + private + nace + size,
+   svydesign = jvs_svy,
+   data = admin,
+   method_outcome = "glm",
+   family_outcome = "binomial"
+ )
R>
R> mi_est1


Call:
nonprob(data = admin, outcome = single_shift ~ region + private +
    nace + size, svydesign = jvs_svy, method_outcome = "glm",
    family_outcome = "binomial")

Estimated population mean with overall std.err and confidence interval:

                  mean         SE lower_bound upper_bound
single_shift 0.7032081 0.01120231    0.681252   0.7251642
```

If a user is interested in the nearest neighbours MI estimator one can specify `method_outcome = "nn"` for the nearest neighbours search using all variables specified in the `outcome` argument, or `method_outcome = "pmm"` if is interested in predictive mean matching. In both cases we are using $k = 5$ nearest neighbours (i.e. `controlOut(k=5)`). For the NN MI estimator there

is no need to specify the `family_outcome` argument as no model is estimated underneath. For both approaches we use the **RANN** package (Jefferis, Kemp, Arya, and Mount 2024).

```
R> mi_est2 <- nonprob(
+   outcome = single_shift ~ region + private + nace + size,
+   svydesign = jvs_svy,
+   data = admin,
+   method_outcome = "nn",
+   control_outcome = controlOut(k=5)
+ )
R>
R> mi_est3 <- nonprob(
+   outcome = single_shift ~ region + private + nace + size,
+   svydesign = jvs_svy,
+   data = admin,
+   method_outcome = "pmm",
+   family_outcome = "binomial",
+   control_outcome = controlOut(k=5)
+ )
```

Results of both estimators seems to be similar, but it should be noted that the NN MI estimator suffers from the curse of dimensionality so one should trust more the PMM MI estimator.

```
R> mi_est2$output
```

```
                  mean         SE
single_shift 0.6799537 0.01575574
```

```
R> mi_est3$output
```

```
                  mean         SE
single_shift 0.7450896 0.01527651
```

As discussed in Section 2 both IPW and MI estimators are asymptotically unbiased only when the model and auxiliary variables are correctly specified. To overcome this problem we focus now on the doubly robust estimators.

The doubly robust approach

To indicate that the doubly robust estimation should used user needs to specify both the `selection` and `outcome` arguments. These formulas can be specified with the same or varying number of aixuliary variables. We also allow, simiarly as in the MI approach, multiple outcomes. In the following example code we specified the non-calibrated IPW and the GLM MI estimator.

```
R> dr_est1 <- nonprob(
+   selection = ~ region + private + nace + size,
+   outcome = single_shift ~ region + private + nace + size,
+   svydesign = jvs_svy,
+   data = admin,
+   method_selection = "logit",
+   method_outcome = "glm",
+   family_outcome = "binomial"
+ )
R> dr_est1


Call:
nonprob(data = admin, selection = ~region + private + nace +
    size, outcome = single_shift ~ region + private + nace +
    size, svydesign = jvs_svy, method_selection = "logit", method_outcome = "glm",
    family_outcome = "binomial")

Estimated population mean with overall std.err and confidence interval:

                  mean         SE lower_bound upper_bound
single_shift 0.7034644 0.01131974   0.6812781   0.7256507
```

Detailed results can be obtained by using `summary` function which prints both set of coefficients for the outcome and selection models. We omit this output due to limited space of the paper. Finally, we can use bias minimisation approach as proposed by Yang *et al.* (2020) by specifying `control_inference = controlInf(bias_correction = TRUE)` argument. This part is implemented in the **Rcpp** (Eddelbuettel, Francois, Allaire, Ushey, Kou, Russell, Ucar, Bates, and Chambers 2024) and **RcppArmadillo** (Eddelbuettel and Sanderson 2014) packages for performance.

```
R> dr_est2 <- nonprob(
+   selection = ~ region + private + nace + size,
+   outcome = single_shift ~ region + private + nace + size,
+   svydesign = jvs_svy,
+   data = admin,
+   method_selection = "logit",
+   method_outcome = "glm",
+   family_outcome = "binomial",
+   control_inference = controlInf(bias_correction = TRUE)
+ )
R> dr_est2


Call:
nonprob(data = admin, selection = ~region + private + nace +
    size, outcome = single_shift ~ region + private + nace +
```

```
    size, svydesign = jvs_svy, method_selection = "logit", method_outcome = "glm",
    family_outcome = "binomial", control_inference = controlInf(bias_correction = TRUE))

Estimated population mean with overall std.err and confidence interval:

                  mean         SE lower_bound upper_bound
single_shift 0.7043248 0.01128182   0.6822129   0.7264368
```

4.3. Comparison of estimates

Finally, as there is no single method for non-probability samples we suggest to compare results in a single table or a plot. In the Figure ... we presents point estimates along with 95% confidence intervals. The various estimators show interesting patterns compared to the naive estimate (red dashed line). MI estimators demonstrate notably different behaviours: while PMM produces the highest point estimate with the widest confidence interval, NN yields the lowest estimate, close to the naive value. The other estimators - MI (GLM), IPW (both MLE and GEE), and DR (with and without bias minimization) – cluster together with similar point estimates and confidence interval widths, suggesting some consensus in their bias correction. These methods all indicate a population parameter higher than the naive estimate, but their relative consistency, except for the extreme estimates from MI (PMM) and MI (NN), provides some confidence in their bias correction capabilities.

```
R> dr_summary <- rbind(cbind(ipw_est1$output, ipw_est1$confidence_interval),
+                      cbind(ipw_est2$output, ipw_est2$confidence_interval),
+                      cbind(mi_est1$output, mi_est1$confidence_interval),
+                      cbind(mi_est2$output, mi_est2$confidence_interval),
+                      cbind(mi_est3$output, mi_est3$confidence_interval),
+                      cbind(dr_est1$output, dr_est1$confidence_interval),
+                      cbind(dr_est2$output, dr_est2$confidence_interval))
R> rownames(dr_summary) <- NULL
R> dr_summary$est <- c("IPW (MLE)", "IPW (GEE)", "MI (GLM)", "MI (NN)",
+                      "MI (PMM)", "DR", "DR (BM)")
R> ggplot(data = dr_summary,
+        aes(y = est, x = mean, xmin = lower_bound, xmax = upper_bound)) +
+   geom_point() +
+   geom_vline(xintercept = mean(admin$single_shift),
+              linetype = "dotted", color = "red") +
+   geom_errorbar() +
+   labs(x = "Point estimator and confidence interval", y = "Estimators")
```

4.4. Advanced usage

Bootstrap Approach for Variance Estimation

In the package we allow user to estimate variance of the mean using analytical (default) or bootstrap approach. In case of analytical variance estimators we use the estimators proposed

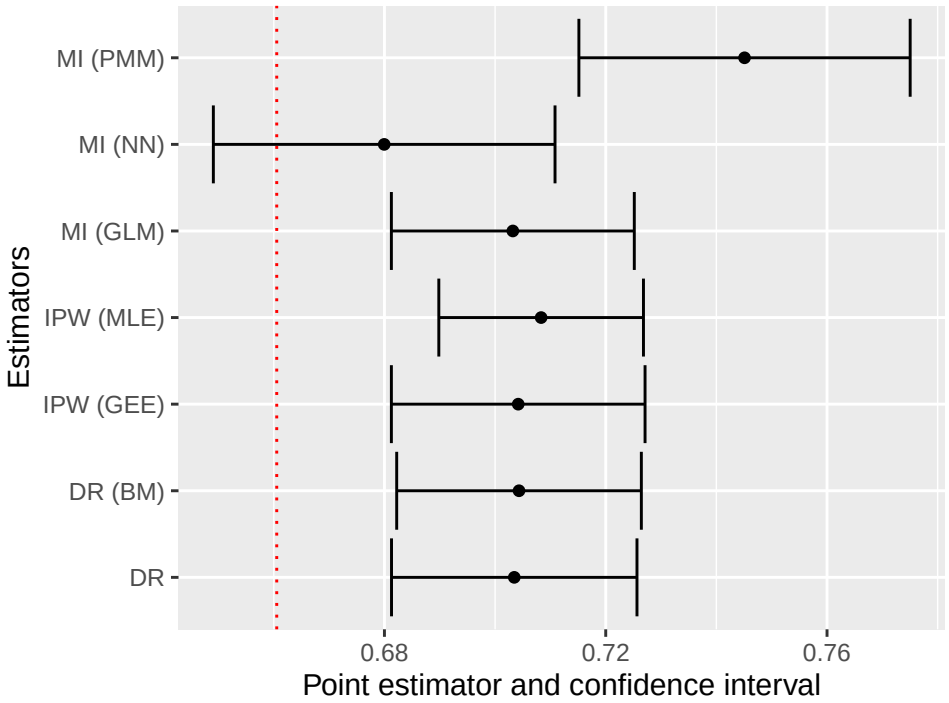

Figure 1: Comparison of estimates of the share of job vacancies offered on a single-shift

in the papers described in the Section 2. Users may disable standard error calculation using `nonprob(se=FALSE)`. The bootstrap approach implemented in the package refers to two samples:

- non-probability – we currently support only simple random sampling with replacement,
- probability – we support all the approaches implemented in the `as.svrepdesign` and we refere the reader to the help file of this function. Currently we do not support the

The bootstrap approach is done in the following way: 1) we independently draw the same number of B bootstrap samples from non-probability and probability survey; 2) we estimate population mean based on selected method (e.g. the DR approach); and 3) calculate bootstrap standard error using the following formulae

$$aaa \tag{11}$$

To specify the bootstrap approach one should use `controlInf()` function with `var_method = "bootstrap"`. Controlling the bootstrap method for probability sample is done by `rep_type` argument which passes the method to the `as.svrepdesign` function. The number of iterations is set in the `num_boot` argument (default 100). If the samples are large or the estimation method is complicated (e.g. involves variable selection) one can set `verbose=TRUE` to track the progress. By default results of bootstrap are stored in the `boot_sample` element of the resulting list (to disable this `keep_boot` should be set to `FALSE`). The following code provides an example of using the IPW approach with the bootstrap approach specified by the argument `control_inference` of the `nonprob` function.

```
R> ipw_est1_boot <- nonprob(
+   selection = ~ region + private + nace + size,
+   target = ~ single_shift,
+   svydesign = jvs_svy,
+   data = admin,
+   method_selection = "logit",
+   control_inference = controlInf(var_method = "bootstrap", num_boot = 50),
+   verbose = F
+ )
```

Next, we compare the estimated standard error with the analytical one below.

```
R> rbind(ipw_est1$output,
+       ipw_est1_boot$output)

                  mean          SE
single_shift  0.7083228 0.009436907
single_shift1 0.7083228 0.010365107
```

To assess the samples one can access the `boot_sample` element of the output list of the `nonprob` function. Note that this is returned as `matrix` because we allow multiple `target` variables.

```
R> head(ipw_est1_boot$boot_sample, n=3)

     single_shift
[1,]    0.7191282
[2,]    0.7233464
[3,]    0.7087409
```

Variable Selection Algorithms

In this section we briefly present how to use variable selection algorithms. In order to specify that a variable selection algorithm should be used one should specify the `control_inference = controlInf(vars_selection = TRUE)` argument. Then, the user should either leave the default or specify the parameters for the outcome via the `controlOut` function or selection outcome (`controlSel`). Both function have the same parameters:

- `penalty` – The penanlization function used during variables selection (possible values: `c("SCAD", "lasso", "MCP")`)
- `nlambda` – The number of λ values. Default is 50.
- `lambda_min` – The smallest value for λ, as a fraction of `lambda.max`. Default is .001.
- `lambda` – A user specified vector of lambdas (only for the `controlSel` function).
- `nfolds` – The number of folds for cross validation. Default is 10.

- `a_SCAD, a_MCP` – The tuning parameter of the SCAD and MCP penalty for selection model. Default is 3.7 and 3 respectively.

For the MI approach we leverage the **ncvreg** package (Breheny and Huang 2011) as it is solely package that uses the SCAD method in R. While for the IPW and DR approaches we developed our own codes in C++ via the **Rcpp** package. In the code below we apply variable selection for the MI GLM estimator using only 5 folds, 25 possible values of λ parameters and apply the LASSO penalty.

```
R> mi_est1_sel <- nonprob(
+   outcome = single_shift ~ region + private + nace + size,
+   svydesign = jvs_svy,
+   data = admin,
+   method_outcome = "glm",
+   family_outcome = "binomial" ,
+   control_outcome = controlOut(nfolds = 5, nlambda = 25, penalty = "lasso"),
+   control_inference = controlInf(vars_selection = TRUE),
+   verbose = TRUE
+ )

Starting CV fold #1
Starting CV fold #2
Starting CV fold #3
Starting CV fold #4
Starting CV fold #5
```

In this case study the MI GLM estimator with variable selection yields almost the same results as the approach without it. Point estimates and standard errors differ at the fourth and third digit respectively.

```
R> rbind("MI without var sel"=mi_est1$output,
+       "MI with var sel"=mi_est1_sel$output)

                        mean         SE
MI without var sel 0.7032081 0.01120231
MI with var sel    0.7034021 0.01119051
```

5. Classes and S3methods

6. Summary and future work

The **nonprobsvy** package provides a comprehensive R toolkit for addressing inference challenges with non-probability samples by integrating them with probability samples or known population totals/means. As non-probability data sources like administrative data, voluntary

online panels, and social media data become increasingly available, statisticians need robust methods to produce reliable population estimates. The package implements *state-of-the-art* approaches including mass imputation, inverse probability weighting (IPW), and doubly robust (DR) methods, each designed to correct selection bias by leveraging auxiliary data. By providing a unified framework and integration with the **survey** package, **nonprobsvy** makes complex statistical methods for non-probability samples more accessible, enabling researchers to produce robust estimates even when working with non-representative data.

There are several avenues for future development of the **nonprobsvy** package. A key priority is implementing model-based calibration and additional methods for estimating propensity scores and weights. The package currently assumes no overlap between probability and non-probability samples, so accounting for potential overlap (e.g., in big data sources and registers) is another important extension. Additional planned developments include handling non-ignorable sample selection through sample selection models, maintaining consistency with calibration weights, and supporting multiple non-probability samples for data integration from various sources.

Further methodological extensions under consideration include empirical likelihood approaches for doubly/multiply robust estimation, integration of machine learning methods like debiased/double machine learning from causal inference, handling measurement error in big data variables, and expanding the bootstrap approach beyond simple random sampling with replacement. The package will also be extended to work with the `svyrep.design` class from the **survey** package and the **svrep** package. These developments will enhance **nonprobsvy**'s capabilities for handling complex survey data structures and modern estimation challenges.

7. Acknowledgements

The authors' work has been financed by the National Science Centre in Poland, OPUS 20, grant no. 2020/39/B/HS4/00941.

Łukasz Chrostowski is the main developer and maintainer of the package. He was also responsible for the first draft of the paper as it is based on his Master's thesis (available at https://github.com/ncn-foreigners/graduation-theses). Piotr Chlebicki contributed to the package and implemented PMM estimators. Maciej Beręsewicz was responsible for the initial idea and the design of the package, testing, reviewing and small contributions code and prepared the final manuscript.

We would like to thank ...

A. List of symbols

Symbol	Description
U	Target population of size N
N	Population size
$\boldsymbol{x}_i$	Vector of auxiliary variables for unit i
y_i	Study variable for unit i
S_A	Non-probability sample
S_B	Probability sample
n_A	Size of non-probability sample
n_B	Size of probability sample
π_i	Inclusion probability for unit i in probability sample
d_i	Design weight $(1/\pi_i)$ for unit i
R_i	Indicator of inclusion into non-probability sample
μ_y	Population mean of target variable Y
π_i^A	Propensity score for unit i in non-probability sample
$m(\boldsymbol{x}_i, \boldsymbol{\beta})$	Regression model for outcome variable
$\hat{\mu}_{IPW}$	Inverse probability weighting estimator
$\hat{\mu}_{MI}$	Mass imputation estimator
$\hat{\mu}_{DR}$	Doubly robust estimator
$\boldsymbol{\theta}$	Parameter vector for selection model
$\boldsymbol{\beta}$	Parameter vector for outcome model
$\hat{y}_i$	Imputed value for unit i
$\hat{N}^A$	Estimated size based on non-probability sample
$\hat{N}^B$	Estimated size based on probability sample

Table 5: List of symbols and their descriptions

B. Algorithms

Algorithm 1: Mass imputation based on a generalized linear model

1: Estimate the regression model $\mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}] = m(\boldsymbol{x}, \boldsymbol{\beta})$ basing on units from S_A sample.
2: For each $i \in S_B$, calculate the imputed value as

$$\hat{y}_i = m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right).$$

Algorithm 2: Mass imputation using the k-nearest-neighbour algorithm

1: If $k = 1$, then for each $i \in S_B$ match $\hat{\nu}(i)$ such that $\hat{\nu}(i) = \underset{j \in S_A}{\arg\min}\, d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$.
2: If $k > 1$, then

$$\hat{\nu}(i, z) = \underset{j \in S_A \setminus \bigcup_{t=1}^{z-1} \{\hat{\nu}(i,t)\}}{\arg\min} d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$$

i.e. $\hat{\nu}(i, z)$ is z-th nearest neighbour from the sample.;
3: For each $i \in S_B$, calculate the imputed value as

$$\hat{y}_i = \frac{1}{k} \sum_{t=1}^{k} y_{\hat{\nu}(i,t)}.$$

Algorithm 3: $\hat{y} - \hat{y}$ Imputation:

1: Estimate regression model $\mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}] = m(\boldsymbol{x}, \boldsymbol{\beta})$.;

2: Impute

$$\hat{y}_i = m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right), \hat{y}_j = m\left(\boldsymbol{x}_j, \hat{\boldsymbol{\beta}}\right)$$

for $i \in S_B, j \in S_A$ and assign each $i \in S_B$ to $\hat{\nu}(i)$, where

$$\hat{\nu}(i) = \underset{j \in S_A}{\arg\min} \|\hat{y}_i - \hat{y}_j\|$$

or

$$\hat{\nu}(i) = \underset{j \in S_A}{\arg\min}\, d\left(\hat{y}_i, \hat{y}_j\right)$$

if d is not induced by the norm.;

3: If $k > 1$, then:

$$\hat{\nu}(i, z) = \underset{j \in S_A \setminus \bigcup_{t=1}^{z-1} \{\hat{\nu}(i,t)\}}{\arg\min} d\left(\hat{y}_i, \hat{y}_j\right)$$

e.g., $\hat{\nu}(i, z)$ is z-th nearest neighbor from a sample.;

4: For $i \in S_B$, calculate imputation value as

$$\hat{y}_i = \frac{1}{k} \sum_{t=1}^{k} y_{\hat{\nu}(i,t)}.$$

Algorithm 4: $\hat{y} - y$ Imputation:

1: Estimate regression $\mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}] = m(\boldsymbol{x}, \boldsymbol{\beta})$.;

2: Impute $\hat{y}_i = m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right)$ for $i \in S_B$ and assign each $i \in S_B$ do $\hat{\nu}(i)$, where $\hat{\nu}(i) = \underset{j \in S_A}{\arg\min} \|\hat{y}_i - y_j\|$ or $\hat{\nu}(i) = \underset{j \in S_A}{\arg\min}\, d\left(\hat{y}_i, y_j\right)$ if d not induced by the norm.;

3: If $k > 1$, then:

$$\hat{\nu}(i, z) = \underset{j \in S_A \setminus \bigcup_{t=1}^{z-1} \{\hat{\nu}(i,t)\}}{\arg\min} d\left(\hat{y}_i, y_j\right).$$

4: For each $i \in S_B$ calculate imputation value as

$$\hat{y}_i = \frac{1}{k} \sum_{t=1}^{k} y_{\hat{\nu}(i,t)}.$$

C. Codes for specific methods

In this section we provide list of methods along with the appropriate set up of the `nonprob` function. We divide this section into two groups: 1) unit-level data from the reference probability sample is available; 2) only a vector of population totals or means (with population size) is available.

C.1. Unit-level probability sample is available

1. Mass imputation based on regression imputation

```
R> nonprob(
+   outcome = y ~ x1 + x2,
+   data = nonprob,
+   svydesign = prob,
+   method_outcome = "glm",
+   family_outcome = "gaussian"
+ )
```

2. Mass imputation based on nearest neighbour imputation

```
R> nonprob(
+   outcome = y ~ x1 + x2,
+   data = nonprob,
+   svydesign = prob,
+   method_outcome = "nn",
+   family_outcome = "gaussian",
+   control_outcome = controlOutcome(k = 2)
+ )
```

3. Mass imputation based on predictive mean matching

```
R> nonprob(
+   outcome = y ~ x1 + x2,
+   data = nonprob,
+   svydesign = prob,
+   method_outcome = "pmm",
+   family_outcome = "gaussian"
+ )
```

4. Mass imputation based on regression imputation with variable selection (LASSO)

```
R> nonprob(
+   outcome = y ~ x1 + x2,
+   data = nonprob,
+   svydesign = prob,
```

```
+   method_outcome = "pmm",
+   family_outcome = "gaussian",
+   control_outcome = controlOut(penalty = "lasso"),
+   control_inference = controlInf(vars_selection = TRUE)
+ )
```

5. Inverse probability weighting (MLE)

```
R> nonprob(
+   selection =  ~ x1 + x2,
+   target = ~ y,
+   data = nonprob,
+   svydesign = prob,
+   method_selection = "logit"
+ )
```

6. Inverse probability weighting with calibration constraint (GEE)

```
R> nonprob(
+   selection =  ~ x1 + x2,
+   target = ~ y,
+   data = nonprob,
+   svydesign = prob,
+   method_selection = "logit",
+   control_selection = controlSel(est_method_sel = "gee", h = 1)
+ )
```

7. Inverse probability weighting with calibration constraint (GEE) with variable selection (SCAD)

```
R> nonprob(
+   selection =  ~ x1 + x2,
+   target = ~ y,
+   data = nonprob,
+   svydesign = prob,
+   method_outcome = "pmm",
+   family_outcome = "gaussian",
+   control_inference = controlInf(vars_selection = TRUE)
+ )
```

8. Doubly robust estimator

```
R> nonprob(
+   selection = ~ x1 + x2,
+   outcome = y ~ x1 + x2,
+   data = nonprob,
```

```
+   svydesign = prob,
+   method_outcome = "glm",
+   family_outcome = "gaussian"
+ )
```

9. Doubly robust estimator with variable selection (SCAD) and bias minimization

```
R> nonprob(
+   selection = ~ x1 + x2,
+   outcome = y ~ x1 + x2,
+   data = nonprob,
+   svydesign = prob,
+   method_outcome = "glm",
+   family_outcome = "gaussian",
+   control_inference = controlInf(
+     vars_selection = TRUE,
+     bias_correction = TRUE
+   )
+ )
```

C.2. Only population totals or means are available

Example declarations

1. Mass imputation based on regression imputation

```
R> nonprob(
+   outcome = y ~ x1 + x2,
+   data = nonprob,
+   pop_totals = c(`(Intercept)`= N,
+                  x1 = tau_x1,
+                  x2 = tau_x2),
+   method_outcome = "glm",
+   family_outcome = "gaussian"
+ )
```

2. Inverse probability weighting

```
R> nonprob(
+   selection =  ~ x1 + x2,
+   target = ~ y,
+   data = nonprob,
+   pop_totals = c(`(Intercept)` = N,
+                  x1 = tau_x1,
+                  x2 = tau_x2),
+   method_selection = "logit"
+ )
```

3. Inverse probability weighting with calibration constraint

```
R> nonprob(
+   selection =  ~ x1 + x2,
+   target = ~ y,
+   data = nonprob,
+   pop_totals = c(`(Intercept)`= N,
+                  x1 = tau_x1,
+                  x2 = tau_x2),
+   method_selection = "logit",
+   control_selection = controlSel(est_method_sel = "gee", h = 1)
+ )
```

4. Doubly robust estimator

```
R> nonprob(
+   selection = ~ x1 + x2,
+   outcome = y ~ x1 + x2,
+   pop_totals = c(`(Intercept)` = N,
+                  x1 = tau_x1,
+                  x2 = tau_x2),
+   method_outcome = "glm",
+   family_outcome = "gaussian"
+ )
```

D. Detailed derivations

Table 6: MLE Functions and Gradients for Different Link Functions

Link	MLE Function	Gradient
`logit`	$\sum_{i \in S_A} \boldsymbol{x}_i^\top \boldsymbol{\theta}$ $- \sum_{i \in S_B} d_i^B \log\left[1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)\right]$	$\sum_{i \in S_A} \boldsymbol{x}_i$ $- \sum_{i \in S_B} d_i^B \pi(\boldsymbol{x}_i, \boldsymbol{\theta}) \boldsymbol{x}_i$
`probit`	$\sum_{i \in S_A} \log\left(\frac{\Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})}{1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})}\right)$ $+ \sum_{i \in S_B} d_i^B \log\left[1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})\right]$	$\sum_{i \in S_A} \frac{\phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})}{\Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})[1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})]} \boldsymbol{x}_i$ $- \sum_{i \in S_B} d_i^B \frac{\phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})}{1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})} \boldsymbol{x}_i$
`cloglog`	$\sum_{i \in S_A} \left\{\log\left[1 - \exp\left(-\exp(\boldsymbol{x}_i^\top \boldsymbol{\theta})\right)\right]\right\}$ $+ \exp(\boldsymbol{x}_i^\top \boldsymbol{\theta}) - \sum_{i \in S_B} d_i^B \exp(\boldsymbol{x}_i^\top \boldsymbol{\theta})$	$\sum_{i \in S_A} \frac{\exp(\boldsymbol{x}_i^\top \boldsymbol{\theta}) \boldsymbol{x}_i}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})}$ $- \sum_{i \in S_B} d_i^B \exp(\boldsymbol{x}_i^\top \boldsymbol{\theta}) \boldsymbol{x}_i$

References

Beaumont JF (2020). "Are probability surveys bound to disappear for the production of official statistics." *Survey Methodology*, **46**(1), 1–28.

Beręsewicz M (2017). "A two-step procedure to measure representativeness of internet data sources." *International Statistical Review*, **85**(3), 473–493.

Beręsewicz M, Szymkowiak M (2024). "Inference for non-probability samples using the calibration approach for quantiles." https://arxiv.org/abs/2403.09726. 2403.09726.

Beręsewicz M, Szymkowiak M, Chlebicki P (2025). "Quantile balancing inverse probability weighting for non-probability samples." *Survey Methodology*, **51**, 0–0.

Breheny P, Huang J (2011). "Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection." *Annals of Applied Statistics*, **5**(1), 232–253. doi:10.1214/10-AOAS388. URL https://doi.org/10.1214/10-AOAS388.

Castro Martín L (2024). *INPS: Inference from Non-Probability Samples*. URL https://pypi.org/project/inps/.

Chen S, Yang S, Kim JK (2022). "Nonparametric Mass Imputation for Data Integration." *Journal of Survey Statistics and Methodology*, **10**(1), 1–24. ISSN 2325-0984, 2325-0992. doi:10.1093/jssam/smaa036. URL https://academic.oup.com/jssam/article/10/1/1/5983829.

Chen Y, Li P, Wu C (2020). "Doubly robust inference with nonprobability survey samples." *Journal of the American Statistical Association*, **115**(532), 2011–2021.

Chlebicki P, Chrostowski Ł, Beręsewicz M (2024). "Data integration of non-probability and probability samples with predictive mean matching."

Citro CF (2014). "From multiple modes for surveys to multiple data sources for estimates." *Survey Methodology*, **40**(2), 137–162.

Cobo B, Ferri-García R, Rueda-Sánchez JL, Rueda MdM (2024). "Software review for inference with non-probability surveys." *The Survey Statistician*, **90**, 40–47.

Daas PJ, Puts MJ, Buelens B, Hurk PAvd (2015). "Big data as a source for official statistics." *Journal of Official Statistics*, **31**(2), 249–262.

Eddelbuettel D, Francois R, Allaire J, Ushey K, Kou Q, Russell N, Ucar I, Bates D, Chambers J (2024). *Rcpp: Seamless R and C++ Integration*. R package version 1.0.13, URL https://CRAN.R-project.org/package=Rcpp.

Eddelbuettel D, Sanderson C (2014). "RcppArmadillo: Accelerating R with high-performance C++ linear algebra." *Computational Statistics and Data Analysis*, **71**, 1054–1063. doi:10.1016/j.csda.2013.02.005.

Elliott MR, Valliant R (2017). "Inference for Nonprobability Samples." *Statistical Science*, **32**(2). ISSN 0883-4237. doi:10.1214/16-STS598. URL https://projecteuclid.org/journals/statistical-science/volume-32/issue-2/Inference-for-Nonprobability-Samples/10.1214/16-STS598.full.

Gelman A (1997). "Poststratification into many categories using hierarchical logistic regression." *Survey Methodology*, **23**, 127.

Goodrich B, Gabry J, Ali I, Brilleman S (2024). *rstanarm: Bayesian applied regression modeling via Stan*. R package version 2.32.1, URL https://mc-stan.org/rstanarm/.

Greifer N (2024). *cobalt: Covariate Balance Tables and Plots*. R package version 4.5.5, URL https://CRAN.R-project.org/package=cobalt.

Jefferis G, Kemp SE, Arya S, Mount D (2024). *RANN: Fast Nearest Neighbour Search (Wraps ANN Library) Using L2 Metric*. R package version 2.6.2, URL https://CRAN.R-project.org/package=RANN.

Kim JK, Park S, Chen Y, Wu C (2021). "Combining Non-Probability and Probability Survey Samples Through Mass Imputation." *Journal of the Royal Statistical Society Series A: Statistics in Society*, **184**(3), 941–963. ISSN 0964-1998, 1467-985X. doi:10.1111/rssa.12696. URL https://academic.oup.com/jrsssa/article/184/3/941/7068406.

Marra G, Rodicw R (2023). *GJRM: Generalized Joint Regression Modelling*.

Rueda MdM, Ferri-García R, Castro L (2020). "The R package NonProbEst for estimation in non-probability surveys." *The R Journal*, **12**, 406–418. ISSN 2073-4859. Https://rjournal.github.io/.

Sant'Anna PHC, Song X, Xu Q (2022). "Covariate Distribution Balance via Propensity Scores." *Journal of Applied Econometrics*, **37**(6), 1093–1120.

Sarig T, Galili T, Eilat R (2023). "balance – a Python package for balancing biased data samples." 2307.06024, URL https://arxiv.org/abs/2307.06024.

Statistics Poland (2021). "The Demand for :abour: Methodological report." *Methodological report*, Statistical Office in Bydgoszcz, Bydgoszcz, Warsaw. URL https://stat.gov.pl/obszary-tematyczne/rynek-pracy/popyt-na-prace/zeszyt-metodologiczny-popyt-na-prace,3,1.html.

Wu C (2022). "Statistical inference with non-probability survey samples." *Survey Methodology*, **48**, 283–311.

Yang S, Kim JK (2020). "Asymptotic theory and inference of predictive mean matching imputation using a superpopulation model framework." *Scandinavian Journal of Statistics*, **47**(3), 839–861. ISSN 0303-6898, 1467-9469. doi:10.1111/sjos.12429. URL https://onlinelibrary.wiley.com/doi/10.1111/sjos.12429.

Yang S, Kim JK, Hwang Y (2021). "Integration of data from probability surveys and big found data for finite population inference using mass imputation." *Survey Methodology*, **47**, 29–58.

Yang S, Kim JK, Song R (2020). "Doubly Robust Inference when Combining Probability and Non-Probability Samples with High Dimensional Data." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **82**(2), 445–465. ISSN 1369-7412, 1467-9868. doi:10.1111/rssb.12354. URL https://academic.oup.com/jrsssb/article/82/2/445/7056072.

Affiliation:

Łukasz Chrostowski
Pearson
First line
Second line
E-mail: lukchr@st.amu.edu.pl
URL: https://posit.co

Piotr Chlebicki
Stockholm University
Matematiska institutionen
Albano hus 1
106 91 Stockholm, Sweden
E-mail: piotr.chlebicki@math.su.se
URL: https://github.com/Kertoo, https://www.su.se/profiles/pich3772

Maciej Beręsewicz
Poznań University of Economics and Business
Statistical Office in Poznań

Poznań University of Economics and Business
Department of Statistics
Institute of Informatics and Quantitative Economics
Al. Niepodległosci 10
61-875 Poznań, Poland

Statistical Office in Poznań
ul. Wojska Polskiego 27/29
60-624 Poznań, Poland
E-mail: maciej.beresewicz@ue.poznan.pl
URL: https://github.com/BERENZ, https://ue.poznan.pl/en/people/dr-maciej-beresewicz/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd