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, **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, non-probability data sources are becoming more popular (Bęręsewicz 2017; Beaumont 2020; Biffignandi and Bethlehem 2021). 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 (Schonlau and Couper 2017) or market research (cf. Grow, Perrotta, Del Fava, Cimentada, Rampazzo, Gil-Clavel, Zagheni, Flores, Ventura, and Weber 2022). Since the selection mechanism in these sources is unknown, standard design-based inference methods cannot be directly applied and in case of large datasets may lead to *big data paradox* as described by Meng (2018).

Table 1 compares the basic characteristics of probability and non-probability samples. In particular, what are the advantages and disadvantages of each type with respect to the selection mechanism, the population coverage, bias, variance, costs and timeliness. In general, non-probability samples suffers from unknown selection mechanism (i.e. unknown probabilities of inclusion) and under-coverage of certain groups from the population (e.g. older people). As a result, direct estimation based on non-probability samples are characterised with bias and, in most cases, small variance due to the sample size which leads to so called *big data paradox* i.e. the larger the sample the larger the bias. 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	May be incomplete
Estimation bias	Unbiased under design	Potential systematic bias
Variance of estimates	Typically high	Typically low
Cost	High	Low
Timeliness	Long	Rapid

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

To address this problem, several approaches based on the estimation of propensity scores (i.e. inclusion probabilities) used to derive inverse probability weights (IPW; also known as propensity score weighting/adjustment, cf. Lee (2006); Lee and Valliant (2009)), model-based prediction (in particular, mass imputation estimators; MI) and doubly robust (DR) approach involving IPW and MI estimators has been proposed for two main scenarios: 1) only population-level means or totals are available, and 2) unit-level data is available either in the form registers covering the whole population or in the form of probability surveys (cf. Elliott and Valliant 2017). Wu (2022) classified these approaches into three groups that require a joint randomization framework involving *probability sampling design* (denoted as p) and one of the outcome regression model (denoted as ξ) or propensity score model (denoted as q). In this approach the IPW estimators are under the qp framework, the MI estimators are under the ξp framework, DR estimators are 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 (i.e. IPW estimators); by modelling target variable using various techniques (i.e. MI estimators); or combining both approaches (for instance DR estimators, cf. Chen, Li, and Wu (2020); see also Multilevel Regression and Post-stratification, MRP; *Mister-P*, cf. Gelman (1997)). This

topic have became very popular and number of new methods were proposed; for instance non-parametric approaches based on nearest neighbours (Yang, Kim, and Hwang 2021), kernel density estimation (Chen, Yang, and Kim 2022), empirical likelihood (Kim and Morikawa 2023), model-calibration with LASSO (Chen, Valliant, and Elliott 2018) or quantile balanced IPW (Beręsewicz, Szymkowiak, and Chlebicki 2025) to name a few. It should be highlighted that, on contrary to probability samples, there is no single method that can be used for non-probability samples. As shown 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 for selected packages. We focused on packages available through CRAN or PyPI (for non-CRAN or non-PyPI software see Cobo, Ferri-García, Rueda-Sánchez, and Rueda (2024)). In the comparison we have included four packages that particularly focus on non-probability samples in R: **NonProbEst** (Rueda, Ferri-García, and Castro 2020), in Python **balance** (Sarig, Galili, and Eilat 2023), **inps** (Castro Martín 2024) and our **nonprobsvy**. In addition, we have included two R packages that implements specific methods: **rstanarm** (MRP; Goodrich, Gabry, Ali, and Brilleman (2024)) and **GJRM** (generalized sample selection models; Marra and Rodicw (2023)).

Functionalities	NonProbEst	balance	inps	rstanarm	GJRM	nonprobsvy
IPW	✓	✓	✓	–	?	✓
Calibrated IPW	–	–	–	–	–	✓
MI	✓	–	–	–	–	✓
DR	–	–	✓	–	–	✓
MRP	–	–	–	✓	–	–
Sample selection	–	–	–	–	✓	–
Variable selection	✓	✓	✓	✓	✓	✓
Analytical variance	–	–	–	–	–	✓
Bootstrap variance	✓	–	–	–	–	✓
Integration with **survey** or **samplics**	–	–	–	–	–	✓

Table 2: Comparison of packages and implemented methods

The **NonProbEst** is the most comprehensive package in comparison to other discussed in this section. It allows for various techniques such as IPW or prediction approaches (e.g. model-calibrated). It allows for several different setting of the IPW weights, variables selection and the variance is estimated using leave-one-out Jackknife procedure. Unfortunately the package is no longer developed (last update was in 2022) and some of the techniques are outdated or proved in the recent literature to be inappropriate for non-probability samples. The package also contains various functions aimed at specific methods and does not allow to leverage the **survey** package for estimation. The **balance** package focuses solely on the PS approach. It assumes that the the population totals are known and the authors implemented variance estimator of the weighted mean as a measure of uncertainty for the IPW estimator. The weights for the IPW estimator are constructed using the approach proposed by Schonlau and Couper (2017). The **inps** contains supports unit-level data from probability sample or population, implements IPW, MI and DR estimators, allows for variable selection and

is at very early stage of development. It also implements kernel weighting and a simple bootstrap approach via the **scipy.stats** module (Virtanen, Gommers, Oliphant, Haberland, Reddy, Cournapeau, Burovski, Peterson, Weckesser, Bright, van der Walt, Brett, Wilson, Millman, Mayorov, Nelson, Jones, Kern, Larson, Carey, Polat, Feng, Moore, VanderPlas, Laxalde, Perktold, Cimrman, Henriksen, Quintero, Harris, Archibald, Ribeiro, Pedregosa, van Mulbregt, and SciPy 1.0 Contributors 2020). Neither **balance** nor **ips** leverage usage of the **samplics** module (Diallo 2021). The **GJRM** package is the only package that allows to estimate sample selection models used widely for correction of selection bias in observational studies (including not missing at random mechanism). Unfortunately, to best of our knowledge there is no theory on how this approach can be used for estimating population quantities nor to conduct statistical inference. Finally, the MRP approach is implemented solely in the **rstanarm** with variable selection specified by an appropriate prior.

There are several advantages of the **nonprobsvy** package over the discussed ones. First, the package implements state-of-the-art methods recently proposed in the literature along with valid statistical inference procedures. Second, the package implements other approaches such as calibrated IPW (i.e. PS weights match the population or estimated totals), NN and PMM matching, various IPW and DR estimators with possibility of selection of link functions for logistic regression. Thirdly, the **nonprobsvy** leverage functionalities of the **survey** package to account for the design of the probability sample. Finally, we provide a user friendly API that mimics `glm` or other functions known in R with one main function that allows to specify the approach and estimators. To our knowledge the **nonprobsvy** is the solely software (open or close) that allows for such functionalities.

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 specific methods in separate subsections. In the paper we Wu (2022) notation. Section 3 the main function and the package functionalities. 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 `S3Methods` implemented in the package. Paper finishes with summary and plans for the future works. In the Appendix, Table 9 presents list of symbols and Section B contains algorithms for selected MI estimators. In the Replication Materials we include additional codes for specific estimators described in the paper.

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 $\boldsymbol{x}_i$ and the study (target) variable y_i. 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 $\left\{\left(\boldsymbol{x}_i, \pi_i^B\right), i \in S_B\right\}$ be a dataset of a probability sample S_B of size n_B, where only information about variables $\boldsymbol{x}$ and inclusion probabilities π^B are known for all units in the population. Each unit in the sample S_B has been assigned a design-based weight given by $d_i^B = 1/\pi_i^B$.

Let $R_i^A = I(i \in S_A)$ and $R_i^B = I(i \in S_B)$ be indicators of inclusion into non-probability S_A and probability S_B sample respectively and defined for all units in the target population. Let

$\pi_i^A = P(R_i^A = 1 \mid \boldsymbol{x}_i, y_i) = P(R_i^A = 1 \mid \boldsymbol{x}_i)$ be the propensity scores (PS) which characterize the sample S_A inclusion and participation mechanisms. On the contrary to π_i^B, the π_i^A and $d_i^A = 1/\pi_i^A$ are unknown. This description of the data is presented in a more concise form in Table 3.

Sample	ID	Inclusion (R)	Design weight (d)	Covariates ($\boldsymbol{x}$)	Study variable (y)
Non-probability	1	1	?	✓	✓
S_A	⋮	⋮	⋮	⋮	⋮
	n_A	1	?	✓	✓
Probability	1	0	✓	✓	?
S_B	⋮	⋮	⋮	⋮	⋮
	n_B	0	✓	✓	?

Table 3: Two sample setting

The goal is to estimate a finite population mean $\mu_y = N^{-1} \sum_{i=1}^N y_i$ 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. Given the absence of a universally accepted methodology for achieving this objective, the assumptions vary considerably, as outlined by Wu (2022). However, the main assumptions that apply to all presented in this section methods are:

A1 R_i^A and the study variable y_i are independent given the set of covariates $\boldsymbol{x}_i$ (i.e., $\left(R_i^A \perp y_i \right) \mid \boldsymbol{x}_i$; missing at random mechanism).

A2 All the units in the target population have non-zero PS, i.e., $\pi_i^A > 0$, $i = 1, 2, \ldots, N$ (i.e. no coverage error).

A3 The indicator variables $R_1, R_2, \ldots, R_N$ are independent given the set of auxiliary variables $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$ (i.e. no clustering).

In addition, we assume no overlap between S_A and S_B, and no measurement error in y_i and $\boldsymbol{x}_i$ is observed. Setting presented in Table 3 may be also extended to calibrated d_i^B weights (i.e. d_i^B adjusted for under-coverage, non-contact or non-response; cf. Särndal and Lundström (2005)) but this requires additional developments in the theory about the consistency of the MI, IPW and DR estimators. In the next sections we briefly present methods that are implemented in the package.

2.2. Prediction-based approach

Prediction estimators

In the prediction approach the following semiparametric model for the finite population is assumed

$$E_\xi \left(y_i \mid \boldsymbol{x}_i \right) = m \left(\boldsymbol{x}_i, \boldsymbol{\beta} \right), \quad \text{and} \quad V_\xi \left(y_i \mid \boldsymbol{x}_i \right) = v \left(\boldsymbol{x}_i \right) \sigma^2, \quad i = 1, 2, \ldots, N, \tag{1}$$

where the mean function $m(\cdot, \cdot)$ and the variance $v(\cdot)$ have known forms and it assumed that y_i are independent given $\boldsymbol{x}_i$. The model (1) is assumed to hold for all units in the non-probability sample S_A. Parameters of the (1) can be estimated using quasi maximum likelihood estimation method which covers linear and non-linear models such as generalized linear models (GLM). Let $\boldsymbol{\beta}_0$ and σ_0^2 be the true values of the model parameters $\boldsymbol{\beta}$ and σ^2 under assumed model and the $\hat{\boldsymbol{\beta}}$ be the quasi maximum likelihood estimator of $\boldsymbol{\beta}_0$. Let $m_i = m(\boldsymbol{x}_i, \boldsymbol{\beta}_0)$ and $\hat{m}_i = m(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}})$ for all units $i = 1, ..., N$. Under this settings, Wu (2022) notes there are two commonly used prediction estimators

$$\hat{\mu}_{y,PR1} = \frac{1}{N} \sum_{i=1}^{N} \hat{m}_i \quad \text{and} \quad \hat{\mu}_{y,PR2} = \frac{1}{N} \left\{ \sum_{i \in S_A} y_i - \sum_{i \in S_A} \hat{m}_i + \sum_{i=1}^{N} \hat{m}_i \right\}. \tag{2}$$

Under linear models where $m(\boldsymbol{x}_i, \boldsymbol{\beta}) = \boldsymbol{x}_i' \boldsymbol{\beta}$, the two estimators (2) reduce to

$$\hat{\mu}_{y,PR1} = \mu_{\boldsymbol{x}}' \hat{\boldsymbol{\beta}} \quad \text{and} \quad \hat{\mu}_{y,PR2} = \frac{n_A}{N} \left(\bar{y}_A - \overline{\boldsymbol{x}}_A' \hat{\boldsymbol{\beta}} \right) + \mu_{\boldsymbol{x}}' \hat{\boldsymbol{\beta}}, \tag{3}$$

where $\mu_{\boldsymbol{x}} = N^{-1} \sum_{i=1}^{N} \boldsymbol{x}_i$ is the vector of the population means of the $\boldsymbol{x}$ variables and $\overline{\boldsymbol{x}}_A = n_A^{-1} \sum_{i \in S_A} \boldsymbol{x}_i$ is the vector of the simple means of $\boldsymbol{x}$ from the non-probability sample S_A. If the linear model contains an intercept and $\hat{\boldsymbol{\beta}}$ is the ordinary least square estimator then $\hat{\mu}_{y,PR1} = \hat{\mu}_{y,PR2}$.

This form is appealing as the only requirement is the availability of a non-probability sample S_A and reference population means (or totals and population size N). If the population means are unknown they can be replaced by estimates provided by the reference probability sample S_B i.e. $\sum_{i=1}^{N} \hat{m}_i$ is replaced with $\sum_{i \in S_B} d_i^B \hat{m}_i$ for (2) and $\mu_{\boldsymbol{x}}$ is replaced by $\hat{\mu}_{\boldsymbol{x}} = \hat{N}_B^{-1} \sum_{i \in S_B} d_i^B \boldsymbol{x}_i$ for (3) where $\hat{N}_B = \sum_{i \in S_B} d_i^B$.

Mass imputation estimators

Model-based prediction estimators of μ can be treated as *mass imputation estimators* as the information on y_i is missing entirely in the reference probability sample S_B (but $\boldsymbol{x}_i$ is available) and y_i can be imputed based on the non-probability sample as $\{(y_i, \boldsymbol{x}_i), i \in S_A\}$ is observed. The general for of the MI estimator is given by

$$\hat{\mu}_{y,MI} = \frac{1}{\hat{N}_B} \sum_{i \in S_B} d_i^B y_i^*, \tag{4}$$

where y_i^* is the imputed value of y_i and $\hat{N}_B$ is defined as previously. Under deterministic regression imputation the $\hat{\mu}_{y,MI}$ estimator reduce to (2) estimators.

There are several approaches how to impute y_i^* and in the package we have implemented the following MI estimators: semiparametric approach based on the generalized linear models (MI-GLM), nearest neighbour matching (MI-NN) and predictive mean matching (MI-PMM).

The properties of the MI-GLM estimator where y_i^* are $\hat{m}_i$ from the semiparametric model was studied by Kim, Park, Chen, and Wu (2021). In the **nonprobsvy** package we allow for the following GLM families: `gaussian`, `binomial` and `poisson`.

The MI-NN estimator was initially proposed by Rivers (2007) under the name *sample matching* and theoretical properties for the MI-NN estimator for large non-probability samples (big

data, i.e. covering a significant part of the target population) was studied by Yang *et al.* (2021). The basic idea of the NN matching is as follows: 1) for each i unit in the probability sample S_B find a donor j (or donors) in sample S_A based on some distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$; 2) use the matched values y_j from S_A to impute missing y_i values in the probability sample S_B. Imputed values y_i^* depends on the number of selected k neighbours, for $k = 1$ the closest one is selected and for $k > 1$ a simple average over a vector of selected y may be calculated. The detailed description is presented in the Algorithm 1 in the Appendix. The MI-NN estimator suffers from the curse of dimensionality as proved by for i.e. asymptotic bias of the MI estimator increases as the number of covariates $\boldsymbol{x}$ increases with a fixed k (Abadie and Imbens 2006; Yang and Kim 2020). To overcome this issue PMM approach was proposed.

In the PMM approach matching is done using the predicted values of $\hat{m}_i = m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right)$ instead of $\boldsymbol{x}_i$, thus the NN algorithm is modified as follows: 1) fit the $m\left(\boldsymbol{x}_i, \boldsymbol{\beta}\right)$ to non-probability sample S_A, 2) assign predicted values $\hat{m}_i$ to all units in S_A and S_B; 3) match all units from sample S_B to donor units from sample S_A based on $\hat{m}$ values. The MI-PMM estimator is the same as in the NN approach. Chlebicki, Łukasz Chrostowski, and Beręsewicz (2024) studied properties of two variants of the MI-PMM estimator for non-probability samples: matching predicted to predicted ($\hat{m} - \hat{m}$ matching; denoted as MI-PMM A) and predicted to observed ($\hat{m} - y$ matching; denoted as MI-PMM B). Details about the procedure can be found in Algorithm 2 and 3 in the Appendix. Chlebicki *et al.* (2024) also prove consistency of the MI-PMM A estimator under model mis-specification i.e. the assumed model may be different from the true one.

Variance estimators for the prediction approach

Variance for the MI estimators can be either estimated using analytical or bootstrap approach. Analytical estimator of the variance of the MI-GLM estimator proposed by Kim *et al.* (2021, p. 950) which contains two components: $\hat{V}_1$ (based on the information from both samples S_A and S_B) and $\hat{V}_2$ (based solely on the probability sample S_B); for the MI-NN estimator Yang *et al.* (2021) proposed variance estimator for large S_A samples which reduces to the probability sample S_B part (i.e. design-based variance estimator of the mean which can be easily obtained from the **survey** package) and a proposal for smaller samples can be found in Chlebicki *et al.* (2024); and for the MI-PMM estimators Chlebicki *et al.* (2024) contains formulas for variance estimators which are the same as for the MI-PMM estimators.

The second approach is based on the bootstrap where in each bootstrap replication $b = 1, ..., B$ the following steps are conducted.

1. Independently:

   - draw a simple random sampling with replacement from the non-probability sample S_A,

   - draw a sample according to the declared sampling design from the probability sample S_B (e.g. one can use `as.svrepdesign` function from the **survey** package).

2. Estimate $\mu_{y,MI}^b$ based on appropriate approach (e.g. MI-GLM, MI-NN or MI-PMM).

After obtaining B bootstrap replicates estimate variance based on the following equation

$$\hat{V}_{\text{boot}} = \frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\mu}_y^b - \hat{\mu}_y \right)^2, \tag{5}$$

where $\hat{\mu}_y$ is the estimated mean using either MI-GLM, MI-NN or MI-PMM estimator.

The above approaches are applied when unit-level data from the probability sample S_B are available. If this is not the case and only population means (or totals and population size) we can estimate variance of the $\mu_{y,MI-GLM}$ estimator using the first component $\hat{V}_1$ of the Kim *et al.* (2021) variance estimator (replaced by the survey based population quantities if available). For the variance of the MI-NN and MI-PMM estimators we only allow the bootstrap approach with known population means. Note, that current version of the **nonprobsvy** does not support the replicated weights in the probability sample S_B for any of the estimators discussed in this paper.

2.3. Inverse Probability Weighting

Another popular approach is inverse probability weighting (IPW) that involves estimation of PS given by $\pi_i^A = P\left(i \in S_A \right)$. Similarly as for the prediction approach we have two variants of the IPW estimator given by

$$\hat{\mu}_{y,IPW1} = \frac{1}{N} \sum_{i \in S_A} \frac{y_i}{\hat{\pi}_i^A} \quad \text{and} \quad \hat{\mu}_{y,IPW2} = \frac{1}{\hat{N}^A} \sum_{i \in S_A} \frac{y_i}{\hat{\pi}_i^A}, \tag{6}$$

where the $\hat{\mu}_{y,IPW1}$ is a version of the Horvitz-Thompson estimator and the $\hat{\mu}_{y,IPW2}$ is the Hájek estimator with the estimated population size given by $\hat{N}^A = \sum_{i \in S_A} (\pi_i^A)^{-1}$. This estimator in the case of non-probability samples was discussed by Lee (2006) and Biffignandi and Bethlehem (2021, chapter 13) and several approaches on using propensity scores and alternative versions of the weights were discussed (cf. Elliott and Valliant 2017, section 3) Recently, Chen *et al.* (2020) studied properties of the (6) estimators, proved their consistency as well as derived closed form estimators. Wu (2022, section 4.2) argue that $\hat{\mu}_{y,IPW2}$ estimator performs better than $\hat{\mu}_{y,IPW1}$ even if the population size is known.

The construction of the IPW estimator involved two steps: 1) estimation of the PS; and 2) deriving d_i^A, which in our case are equal to $1/\pi_i^A$. To estimate the propensity scores $\pi_i^A = \pi(\boldsymbol{x}_i, \boldsymbol{\gamma})$ one can use the likelihood approach under assumption that the information about $\boldsymbol{x}_i$ are available for each unit in the population given by (7).

$$\ell(\boldsymbol{\gamma}) = \log \left\{ \prod_{i=1}^{N} \left(\pi_i^A \right)^{R_i} \left(1 - \pi_i^A \right)^{1-R_i} \right\} = \sum_{i \in S_A} \log \left\{ \frac{\pi\left(\boldsymbol{x}_i, \boldsymbol{\gamma} \right)}{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\gamma} \right)} \right\} + \sum_{i=1}^{N} \log \left\{ 1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\gamma} \right) \right\}. \tag{7}$$

In practice, a function of this form cannot be used because we do not observe all units from the population. More realistic approach involves a reference probability sample S_B and as a result the second component of the (7) is replaced forming pseudo log-likelihood function given by (8)

$$\ell^*(\boldsymbol{\gamma}) = \sum_{i \in S_A} \log \left\{ \frac{\pi\left(\boldsymbol{x}_i, \boldsymbol{\gamma} \right)}{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\gamma} \right)} \right\} + \sum_{i \in S_B} d_i^B \log \left\{ 1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\gamma} \right) \right\}. \tag{8}$$

The maximum pseudo-likelihood estimator $\hat{\boldsymbol{\gamma}}$ can be obtained as the solution to the pseudo score equation which under logit function assumed for π_i^A is given by (9)

$$\boldsymbol{U}(\boldsymbol{\gamma}) = \sum_{i \in S_A} \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \pi(\boldsymbol{x}_i, \boldsymbol{\gamma}) \boldsymbol{x}_i. \tag{9}$$

In general, the pseudo score functions $\boldsymbol{U}(\boldsymbol{\gamma})$ at the true values of the model parameters $\boldsymbol{\gamma}_0$ are unbiased under the joint *qp* randomization in the sense that $E_{qp}\{\boldsymbol{U}(\boldsymbol{\gamma}_0)\} = \boldsymbol{0}$, which implies that the estimator $\hat{\boldsymbol{\alpha}}$ is *qp*-consistent for $\boldsymbol{\gamma}_0$ (Wu 2022).

The system (9) can be replaced by a system of general estimation equations. Let $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\gamma})$ be a user-specified vector of functions with the same dimension of $\boldsymbol{\gamma}$ and $\boldsymbol{G}(\boldsymbol{\gamma})$ is defined as

$$\boldsymbol{G}(\boldsymbol{\gamma}) = \sum_{i \in S_A} \boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{\gamma}) - \sum_{i \in S_B} d_i^B \pi(\boldsymbol{x}_i, \boldsymbol{\gamma}) \boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{\gamma}), \tag{10}$$

then solving $\boldsymbol{G}(\boldsymbol{\gamma}) = \boldsymbol{0}$ with the chosen parametric form of π_i^A and the chosen $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\gamma})$ leads to the consistent estimator of $\hat{\boldsymbol{\gamma}}$. In the literature, the most commonly considered functions are $\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{\gamma}) = \boldsymbol{x}_i$ and $\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{\gamma}) = \boldsymbol{x}_i \pi(\boldsymbol{x}_i, \boldsymbol{\gamma})^{-1}$. Note that if the function $\boldsymbol{h}_i = \boldsymbol{x}_i$, then $\boldsymbol{G}$ is reduced to

$$\boldsymbol{G}(\boldsymbol{\gamma}) = \sum_{i \in S_A} \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \pi(\boldsymbol{x}_i, \boldsymbol{\gamma}) \boldsymbol{x}_i,$$

and for the second variant of the $\boldsymbol{h}$ function we get the following form of the function $\boldsymbol{G}$

$$\boldsymbol{G}(\boldsymbol{\theta}) = \sum_{i \in S_A} \frac{\boldsymbol{x}_i}{\pi(\boldsymbol{x}_i, \boldsymbol{\gamma})} - \sum_{i \in S_B} d_i^B \boldsymbol{x}_i, \tag{11}$$

which can be viewed as *calibrated* IPW and equation (11) requires only the knowledge of population totals for the auxiliary variables $\boldsymbol{x}$. Moreover, usage of the (11) leads to doubly robust estimator under assumption that the outcome model is linear (Kim and Riddles 2012).

Variance estimators for the inverse probability weighting approach

Chen *et al.* (2020, section 3.2) derived asymptotic variance estimators for both IPW estimators presented in (6) and presented the plug-in variance estimator for the $\hat{\mu}_{y,IPW2}$ estimator assuming logistic regression. In the package we have implemented this approach for `logit`, `probit` and `cloglog` link functions. We refer the reader to the Wu (2022, section 6.2) and Chrostowski (2024, chapter 3) for more details on how this estimators are derived based on the general estimating equations approach.

Another approach is to use bootstrap which essentially is the same as the one presented in the (5) where the $\hat{\mu}_y$ is replaced by one of the (6) estimators.

2.4. Doubly Robust approach

The IPW and MI estimators are sensible to mis-specified models for PS and outcome respectively. To improve robustness and efficiency, the DR approach was proposed (cf. Robins, Rotnitzky, and Zhao 1994). It incorporates a prediction model for the response variable y_i

and PS model for participation R_i^A. This approach is doubly robust in that sense that the DR estimator remains consistent even if one of the models is mis-specified. We need to consider a joint randomization approach consisting of a non-probability S_A and probability S_B sample and DR inference is done within the qp or ξp framework with no specification which one is correct. The general formulae for the for the DR estimator is given by

$$\tilde{\mu}_{y,DR} = \frac{1}{N} \sum_{i \in S_A} \frac{y_i - m_i}{\pi_i^A} + \frac{1}{N} \sum_{i=1}^{N} m_i,$$

where π_i^A and m_i are defined as previously. In the next subsections we discuss two approaches to the DR estimation.

Parameters estimated separately

Chen *et al.* (2020) proposed two DR estimators given in (12) and (13) under assumption that population size either known or estimated

$$\hat{\mu}_{y,DR1} = \frac{1}{N} \sum_{i \in S_A} d_i^A \left\{ y_i - m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right) \right\} + \frac{1}{N} \sum_{i \in S_B} d_i^B m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right), \tag{12}$$

and

$$\hat{\mu}_{y,DR2} = \frac{1}{\hat{N}^A} \sum_{i \in S_A} d_i^A \left\{ y_i - m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right) \right\} + \frac{1}{\hat{N}^B} \sum_{i \in S_B} d_i^B m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right), \tag{13}$$

where $d_i^A = \pi\left(\boldsymbol{x}_i, \boldsymbol{\gamma}\right)^{-1}$, $\hat{N}^A = \sum_{i \in S_A} d_i^A$ and $\hat{N}^B = \sum_{i \in S_B} d_i^B$. The estimator (13) using the estimated population size has better performance in terms of bias and mean squared error and should be used in practice. However, the main limitation is estimation of the variance which will be discussed at the end of this section.

Chen *et al.* (2020) suggested to construct the (12) or (13) estimator based on the two models estimated separately which is attractive as it is possible to specify different number of variables for the propensity and outcome model. An alternative approach was proposed by Yang, Kim, and Song (2020), similar to Kim and Haziza (2014) and will be discussed in the next subsection.

Minimization of the bias for doubly robust methods

Yang *et al.* (2020) discussed variable selection for high-dimensional setting and noted that 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 $\boldsymbol{\gamma}$ parameters based on the bias of the population mean estimator. This method allows for the estimation of the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ in a single step, rather than in two separate steps. First, they derived the bias of the $\hat{\mu}_{DR}$, assuming $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\gamma}) = \boldsymbol{x}\pi(\boldsymbol{x}, \boldsymbol{\gamma})^{-1}$ for the IPW estimator, which is given by equation (14)

$$\text{bias}\left(\hat{\mu}_{DR}\right) = \mid \hat{\mu}_{DR} - \mu \mid = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{R_i^A}{\pi\left(\boldsymbol{x}_i,\boldsymbol{\gamma}\right)} - 1\right\}\left\{y_i - m\left(\boldsymbol{x}_i,\boldsymbol{\beta}\right)\right\}$$
$$+ \frac{1}{N}\sum_{i=1}^{N}\left(R_i^B d_i^B - 1\right)m\left(\boldsymbol{x}_i,\boldsymbol{\beta}\right) \tag{14}$$

The goal of this approach is to minimize $\text{bias}\left(\hat{\mu}_{DR}\right)^2$ which leads to solving the following system of equations

$$\left(\begin{array}{c} \sum_{i=1}^{N} R_i^A \left\{\frac{1}{\pi(\boldsymbol{x}_i,\boldsymbol{\gamma})} - 1\right\}\left\{y_i - m\left(\boldsymbol{x}_i,\boldsymbol{\beta}\right)\right\}\boldsymbol{x}_i \\ \sum_{i=1}^{N} \frac{R_i^A}{\pi(\boldsymbol{x}_i,\boldsymbol{\gamma})}\dot{m}\left(\boldsymbol{x}_i,\boldsymbol{\beta}\right) - \sum_{i\in S_B} d_i^B \dot{m}\left(\boldsymbol{x}_i,\boldsymbol{\beta}\right) \end{array}\right) = \boldsymbol{0}, \tag{15}$$

where $\dot{m}\left(\boldsymbol{x}_i,\boldsymbol{\beta}\right) = \frac{\partial m(\boldsymbol{x}_i,\boldsymbol{\beta})}{\partial\boldsymbol{\beta}}$ and system (15) can be solved using Newton–Raphson method. This approach, without variable selection, is equivalent to Kim and Haziza (2014) and was extensively discussed by Chen *et al.* (2020) and Wu (2022) in the context of estimation of parameters and its variance estimator. The main limitation is the existence of the solution to (15) as it may not exist unless the two sets of covariates used, respectively, in the outcome regression model and the PS model have the same dimensions. That is why Yang *et al.* (2020) suggested to use this approach on a union of variables from both models (e.g. after variable selection).

In the **nonprobsvy** package we have implemented these approaches not only for $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{\gamma}) = \boldsymbol{x}_i\pi\left(\boldsymbol{x}_i,\boldsymbol{\gamma}\right)^{-1}$ but also for $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{\gamma}) = \boldsymbol{x}$ and various link functions for propensity score model. We also allow selection of this estimation strategy with or without variable selection methods discussed in Section 2.5. As noted in the beginning the selection between (12) and (13) result in a different approach of estimating variance which will be presented in the next subsection.

Variance estimators for the doubly robust approach

Yang *et al.* (2020) derived a closed form estimator for the (12) but this requires knowledge of the population and it is requires bias correction to obtain valid estimator for $V_{\xi p}\left(\hat{\mu}_{y,DR} - \mu_y\right)$ under the outcome regression model ξ. A doubly robust variance estimator for the $\hat{\mu}_{y,DR2}$ given by (13) is not yet available in the literature. In the package, for the analytical variance estimator of the $\hat{\mu}_{y,DR2}$ we simply replace N with estimated $\hat{N}_A$ and $\hat{N}_B$ and we strongly advice users to be be careful when using this approach.

Alternatively, one can use the bootstrap approach. Chen *et al.* (2020) showed that bootstrap approach presented in the Section 2.2 have good performance in terms of coverage rate when one of the working models is correctly specified. That is why this approach is recommended for all users.

2.5. Variable selection algorithms

Yang and Kim (2020) point that using variable selection techniques during estimation is crucial, especially when dealing with high-dimensional non-probability sample. Variable selection not only improves the model stability and computational feasibility, but also reduces the variance, which can increase when irrelevant auxiliary variables are included.

The most popular approaches in the statistical literature are penalisation 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 (cf. Tibshirani 1996; Breheny and Huang 2011).

The selection procedure works in a similar way for non-probability methods with loss functions modified to take into account external data sources such as sample or population totals or averages. In particular, the technique is divided into two steps: 1) we select the relevant variables using an appropriately constructed loss function (and possibly using the (15) approach to obtain the final estimates of the model parameters); and 2) we construct chosen estimator using variables selected from the first step. For instance, Yang *et al.* (2020) used (16) as a loss function for estimating outcome equation parameters

$$\text{Loss}\left(\lambda_{\boldsymbol{\beta}}\right) = \sum_{i=1}^{N} R_i^A \left[y_i - m\left\{\boldsymbol{x}_i, \boldsymbol{\beta}(\lambda_{\boldsymbol{\beta}})\right\}\right]^2, \tag{16}$$

where $m\left\{\boldsymbol{x}_i, \boldsymbol{\beta}(\lambda_{\boldsymbol{\beta}})\right\}$ is the penalized function for the $\boldsymbol{\beta}$ parameters with a tuning parameter $\lambda_{\boldsymbol{\beta}}$ and loss functions for the PS function presented in the Table 4 where $\lambda_{\boldsymbol{\gamma}}$ is the tuning parameter.

$\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{\gamma})$ function	Loss function $\lambda_{\boldsymbol{\gamma}}$
$\boldsymbol{x}_i$	$\sum_{j=1}^{p}\left(\sum_{i=1}^{N}\left[R_i^A - \frac{R^B \pi\{\boldsymbol{x}_i, \boldsymbol{\gamma}(\lambda_{\boldsymbol{\gamma}})\}}{\pi_i^B}\right]\boldsymbol{x}_{i,j}\right)^2$
$\boldsymbol{x}_i\pi_i(\boldsymbol{x}_i, \boldsymbol{\gamma})^{-1}$	$\sum_{j=1}^{p}\left(\sum_{i=1}^{N}\left[\frac{R_i^A}{\pi\{\boldsymbol{x}_i, \boldsymbol{\gamma}(\lambda_{\boldsymbol{\gamma}})\}} - \frac{R_i^B}{\pi_i^B}\right]\boldsymbol{x}_{i,j}\right)^2$

Table 4: Loss functions for the PS function depending on the $\boldsymbol{h}(\cdot, \cdot)$ function

where R_i^A and R_i^B are indicator functions defining the inclusion into non-probability S_A and probability S_B samples respectively. Yang *et al.* (2020) discussed only the SCAD penalty and the $\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{\gamma}) = \boldsymbol{x}_i$ function for the DR estimator only. In the **nonprobsvy** package we have extended this approach to first variant of the $\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{\gamma})$ as presented in the first row of the Table (4), allowed user to select other link functions for the π_i^A, implemented other penalty functions and extended the variable selection to MI and IPW estimators. In the next section we will discuss how to define the approaches presented in this section.

3. The main function and the package functionalities

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 5 and the obligatory ones is `data`, and one of the `selection`, `outcome`, or `target` arguments must be specified depending on the chosen method. For the `outcome`, or `target` we allow for multiple y variables.

Argument	Description
`data`	a `data.frame` with data from the non-probability sample
`selection`	a `formula` for the selection (PS) equation
`outcome`	a `formula` for the outcome equation (e.g. `y1+y2   x1 + x2`)
`target`	a `formula` with target variables (e.g. `y1+y2+y3`)
`svydesign`	an optional `svydesign2` object
`pop_totals`, `pop_means`, `pop_size`	an optional named `vector` with population totals or means of the covariates and population size
`method_selection`	a link function for the IPW approach (`c("logit", "probit", "cloglog")`)
`method_outcome`	specification of the MI approach (`c("glm", "nn", "pmm")`)
`family_outcome`	an GLM family for the MI approach (`c("gaussian", "binomial", "poisson")`)
`subset`	an optional `vector` specifying a subset of observations to be used in the fitting process
`weights`	an optional `vector` of prior case-weights to be used in the fitting process
`na_action`	a `function` indicating what to do with `NA`'s
`control_selection`, `control_outcome`, `control_inference`	control functions with parameters for PS, outcome model and the variance estimation respectively
`start_selection`, `start_outcome`	an optional `vector` with starting values for the parameters of the PS and the outcome equation
`verbose`	a `logical` value indicating if information should be printed
`se`	a `logical` value indicating whether to calculate and return the standard error of the estimated mean
`...`	Additional optional arguments

Table 5: `nonprob` function arguments description

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). The `nonprob` function is used to specify inference methods through `selection` and `outcome` arguments.

If a `svydesign2` object is provided and the `selection` specified then the IPW estimators are used (by default parameters of the PS model are using (9)), if the `outcome` then the MI approach is used (the default is the MI-GLM with `gaussian` family) and if both are specified the DR approach is applied (parameters (β, γ) are estimated separately and the (13) is used). Selection of the specific 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.

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

- `data` – a `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` – PS weights or `NULL` (for the MI estimators),

- `output` - a `data.frame` containing point and standard error estimates,

- `confidence_interval` - a `data.frame` confidence intervals for the mean,

- `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.

3.2. Controlling type of estimators

The `control_out` function can be used to specify 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 defined in the same way as in the `control_sel` function), and detailed configuration for NN and PMM approaches (using parameters like `predictive_match`, `pmm_weights`, and `pmm_k_choice`). For both approaches we use the **RANN** package (Jefferis, Kemp, Arya, and Mount 2024) with the kd-tree algorithm and Euclidean distance as a default. We currently do not support other distances (e.g. Gower). Table 6 presents example usage of the `control_sel` function for three types of MI estimators.

Estimator	Declaration with the `control_out`
MI-GLM with the LASSO penalty and 5 folds	`nonprob(outcome = y1 ~ x1 + x2, data = df, svydesign=prob, control_outcome = control_out(penalty="lasso", folds = 5))`
MI-NN with the `bd` algorithm	`nonprob(outcome = y1 ~ x1 + x2, data = df, svydesign=prob, control_outcome = control_out(treetype = "bd"))`
MI-PMM A with $k = 3$	`nonprob(outcome = y1 ~ x1 + x2, data = df, svydesign=prob, control_outcome = control_out(k=3,predictive_match=2))`

Table 6: Example declarations of the MI estimators

The `control_sel` function provides essential control parameters for fitting the selection model in the `nonprob` function. It allows users to select between the MLE given by (9) or GEE defined in (11) approach through `est_method_sel` argument and specify the $h(\cdot, \cdot)$ function through the `h` argument, specify the optimizer (`optimizer` argument) and the which variable selection should be applied (using different penalty functions like SCAD, lasso, and MCP by specifying the `penalty` argument) along with parameters (e.g. number of folds via the `nfolds` argument). Estimation of the parameters of the PS for the calibrated IPW is done by using the **nleqslv** package and fitting parameters (arguments starting with the `nleqslv_*`). Table 7 presents example usage of the `control_sel` function for two types of IPW estimators.

Estimator	Declaration with the `control_sel`

Calibrated IPW	`nonprob(selection = ~ x1 + x2, target = ~y1,`
	`data = df, svydesign = prob, control_selection =`
	`control_sel(est_method_sel="gee"))`

IPW with the MCP penalty and 5 folds	`nonprob(selection = ~ x1 + x2, target = ~y1,`
	`data = df, svydesign = prob, control_selection =`
	`control_sel(penalty="MCP", nfolds=5))`

Table 7: Example declarations of the IPW estimators

3.3. Controlling the variance estimation

Finally, the `control_inf` 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). Table 8 presents example usage of the `control_inf` for the IPW and DR estimators.

Estimator	Declaration with the `control_sel`

Calibrated IPW with variable selection, bootstrap and $B = 50$	`nonprob(selection = ~ x1 + x2, target = ~y1,`
	`data = df, svydesign = prob, control_selection`
	`= control_sel(est_method_sel="gee"),`
	`control_inference = control_inf(vars_selection=TRUE,`
	`var_method="bootstrap", rep_type =`
	`"subbootstrap", B=50))`

The DR with the SCAD penalty, 5 folds and bias correction	`nonprob(selection = ~ x1 + x2, outcome =`
	`y1 ~ x1 + x2, data = df, svydesign = prob,`
	`control_selection = control_sel(penalty="SCAD",`
	`nfolds=5), control_inference =`
	`control_inf(vars_selection=TRUE,`
	`bias_correction=TRUE))`

Table 8: Example declarations of the IPW estimators

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 packages.

```
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 a stratified sampling design. The details regarding the survey can be found in Statistics Poland (2021). The dataset contains information about industry code NACE (14 levels, `nace` column), `region` (16 levels), sector (2 levels, `private` column), size of the entity (3 levels: Small to 9, Medium 10-49 and Large 50+) and the final weight (i.e. design weight corrected for non-contact and non-response). We treat it here as the d weight.

```
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
```

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 so we use a simplified version by specifying `~ size + nace + region` and we do not know have information on the non-response and its correction so we simply assume that the `weight` 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 seek 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 i.e. `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 `control_sel` 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`.

```
R> ipw_est2 <- nonprob(
```

```
+    selection = ~ region + private + nace + size,
+    target = ~ single_shift,
+    svydesign = jvs_svy,
+    data = admin,
+    method_selection = "logit",
+    control_selection = control_sel(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 slightly higher.

```
R> ipw_est2


Call:
nonprob(data = admin, selection = ~region + private + nace +
    size, target = ~single_shift, svydesign = jvs_svy, method_selection = "logit",
    control_selection = control_sel(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 significantly improves the balance as can be accessed by the `check_balance` function:

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


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

Notice that, neither in the package nor in 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 MI 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`) and 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. `control_out(k=5)`). For the NN MI estimator there is no need to specify the `family_outcome` argument as no model is estimated underneath.

```
R> mi_est2 <- nonprob(
+   outcome = single_shift ~ region + private + nace + size,
+   svydesign = jvs_svy,
+   data = admin,
+   method_outcome = "nn",
+   control_outcome = control_out(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 = control_out(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 = control_inf(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
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 = control_inf(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 = control_inf(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 1 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
R> df_s <- 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(df_s) <- NULL
R>
R> df_s$est <- c("IPW (MLE)", "IPW (GEE)", "MI (GLM)", "MI (NN)",
+                 "MI (PMM)", "DR", "DR (BM)")
R>
R> ggplot(data = df_s,
+         aes(y = est, x = mean, xmin = lower_bound, xmax = upper_bound)) +
+   geom_point() +
+   geom_vline(xintercept = mean(admin$single_shift),
+              linetype = "dotted", color = "red") +
+   geom_errorbar() +
+   labs(x = "Point estimator and confidence interval", y = "Estimators") +
+   theme_bw()
```
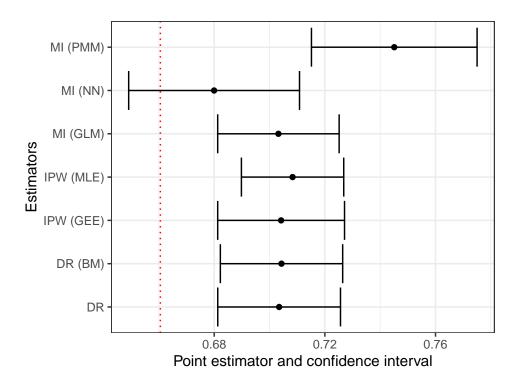

Figure 1: Comparison of estimates of the share of job vacancies offered on a single-shift

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 as described in Section 2.2. In case of analytical variance estimators we use the estimators proposed 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 refer the reader to the help file of this function.

To specify the bootstrap approach one should use `control_inf()` 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 = control_inf(var_method = "bootstrap", num_boot = 50),
+    verbose = F
+ )
```

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

```
R> rbind("IPW analytic variance"=ipw_est1$output,
+        "IPW bootstrap variance"=ipw_est1_boot$output)


                            mean          SE
IPW analytic variance  0.7083228 0.009436907
IPW bootstrap variance 0.7083228 0.010304809
```

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,]     0.7027637
[2,]     0.7180065
[3,]     0.7139336
```

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 = control_inf(vars_selection = TRUE)` argument. Then, the user should either leave the default or specify the parameters for the outcome via the `control_out` function or selection outcome (`control_sel`). Both function have the same parameters:

- `penalty` – The penalization 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 `control_sel` 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. For the IPW and DR approaches we have developed our own codes in C++ via the **Rcpp** and **RcppArmadillo** packages. 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 = control_out(nfolds = 5, nlambda = 25, penalty = "lasso"),
+   control_inference = control_inf(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

A. List of symbols

Symbol	Description
U	Target population of size N
S_A	Non-probability sample
S_B	Probability sample
N	Population size
n_A	Size of non-probability sample
n_B	Size of probability sample
$\hat{N}^A$	Estimated size based on non-probability sample
$\hat{N}^B$	Estimated size based on probability sample
$\boldsymbol{x}_i$	Vector of auxiliary variables for unit i
y_i	Value of the study/target variable for unit i
y_i^*	Imputed value for unit i in S_B
π_i^A	Propensity score for unit i in non-probability sample
π_i^B	Inclusion probability for unit i in probability sample
d_i^A	Inverse probability weight $(1/\pi_i^A)$ for non-probability sample
d_i^B	Design weight $(1/\pi_i^B)$ for probability sample
R_i^A	Indicator of inclusion into non-probability sample
R_i^B	Indicator of inclusion into probability sample
μ	Population mean of target variable y
$\mu_{\boldsymbol{x}}$	Population means of auxiliary variables $\boldsymbol{x}$
$m(\boldsymbol{x}_i, \boldsymbol{\beta})$	Semiparametric model for outcome variable
$\dot{m}(\boldsymbol{x}_i, \boldsymbol{\beta})$	First derivative of the $m(\boldsymbol{x}_i, \boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$
$\pi(\boldsymbol{x}_i, \boldsymbol{\gamma})$	Propensity score model for R_i^A
$\boldsymbol{\beta}$	Parameter vector for outcome model
$\boldsymbol{\gamma}$	Parameter vector for propensity score model
$\lambda_{\boldsymbol{\beta}}, \lambda_{\boldsymbol{\gamma}}$	Tuning parameters for penalisation methods
$\hat{\mu}_{\boldsymbol{x}}$	Estimator for the population means of auxiliary variables $\boldsymbol{x}$
$\bar{\boldsymbol{x}}_A$	A vector of the sample means of the auxiliary variables $\boldsymbol{x}$ from S_A
$\hat{\mu}_{PR}$	Prediction estimators
$\hat{\mu}_{MI}$	Mass imputation estimator
$\hat{\mu}_{IPW}$	Inverse probability weighting estimator
$\hat{\mu}_{DR}$	Doubly robust estimator
$\hat{V}_{boot}$	Variance estimator based on the bootstrap

Table 9: List of symbols and their descriptions

B. Algorithms for the MI-NN and MI-PMM estimators

Algorithm 1: 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 2: $\hat{y} - \hat{y}$ 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 3: $\hat{y} - y$ 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

$$\hat{\nu}(i) = \arg\min_{j \in S_A} \|\hat{y}_i - y_j\| \text{ or } \hat{\nu}(i) = \arg\min_{j \in S_A} d\left(\hat{y}_i, y_j\right) \text{ if } d \text{ not induced by the norm.};$$

3: If $k > 1$, then:

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

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

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

References

Abadie A, Imbens GW (2006). "Large sample properties of matching estimators for average treatment effects." *econometrica*, **74**(1), 235–267.

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

Biffignandi S, Bethlehem J (2021). *Handbook of Web Surveys*. John Wiley & Sons. ISBN 9781119371687. doi:10.1002/9781119371717.

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 J, Valliant R, Elliott M (2018). "Model-assisted calibration of non-probability sample survey data using adaptive LASSO." *Survey Methodology*, **44**(1), 117–144.

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, Łukasz Chrostowski, Beręsewicz M (2024). "Data integration of non-probability and probability samples with predictive mean matching." 2403.13750, URL https://arxiv.org/abs/2403.13750.

Chrostowski L (2024). "Statistical inference with non-probability samples." Master's thesis, Adam Mickiewicz Universith.

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.

Diallo MS (2021). "samplics: a Python Package for selecting, weighting and analyzing data from complex sampling designs." *Journal of Open Source Software*, **6**(68), 3376. doi:10.21105/joss.03376. URL https://doi.org/10.21105/joss.03376.

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.

Grow A, Perrotta D, Del Fava E, Cimentada J, Rampazzo F, Gil-Clavel S, Zagheni E, Flores RD, Ventura I, Weber I (2022). "Is Facebook's Advertising Data Accurate Enough for Use in Social Science Research? Insights from a Cross-National Online Survey." *Journal of the Royal Statistical Society Series A: Statistics in Society*, **185**(2), 343–363. ISSN 0964-1998. doi:10.1111/rssa.12948. URL https://doi.org/10.1111/rssa.12948.

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, Haziza D (2014). "Doubly robust inference with missing data in survey sampling." *Statistica Sinica*, **24**(1), 375–394.

Kim JK, Morikawa K (2023). "An Empirical Likelihood Approach to Reduce Selection Bias in Voluntary Samples." *Calcutta Statistical Association Bulletin*, **75**(1), 8–27. `doi:10.1177/00080683231186488`.

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`.

Kim JK, Riddles MK (2012). "Some theory for propensity-score-adjustment estimators in survey sampling." *Survey Methodology*, **38**(2), 157–165.

Lee S (2006). "Propensity score adjustment as a weighting scheme for volunteer panel web surveys." *Journal of official statistics*, **22**(2), 329.

Lee S, Valliant R (2009). "Estimation for volunteer panel web surveys using propensity score adjustment and calibration adjustment." *Sociological Methods & Research*, **37**(3), 319–343.

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

Meng XL (2018). "Statistical Paradises and Paradoxes in Big Data (I): Law of Large Populations, Big Data Paradox, and the 2016 US Presidential Election." *Annals of Applied Statistics*, **12**, 685–726. `doi:10.1214/18-AOAS1161SF`.

Rivers D (2007). "Sampling for web surveys." In *Proceedings of the Survey Research Methods Section, Joint Statistical Meetings*, pp. 1–26. American Statistical Association, Alexandria, VA.

Robins JM, Rotnitzky A, Zhao LP (1994). "Estimation of regression coefficients when some regressors are not always observed." *Journal of the American statistical Association*, **89**(427), 846–866.

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/.

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`.

Särndal CE, Lundström S (2005). *Estimation in surveys with nonresponse.* John Wiley & Sons.

Schonlau M, Couper MP (2017). "Options for Conducting Web Surveys." *Statistical Science*, **32**(2), 279 – 292. `doi:10.1214/16-STS597`. URL `https://doi.org/10.1214/16-STS597`.

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.

Tibshirani R (1996). "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **58**(1), 267–288.

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, SciPy 10 Contributors (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." *Nature Methods*, **17**, 261–272. doi:10.1038/s41592-019-0686-2.

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/jrssb/article/82/2/445/7056072.

Affiliation:

Łukasz Chrostowski
Adam Mickiewicz University
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

Centre for the Methodology of Population Studies
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/