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 **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. 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 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 In-

stitutes (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.

To make it clear what we are dealing with we prepared 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) or kernel density estimation (KDE) described by Chen, Yang and Kim (2022) 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, popular, methods: **rstanarm** (Bayesian approach; 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).

In should be noted that only **nonprobsvy** leveraged the use of the **survey** package to estimate analytic as well as bootstrap variance of the non-probability estimators. Furthermore, none of the packages focusing on frequentist approach to statistical inference implements correct

This paper describes the nonprobsvy package for inference with non–probability samples, available from the Comprehensive R Archive Network (CRAN) at CRAN.R-project. Development version of the package can be also found at github.

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 of size n_A and let $\{(\boldsymbol{x}_i, \pi_i), i \in S_B\}$ be a dataset of a probability sample 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. Each unit in the sample S_B has been assigned a design-based weight given by $d_i = 1/\pi_i$.

The goal is to estimate a finite population mean $\mu_y = \dfrac{1}{N} \sum_{i=1}^{N} y_i$ 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. In this paper we do not consider modifications for the possibly occurring overlap. 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.

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 neigbour imputation (NN),

- Predictive mean mathing (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

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

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

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

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 $h\left(\boldsymbol{x}_i; \boldsymbol{\beta}\right)$, where $h\left(\boldsymbol{x}_i; \boldsymbol{\beta}\right) = \boldsymbol{x}_i$ 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 Bęręsewicz 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. $\pi_i^A > 0$, where $\pi_i^A = P_q\left(R_i^A = 1 \mid \boldsymbol{x}_i, y_i\right)$, 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

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

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

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

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

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

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

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

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

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

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. Package contents and implementation

All of the methods described in this paper have been implemented in the R package **nonprobsvy** (Chrostowski, Berȩsewicz, and Chlebicki 2024). In this chapter, we will show you how to use the main `nonprob` function of the package and what its main features are. The package has been written to be as compatible as possible with the survey package for.inference for probability samples. Namely, the first step to use the nonprobsvy package is to define an object using the `svydesign` function that stores the probability sample `data.frame` and

other objects, such as design weights. This is a negligible step if, instead of the probability sample, we have access to the values of the vector of sums of variables in the population. It is also worth mentioning that in order to speed up the calculations in the case of variable selection, part of the package, or more precisely the whole variable selection algorithm, was written in C++ using the **Rcpp** (Eddelbuettel, Francois, Allaire, Ushey, Kou, Russell, Ucar, Bates, and Chambers 2024) package, which allows the C++ code to be called in the R environment. Moreover, the package is supported by other R packages such as **foreach** (Folashade Daniel Hong Ooi and Weston 2023) (looping construct), **maxLik** (Henningsen and Toomet 2023) (maximum likelihood estimation), **Matrix** (Bates, Maechler *et al.* 2023) (matrix operations), **MASS** Ripley, Venables *et al.* (2023) (statistical functions and datasets), **ncvreg** (Breheny and Huang 2023) (regularization methods), **mathjaxr** (Epskamp 2023) (rendering equations in documentation), **nleqslv** \cite[{nleqslv} (solving non-linear equations), and **doParallel** (Steve Weston and Tenenbaum 2022) (parallel computing).

3.1. Arguments

Below is the definition of most of the arguments we can pass to the function. These are described in more detail in the documentation on the CRAN platform.

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.

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.

4. A case study

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

```
Warning: package 'survival' was built under R version 4.3.3
```

We apply our methods 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 ghe survey can be found in ... The dataset we have included in the package information about NACE (The Nomenclature of Economic Activities) (14 levels), region (16 levels), sector (2 levels), size (3 levels) 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    O     14      1
2 j_2       0    L    O     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
```

For the purpose of the study we define the the sample in a simplified way as provided below using `svydesign` function.

```
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 and can be accessed via the CBOP API. We treat this as the *non-probability sample* as this admin data is voluntary 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. This dataset contains the same variables as JVS with one additional `single_shift`. This 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
```

As the goal of this paper is not to provide a tutorial on using non-probability samples for statistical inference 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%.

4.1. Standard Estimation

We will begin by showing how to use the package for estimation with the standard methods described in Chapter 2. We will start by calling the nonprob function, in the case of analysis with the IPW method.

```
R> est1_logit <- nonprob(
+   selection = ~ region + private + nace + size,
+   target = ~ single_shift,
+   svydesign = jvs_svy,
+   data = admin,
+   method_selection = "logit"
+ )
```

To do this, we define the formulation of the independent variables included in the estimation of the probability of inclusion in the selection argument, as well as the name of the variable for which we are estimating the population mean (`target` argument). In addition, we apply the svydesign object containing the probability sample and non-probability data (data). We estimate the probability using a logit linking function.

```
R> summary(est1_logit)


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: 0.01063
And std.err for nonprobability and probability samples being respectively:
0.0063 and 0.008567

95% Confidence inverval for popualtion mean:
            lower_bound upper_bound
single_shift    0.6874801    0.7291656


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
------------------------

Covariate balance:
(Intercept)     region04     region06     region08     region10     region12
  1028.1277  -2134.5371    674.2230    -567.9955   2183.9110   2625.0580
    region14     region16     region18     region20     region22     region24
  5591.7654 -10057.9909    485.5049    -575.8170   3046.7586   3556.7265
    region26     region28     region30     region32      private      naceD.E
 -6137.0800    579.8114   4063.8814   -3833.6662  46302.3921 -46796.2897
       naceF        naceG        naceH        naceI        naceJ      naceK.L
```

```
   200.0687  13948.8179  -5572.8465 -11450.3311  -2455.3952  -1200.0693
       naceM        naceN       naceO       naceP       naceQ     naceR.S
   1727.5732     29.8287  -1121.2824   -131.6262    580.7330    575.0292
       sizeM       sizeS
 11575.6010  29388.1418
------------------------


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
```

By calling the summary method on an object of class nonprobsvy, we can approach, among other things the value of the estimated mean, standard deviation or its 95% confidence interval. In addition, we display information on the sets 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 estimation. In addition, for diagnostic purposes, we have access to the distribution of the weights calculated from the inclusion probabilities, an assessment of the level of calibration of the variables from the two samples (covariate balance), i.e. the difference in their totals after adjusting for the weights from the non-probability sample, the distribution of the residuals from the model, as well as the values of the AIC, BIC statistics in the case of models based on MLE.

To call the calibration method, we define a `controlSel` function in the `control_selection` argument with the `est_method_sel` argument equal to 'gee' (default `mle`) and set the value of `h` (corresponding to the h-function with **??**).

```
R> est2_logit <- 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")
+ )
```

Another family of estimators (Mass Imputation) requires a slightly different function call. As you might have noticed, the definition of which estimation method we want to use is not defined in the argument of the `nonprob` function, but in the result of the combination of arguments we have defined (specifically formulas). Now the argument in which we define the formula is outcome, which mirrors the definition of formulas for models in functions from the `glm` family (for example).

```
R> est5_glm <- nonprob(
+    outcome = single_shift ~ region + private + nace + size,
```

```
+   svydesign = jvs_svy,
+   data = admin,
+   method_outcome = "glm",
+   family_outcome = "gaussian"
+ )
```

In order to induce doubly robust estimation by the joint randomisation approach, we need to combine, so to speak, the way the formulas and arguments of the two previous approaches are defined, as follows.

```
R> est8_dr1 <- 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"
+ )
```

4.2. Predictive Mean Matching and Nearest Neighbours for Mass Imputation problem

```
R> est6_glm_nn <- nonprob(
+   outcome = single_shift ~ region + private + nace + size,
+   svydesign = jvs_svy,
+   data = admin,
+   method_outcome = "nn",
+   control_outcome = controlOut(k=5)
+ )
```

```
R> est6_glm_pmm1 <- nonprob(
+   outcome = single_shift ~ region + private + nace + size,
+   svydesign = jvs_svy,
+   data = admin,
+   method_outcome = "pmm",
+   control_outcome = controlOut(k=5, predictive_match=1),
+   verbose = TRUE
+ )
```

4.3. Doubly Robust with bias minimization application

```
R> est9_dr2 <- 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)
+ )
```

Previous versions assumed an analytical way of calculating the variance, based on the Taylor method of approximating functions. More details can be found in papers such as Chen *et al.* (2020), Yang and Kim (2020) or Yang *et al.* (2021). In addition, the package allows estimation by the popular bootstrap methods, which are based on the creation of `l` artificial samples drawn from the original samples using a method with return with a specified probability. In our setup, these will be the inverse of the design weights in the case of a probability sample, and the simple return in the case of a non-probability sample, unless the weights for that sample have also been entered in the `weights` argument of `nonprob` function.

4.4. Bootstrap Approach for Variance Estimation

In each variant, we first draw bootstrap samples based on probability and non-probability data, and then we perform the appropriate calculations related to the given estimator to obtain the mean of the population of the `l`-th iteration. Finally, we use the usual formula for the bootstrap variance, and in this way we obtain the estimate of interest.

The boostrap method is called if its type is defined in the control argument `var_method`, which defaults to `analytical`. The number of iterations is set in the `num_boot` argument (default 100).

```
R> est3_logit <- 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,
+ )
```

4.5. Variable Selection Algorithms

```
R> est7_glm_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 = 10),
+    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
```

5. Classes and S3methods

6. Summary and future work

The **nonprobsvy** package is an R tool developed to address challenges associated with making inferences from non-probability samples. As non-probability data sources such as administrative data, voluntary online panels, and social media data become increasingly available, statisticians are faced with the problem of how to integrate these sources with traditional probability samples to produce reliable population estimates. This package provides a suite of methods specifically designed to handle the unique characteristics of non-probability samples, where selection mechanisms are often unknown or biased.

nonprobsvy supports a range of estimation techniques including mass imputation, inverse probability weighting (IPW), and doubly robust (DR) methods. Each of these methods allows researchers to correct for selection bias in non-probability samples by leveraging auxiliary data from probability samples or known population totals. The mass imputation approach imputes values for the outcome variable in the probability sample using models trained on the non-probability sample. The IPW estimator uses estimated propensity scores to weight non-probability samples in a way that aligns them more closely with the population. The DR estimator combines both approaches, resulting in an estimator that remains consistent if either the outcome model or the propensity score model is correctly specified, thus providing a level of robustness against model misspecification.

Designed to integrate with the widely used **survey** package in R, **nonprobsvy** enables researchers to define survey designs, specify selection and outcome models, and estimate population parameters using both probability and non-probability samples. The package supports advanced modeling options, including generalized linear models, nearest neighbor algorithms, and predictive mean matching, and allows for customization of tuning parameters, variable selection, and variance estimation methods (e.g., bootstrap).

Overall, **nonprobsvy** is a comprehensive toolkit for researchers and practitioners working with mixed data sources. It simplifies the implementation of complex statistical methods for non-probability samples, making it easier to produce robust, reliable estimates even when working with non-representative data. By providing a unified framework for inference with

non-probability samples, **nonprobsvy** contributes to the growing field of data integration and enhances the ability of statisticians to make informed decisions from diverse data sources.

A natural extension of this study is that consistency can be maintained when the design weights d_i are replaced by calibration weights, which is often the case when working with survey sample datasets. In addition, other methods of estimating propensity scores proposed in the literature can be considered, such as estimation using the empirical likelihood approach. It is also worth exploring the situation where the two probability and non-probability samples overlap, i.e. there are units present in both datasets. This type of problem will be the focus of the author's future work on non–probability sampling, in particular its integration with other statistical data sources.

7. Acknowledgements

A. List of symbols

B. Example codes for specific methods

C. 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

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

D. Detailed derivations

References

Bates D, Maechler M, *et al.* (2023). *Matrix: Sparse and Dense Matrix Classes and Methods.* R package version 1.6-1.1, URL https://CRAN.R-project.org/package=Matrix.

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

4: For each $i \in S_B$ calculate imputation value as
$$\hat{y}_i = \frac{1}{k}\sum_{t=1}^{k} y_{\hat{\nu}(i,t)}.$$

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.

Breheny P, Huang J (2023). *ncvreg: Regularization Paths for SCAD, MCP, and Elastic Net.* R package version 3.14-0, URL https://CRAN.R-project.org/package=ncvreg.

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."

Chrostowski Ł, Beręsewicz M, Chlebicki P (2024). *nonprobsvy: Package for Inference Based on Non-Probability Samples.* R package version 0.1.1, https://ncn-foreigners.github.io/nonprobsvy/, URL https://github.com/ncn-foreigners/nonprobsvy.

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.

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.

Epskamp S (2023). *mathjaxr: Using MathJax in Rd Files for Dynamic Rendering of Equations.* R package version 1.6-0, URL https://CRAN.R-project.org/package=mathjaxr.

Folashade Daniel Hong Ooi RC, Weston S (2023). *foreach: Provides Foreach Looping Construct for R.* R package version 1.5.2, URL https://CRAN.R-project.org/package=foreach.

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/.

Henningsen A, Toomet O (2023). *maxLik: Maximum Likelihood Estimation and Related Tools.* R package version 1.8-5, URL https://CRAN.R-project.org/package=maxLik.

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.*

Ripley B, Venables W, *et al.* (2023). *MASS: Support Functions and Datasets for Venables and Ripley's MASS.* R package version 7.3-60, URL https://CRAN.R-project.org/package=MASS.

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.

Steve Weston Folashade Daniel SW, Tenenbaum D (2022). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package.* R package version 1.0.17, URL `https://CRAN.R-project.org/package=doParallel`.

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/

Argument	Description	Default
`data`	`data.frame` with data from the non-probability sample.	-
`selection`	`formula` for the selection (propensity) equation.	`NULL`
`outcome`	`formula` for the outcome equation.	`NULL`
`target`	`formula` with target variables.	`NULL`
`svydesign`	Optional `svydesign` object containing probability sample and design weights.	`NULL`
`pop_totals`	Optional named `vector` with population totals of the covariates.	`NULL`
`pop_means`	Optional named `vector` with population means of the covariates.	`NULL`
`pop_size`	Optional `double` with population size.	`NULL`
`method_selection`	Character string specifying the method for propensity score estimation (e.g., `"logit"`).	`"logit"`
`method_outcome`	Character string specifying the method for response variable estimation (e.g., `"glm"`).	`"glm"`
`family_outcome`	Character string describing the error distribution and link function to be used in the model (e.g., `"gaussian"`).	`"gaussian"`
`subset`	Optional `vector` specifying a subset of observations to be used in the fitting process.	`NULL`
`strata`	Optional `vector` specifying strata.	`NULL`
`weights`	Optional `vector` of prior weights to be used in the fitting process.	`NULL`
`na_action`	`function` indicating what should happen when the data contain `NA`'s.	`NULL`
`control_selection`	`list` indicating parameters to use in fitting selection model for propensity scores.	`controlSel()`
`control_outcome`	`list` indicating parameters to use in fitting model for outcome variable.	`controlOut()`
`control_inference`	`list` indicating parameters to use in inference based on probability and non-probability samples.	`controlInf()`
`start_selection`	Optional `vector` with starting values for the parameters of the selection equation.	`NULL`
`start_outcome`	Optional `vector` with starting values for the parameters of the outcome equation.	`NULL`
`verbose`	Logical value indicating if verbose output should be printed.	`FALSE`
`x`	Logical value indicating whether to return the model matrix of covariates as part of the output.	`TRUE`
`y`	Logical value indicating whether to return the vector of outcome variable as part of the output.	`TRUE`
`se`	Logical value indicating whether to calculate and return the standard error of the estimated mean.	`FALSE`
`...`	Additional optional arguments.	

Table 4: `nonprob` function arguments description

Table 5: MLE Functions and Gradients for Different Link Functions

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