



## 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 existence of either full population or probability-based population information and leverage 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 statisticians, public opinion or market researchers who would like to use non-probability samples (e.g. big data, opt-in web panels, social media) to accurately estimate population 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, censuses or is obtained from administrative registers and covers all (or nearly all) units of the population. However, owing to increasing non-response rates, particularly unit non-response and non-contact, which result 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 (Beręsewicz 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 underlying these sources is unknown, standard design-based inference methods cannot be directly applied and, in the case of large datasets, can lead to the *big data paradox* described by Meng (2018).

Table 1 compares basic characteristics of probability and non-probability samples. In particular, it shows the advantages and disadvantages of each type with respect to the selection mechanism, the population coverage, bias, variance, costs and timeliness. In general, the quality of non-probability samples suffers from an 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 estimates based on non-probability samples are biased and, in most cases, are characterised by small variance owing to their size, which is known as the *big data paradox*, i.e. the larger the sample, the larger the bias. Certainly, the costs and timeliness of these surveys are significantly smaller than those of 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 delay	Very short delay

Table 1: A comparison of probability and non-probability samples and their characteristics

To address this problem, several approaches have been proposed, which rely on the estimation of propensity scores (i.e. inclusion probabilities) for deriving inverse probability weights (IPW; also known as propensity score weighting/adjustment, cf. Lee (2006); Lee and Valliant (2009)), on model-based prediction (in particular, mass imputation estimators; MI) and on the doubly robust (DR) approach involving IPW and MI estimators. Two main scenarios are usually considered: 1) only population-level means or totals are available, and 2) unit-level data are available either in the form of 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 a *probability sampling design* (denoted as  $p$ ) and an outcome regression model (denoted as  $\xi$ ) or a propensity score model (denoted as  $q$ ). According to this classification, IPW estimators represent the  $qp$  framework, MI estimators represent the  $\xi p$  framework, and DR estimators can represent either the  $qp$  or the  $\xi p$  framework.

Most approaches assume that population data are used to reduce the bias of non-probability sampling by the right reweighting to reproduce known population totals/means (i.e. IPW estimators); by modelling the target variable using various techniques (i.e. MI estimators); or by combining both approaches (e.g. DR estimators, cf. Chen, Li, and Wu (2020); see also

Multilevel Regression and Post-stratification, MRP; *Mister-P*, cf. [Gelman \(1997\)](#)). This topic has become very popular and a number of new methods have been 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, in contrast to probability samples, there is no single method that can be used for non-probability samples. Based on the methods available in the literature several statistical software solutions have been developed, which are presented in the next section.

### 1.1. Software for non-probability samples

Table 2 presents a comparison of selected packages in terms of the availability of various inference methods. We focus on packages available through CRAN or PyPI (for non-CRAN or non-PyPI software see [Cobo, Ferri-García, Rueda-Sánchez, and Rueda \(2024\)](#)). The comparison includes four packages that particularly focus on non-probability samples: in R – **NonProbEst** ([Rueda, Ferri-García, and Castro 2020](#)) and our **nonprobsvy**, and in Python – **balance** ([Sarig, Galili, and Eilat 2023](#)), **inps** ([Castro Martín 2024](#)). 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 <b>survey</b> or <b>samplics</b>	–	–	–	–	–	✓

Table 2: A comparison of packages and implemented methods

The **NonProbEst** is the most comprehensive package in the comparison. It offers various techniques, such as IPW or prediction approaches (e.g. model-calibrated). Users can choose several different settings for IPW weights, variable selection and can estimate variance using the leave-one-out Jackknife procedure. Unfortunately, the package is no longer developed (it was last updated in 2022) and some of the techniques are either outdated or have been shown to be inappropriate for non-probability samples. While the package contains functions designed for specific methods, it does not allow users to leverage the **survey** package for estimation. The **balance** package is solely dedicated to the PS approach. It assumes that the the population totals are known and the authors have implemented the 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** supports the use of unit-level data from a probability sample or the population, implements IPW, MI and DR estimators, and offers users the possibility of selecting variables but is still at a 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** supports the use of the **samplix** module (Diallo 2021). The **GJRM** package is the only package that offers functions to estimate sample selection models used widely for correcting selection bias in observational studies (including the 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 or conducting statistical inference. Finally, the MRP approach is implemented solely in the **rstanarm** package with variable selection specified by an appropriate prior.

The **nonprobsvy** package has several advantages over those presented above. Firstly, it implements state-of-the-art methods recently proposed in the literature, along with valid statistical inference procedures. Secondly, it offers other approaches, such as calibrated IPW (where PS weights match the population or estimated totals), NN and PMM matching, various IPW and DR estimators with the possibility of selecting link functions for logistic regression. Thirdly, it supports the functions included in 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, together with a key function to specify the approach and estimators. As far as we know, the **nonprobsvy** is the only software (open-access or commercial) that offers such functionalities.

The remaining part of the paper is structured as follows. Section 2 is dedicated to the theory of statistical inference based on non-probability samples. We provide the basic setup and introduce specific methods in separate subsections. We follow the notation used by Wu (2022) throughout the paper. Section 3 describes the main function and the package functionalities. Section 4 presents an empirical study showcasing the process of integrating data from the Polish Job Vacancy Survey with voluntary administrative data from the Central Job Offers Database in order to estimate the number of companies with at least one vacancy offered on a single shift. Section 5 presents classes and **S3Methods** implemented in the package. The paper ends with a summary and plans for future work. The Appendix contains Table 10 showing a list of symbols and Section B, which presents algorithms for selected MI estimators. The Replication Materials include additional codes for specific estimators described in the paper.

## 2. Methods for non-probability samples

### 2.1. Basic setup

Let  $U = \{1, \dots, N\}$  denote the target population consisting of  $N$  labelled units. Each unit  $i$  has an associated vector of auxiliary variables  $\mathbf{x}_i$  and the study (target) variable  $y_i$ . Let  $\{(y_i, \mathbf{x}_i), i \in S_A\}$  be a non-probability sample  $S_A$  of size  $n_A$  and let  $\{(\mathbf{x}_i, \pi_i^B), i \in S_B\}$  be a probability sample  $S_B$  of size  $n_B$ , where the only information known for all units in the

population refers to auxiliary variables  $\mathbf{x}$  and inclusion probabilities  $\pi^B$ . Each unit in the  $S_B$  sample 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 in the non-probability sample  $S_A$  and the probability sample  $S_B$ , respectively, which are defined for all units in the target population. Let  $\pi_i^A = P(R_i^A = 1 \mid \mathbf{x}_i, y_i) = P(R_i^A = 1 \mid \mathbf{x}_i)$  be propensity scores (PS), which characterize the  $S_A$  sample's inclusion and participation mechanisms. Unlike  $\pi_i^B$ , the  $\pi_i^A$  and  $d_i^A = 1/\pi_i^A$  are unknown. The description of the data is presented in a more concise form in Table 3.

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

Table 3: Two-sample setting

The goal is to estimate the 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  $\mu_y$ . Given the absence of a universally accepted method for achieving this objective, assumptions vary considerably, as outlined by Wu (2022). However, the following are the main assumptions that apply to all methods presented in this section:

- A1  $R_i^A$  and the study variable  $y_i$  are independent given the set of covariates  $\mathbf{x}_i$  (i.e.,  $(R_i^A \perp y_i) \mid \mathbf{x}_i$ ; the MAR mechanism).
- A2 All the units in the target population have non-zero PS, i.e.,  $\pi_i^A > 0$ ,  $i = 1, 2, \dots, N$  (i.e. no coverage error).
- A3 The indicator variables  $R_1, R_2, \dots, R_N$  are independent given the set of auxiliary variables  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$  (i.e. no clustering).

In addition, we assume no overlap between  $S_A$  and  $S_B$ , and no measurement error in  $y_i$  and the fact that values of  $\mathbf{x}_i$  are known. The setting presented in Table 3 can also be 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 the methods implemented in the package.

## 2.2. Prediction-based approach

### *Prediction estimators*

In the prediction approach the following semi-parametric model for the finite population is assumed:

$$E_{\xi}(y_i | \mathbf{x}_i) = m(\mathbf{x}_i, \boldsymbol{\beta}), \text{ and } V_{\xi}(y_i | \mathbf{x}_i) = v(\mathbf{x}_i) \sigma^2, \quad i = 1, 2, \dots, N, \quad (1)$$

where the mean function  $m(\cdot, \cdot)$  and the variance  $v(\cdot)$  have known forms and  $y_i$  are assumed to be independent of  $\mathbf{x}_i$ . The model (1) is assumed to hold for all units in the non-probability  $S_A$  sample. The parameters of the (1) can be estimated using the quasi maximum likelihood estimation method, which includes linear and non-linear models, such as generalized linear models (GLM). Let  $\boldsymbol{\beta}_0$  and  $\sigma_0^2$  be the true values of the model parameters  $\boldsymbol{\beta}$  and  $\sigma^2$  under the adopted model and  $\hat{\boldsymbol{\beta}}$  be the quasi maximum likelihood estimator of  $\boldsymbol{\beta}_0$ . Let  $m_i = m(\mathbf{x}_i, \boldsymbol{\beta}_0)$  and  $\hat{m}_i = m(\mathbf{x}_i, \hat{\boldsymbol{\beta}})$  be calculated for all units  $i = 1, \dots, N$ . Under this settings, as Wu (2022) notes, there are two commonly used prediction estimators:

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

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

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

where  $\mu_{\mathbf{x}} = N^{-1} \sum_{i=1}^N \mathbf{x}_i$  is the vector of the population means of the  $\mathbf{x}$  variables and  $\bar{\mathbf{x}}_A = n_A^{-1} \sum_{i \in S_A} \mathbf{x}_i$  is the vector of the simple means of  $\mathbf{x}$  from the non-probability  $S_A$  sample. 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 it only requires the 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  $\mu_{\mathbf{x}}$  is replaced by  $\hat{\mu}_{\mathbf{x}} = \hat{N}_B^{-1} \sum_{i \in S_B} d_i^B \mathbf{x}_i$  for (3) where  $\hat{N}_B = \sum_{i \in S_B} d_i^B$ .

### *Mass imputation estimators*

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

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

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 reduces to the estimators in (2).

There are several ways of imputing  $y_i^*$  and in the package we have implemented the following MI estimators: the semi-parametric approach based on 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 imputed with  $\hat{m}_i$  from the semi-parametric model, were studied by Kim, Park, Chen, and Wu (2021). In the **nonprobsvy** package, we account 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 of the MI-NN estimator for large non-probability samples (big data, i.e. covering a significant part of the target population) were studied by Yang *et al.* (2021). The basic idea of NN matching is as follows: 1) for each  $i$  unit in the probability sample  $S_B$  find a donor  $j$  (or donors) in the  $S_A$  sample based on some distance between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ ; 2) use the matched values  $y_j$  from  $S_A$  to impute missing  $y_i$  values in the probability sample  $S_B$ . Imputed values of  $y_i^*$  depend on the number of selected  $k$  neighbours: for  $k = 1$ , the closest one is selected, and for  $k > 1$ , one can calculate a simple average over a vector of selected  $y$  values. A detailed description of the procedure is presented in Algorithm 1 in the Appendix. The MI-NN estimator suffers from the curse of dimensionality, as shown by [...], i.e. asymptotic bias of the MI estimator increases as the number of covariates  $\mathbf{x}$  increases with a fixed  $k$  (Abadie and Imbens 2006; Yang and Kim 2020). The PMM approach has been proposed to overcome this issue.

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

### *Variance estimators for the prediction approach*

Variance of the MI estimators can be estimated analytically or using the bootstrap approach. The analytical estimator of the variance of the MI-GLM estimator proposed by Kim *et al.* (2021, p. 950) contains two components:  $\hat{V}_1$  (based on the information from both samples  $S_A$  and  $S_B$ ) and  $\hat{V}_2$  (based exclusively on the probability sample  $S_B$ ); for the MI-NN estimator Yang *et al.* (2021) proposed a variance estimator for large  $S_A$  samples, which reduces to the part for the probability sample  $S_B$  (i.e. the design-based variance estimator of the mean, which can easily be obtained from the **survey** package) and a version for smaller samples can be found in Chlebicki *et al.* (2024); with respect to the variance of MI-PMM estimators, Chlebicki *et al.* (2024) propose formulas which are the same as those for the MI-NN estimators. In the bootstrap approach each bootstrap replication  $b = 1, \dots, B$  consists of the following steps.

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 the `as.svrepdesign` function from the **survey** package).
2. Estimate  $\mu_{y,MI}^b$  using an appropriate approach (e.g. MI-GLM, MI-NN or MI-PMM).

After obtaining  $B$  bootstrap replicates, estimate variance using the following equation:

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

where  $\hat{\mu}_y$  is the mean estimated using either the 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) are available, we can estimate the 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). To estimate the variance of the MI-NN and MI-PMM estimators we only allow the bootstrap approach with known population means. Note that the current version of the **nonprobsvy** does not support the use of replicated weights in the probability sample  $S_B$  for any of the estimators discussed in this paper.

### 2.3. Inverse Probability Weighting

Inverse probability weighting (IPW), another popular estimation approach, involves estimating PS given by  $\pi_i^A = P(i \in S_A)$ . As in the case of the prediction-based approach, there are two variants of the IPW estimator, given by

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

where the  $\hat{\mu}_{y,IPW1}$  is a version of the Horvitz-Thompson estimator, and the  $\hat{\mu}_{y,IPW2}$  is the Hájek estimator, where the estimated population size is given by  $\hat{N}^A = \sum_{i \in S_A} (\pi_i^A)^{-1}$ . The use of this estimator with respect to non-probability samples is discussed by [Lee \(2006\)](#) and [Biffignandi and Bethlehem \(2021, chapter 13\)](#) and there are several approaches to using propensity scores along with alternative versions of the weights (cf. [Elliott and Valliant 2017, section 3](#)) In a recent article by [Chen et al. \(2020\)](#) dedicated to the properties of the estimators in (6), the authors proved their consistency and derived their closed form versions. [Wu \(2022, section 4.2\)](#) argue that the  $\hat{\mu}_{y,IPW2}$  estimator performs better than  $\hat{\mu}_{y,IPW1}$  even if the population size is known.

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

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

In practice, a function of this form cannot be used because not all units from the population are observed. A more realistic approach consists in using a reference probability sample  $S_B$ , which means that the second component of the (7) is replaced, yielding a pseudo log-likelihood function given by (8)



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

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

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

In general, pseudo score functions  $\mathbf{U}(\gamma)$  for true values of model parameters  $\gamma_0$  are unbiased under the joint *qp* randomization in the sense that  $E_{qp} \{\mathbf{U}(\gamma_0)\} = \mathbf{0}$ , which implies that the estimator  $\hat{\alpha}$  is *qp*-consistent for  $\gamma_0$  (Wu 2022).

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

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

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

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

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

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

which can be viewed as *calibrated* IPW and equation (11) only requires the knowledge of population totals for auxiliary variables  $\mathbf{x}$ . Moreover, the use of equation (11) yields a doubly robust estimator under the 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 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 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 is essentially the same as the one presented in (5), where  $\hat{\mu}_y$  is replaced by one of the estimators in (6).

## 2.4. Doubly robust approach

The IPW and MI estimators are suited to mis-specified models for PS and outcome regression models, respectively. The DR approach was proposed to improve robustness and efficiency (cf. [Robins, Rotnitzky, and Zhao 1994](#)). It incorporates a prediction model for the response variable  $y_i$  and a PS model for participation  $R_i^A$ . This approach is doubly robust in the sense that the DR estimator remains consistent even if one of the models is mis-specified. We need to consider a joint randomization approach involving a non-probability sample  $S_A$  and a probability sample  $S_B$  and DR inference is conducted within the  $qp$  or  $\xi p$  framework without specifying which one is correct. The general formula 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  $\pi_i^A$  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) assuming that the population size is either known or estimated:

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

and

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

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

[Chen et al. \(2020\)](#) suggested constructing the estimators in (12) or (13) based on the two models estimated separately, which is an attractive option, since once can specify a different number of variables for the propensity and outcome model. An alternative approach, proposed by [Yang, Kim, and Song \(2020\)](#) and similar to the one described by [Kim and Haziza \(2014\)](#), is discussed in the next subsection.

### *Minimization of the bias for doubly robust methods*

[Yang et al. \(2020\)](#) discussed variable selection for a high-dimensional setting and noted that the bias of the estimator, which can increase, cannot be controlled. Therefore, according to [Yang et al. \(2020\)](#), the idea is to develop equations that can be used to estimate the  $\beta$  and  $\gamma$  parameters based on the bias of the population mean estimator. In this way the parameters can be estimated in a single step, rather than in two separate steps. First, the authors derived

the bias of the  $\hat{\mu}_{DR}$ , assuming  $\mathbf{h}(\mathbf{x}, \gamma) = \mathbf{x}\pi(\mathbf{x}, \gamma)^{-1}$  for the IPW estimator, which is given by equation (14)

$$\begin{aligned} \text{bias}(\hat{\mu}_{DR}) = |\hat{\mu}_{DR} - \mu| &= \frac{1}{N} \sum_{i=1}^N \left\{ \frac{R_i^A}{\pi(\mathbf{x}_i, \gamma)} - 1 \right\} \{y_i - m(\mathbf{x}_i, \beta)\} \\ &+ \frac{1}{N} \sum_{i=1}^N \left( R_i^B d_i^B - 1 \right) m(\mathbf{x}_i, \beta) \end{aligned} \quad (14)$$

The goal of this approach is to minimize bias  $(\hat{\mu}_{DR})^2$ , which consists in solving the following system of equations:

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

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

In the **nonprobsvy** package we have implemented these approaches not only for  $\mathbf{h}(\mathbf{x}, \gamma) = \mathbf{x}_i \pi(\mathbf{x}_i, \gamma)^{-1}$  but also for  $\mathbf{h}(\mathbf{x}, \gamma) = \mathbf{x}$  and various link functions for the propensity score model. We also enable the user to select this estimation strategy with or without variable selection methods discussed in Section 2.5. As noted in the beginning, the choice of either (12) or (13) results in a different approach to estimating variance, which is discussed in the next subsection.

### *Variance estimators for the doubly robust approach*

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

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

## 2.5. Variable selection algorithms

Yang and Kim (2020) point out that it is crucial to use variable selection techniques during estimation, especially when dealing with high-dimensional non-probability samples. Variable

selection not only improves model stability and computational feasibility, but also reduces variance, which can increase when irrelevant auxiliary variables are included.

The most popular approaches described in the 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 for non-probability methods works in a similar way, with loss functions modified to account for external data sources, such as sample or population totals or averages. In particular, the technique consists of two steps: 1) we select the relevant variables using an appropriately constructed loss function (and possibly using the approach shown in (15) to obtain the final estimates of the model parameters); and 2) we construct the selected 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:

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

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

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

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

where  $R_i^A$  and  $R_i^B$  are indicator functions defining the fact of being included in the non-probability sample  $S_A$  and the probability sample  $S_B$ , respectively. Yang *et al.* (2020) only discussed the SCAD penalty and the  $\mathbf{h}(\mathbf{x}_i, \gamma) = \mathbf{x}_i$  function for the DR estimator. In the **nonprobsvy** package we have extended this approach to the first variant of  $\mathbf{h}(\mathbf{x}_i, \gamma)$ , shown in the first row of Table (4), and have allowed the user to select other link functions for the  $\pi_i^A$ , implemented other penalty functions and extended the possibility of selecting variables to MI and IPW estimators. In the next section we discuss how to define the approaches presented above.

### 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 the use of **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 include `data` as well as one of the following three `selection`, `outcome`, or `target` – depending on which method has been selected. In the case of `outcome` and `target` multiple  $y$  variables can be specified.

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

Table 5: A description of the `nonprob` function arguments

The package allows the user 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 from the **survey** package). The `nonprob` function is used to specify inference methods through the `selection` and `outcome` arguments.

If a `svydesign2` object is provided and the `selection` argument is specified, then the IPW estimators are used (by default parameters of the PS model employ (9)), if the `outcome` argument is specified, then the MI approach is used (the default option is the MI-GLM with the `gaussian` family) and if both are specified, then the DR approach is applied (parameters  $(\beta, \gamma)$  are estimated separately and the (13) is used). A particular inference method is selected through the `method_selection`, `method_outcome`, `family_outcome`, `control_selection` and `control_outcome` arguments. The variance estimation method is selected through the `control_inference` argument.

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

- `data` – a `data.frame` containing the non-probability sample.
- `X` – a `matrix` containing both samples,
- `y` – a `list` containing all variables declared in either the `target` or `outcome` arguments,
- `R` – a numeric `vector` informing about inclusion in the 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` containing confidence intervals for the mean,
- `outcome` – a `list` of results for each `outcome` model,
- `selection` – a `list` of results for the `selection` model,
- `svydesign` – a `svydesign2` object passed by the `svydesign` argument.

### 3.2. Controlling the 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 the Euclidean distance as 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 <code>control_out</code>
MI-GLM with the LASSO penalty and 5 folds	<code>nonprob(outcome = y1 ~ x1 + x2, data = df, svydesign=prob, control_outcome = control_out(penalty="lasso", folds = 5))</code>
MI-NN with the bd algorithm	<code>nonprob(outcome = y1 ~ x1 + x2, data = df, svydesign=prob, control_outcome = control_out(treetype = "bd"))</code>
MI-PMM A with $k = 3$	<code>nonprob(outcome = y1 ~ x1 + x2, data = df, svydesign=prob, control_outcome = control_out(k=3,predictive_match=2))</code>

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 MLE given by (9) or GEE defined in (11) through the `est_method_sel` argument, specify the  $h(\cdot, \cdot)$  function through the `h` argument, specify the optimizer (`optimizer` argument) and which variable selection method should be applied (using different penalty functions like SCAD, lasso, and MCP by specifying



the `penalty` argument) along with parameters (e.g. the number of folds through the `nfolds` argument). The parameters of the PS for the calibrated IPW is estimated 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 <code>control_sel</code>
Calibrated IPW	<code>nonprob(selection = ~ x1 + x2, target = ~y1, data = df, svydesign = prob, control_selection = control_sel(est_method_sel="gee"))</code>
IPW with the MCP penalty and 5 folds	<code>nonprob(selection = ~ x1 + x2, target = ~y1, data = df, svydesign = prob, control_selection = control_sel(penalty="MCP", nfolds=5))</code>

Table 7: Example declarations of the IPW estimators

### 3.3. Controlling variance estimation

Finally, the `control_inf` function configures the parameters for variance estimation in the `nonprob` function. It allows users to specify whether the analytical or bootstrap approach should be used (the `var_method` argument), whether the variable selection method should be applied (the `vars_selection` argument) and what type of bootstrap should be applied for the probability sample (the `rep_type` argument). This function is also used to specify the inference procedure for the DR approach: 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` function for the IPW and DR estimators.

Estimator	Declaration with the <code>control_sel</code>
Calibrated IPW with variable selection, boot- strap and $B = 50$	<code>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))</code>
The DR with the SCAD penalty, 5 folds and bias correction	<code>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))</code>

Table 8: Example declarations of the IPW estimators

In the next sections we present a case study illustrating the process of integrating a non-probability sample with a reference probability sample. We present various estimators and compare them. Finally, we describe more advanced options available in the package.

## 4. Data analysis example

### 4.1. Description of the data

The package can be installed in the standard manner using:

```
R> install.packages("nonprobsvy")
```

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 is 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	0	14	1
2	j_2	0	L	0	24	6
3	j_3	0	L	R.S	14	1
4	j_4	0	L	R.S	14	1
5	j_5	0	L	R.S	22	1
6	j_6	0	M	R.S	26	1

As the package leverages 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 how to handle 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 naïve, 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 two 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 interval for population 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  $y \sim x_1 + x_2 + \dots + x_k$ ) and multiple outcomes (as  $y_1 + y_2 + y_2 \sim x_1 + x_2 + \dots + x_k$ ). 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 be used user needs to specify both the `selection` and `outcome` arguments. These formulas can be specified with the same or varying number of auxiliary variables. We also allow, similarly 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> 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> 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()
```

#### 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`

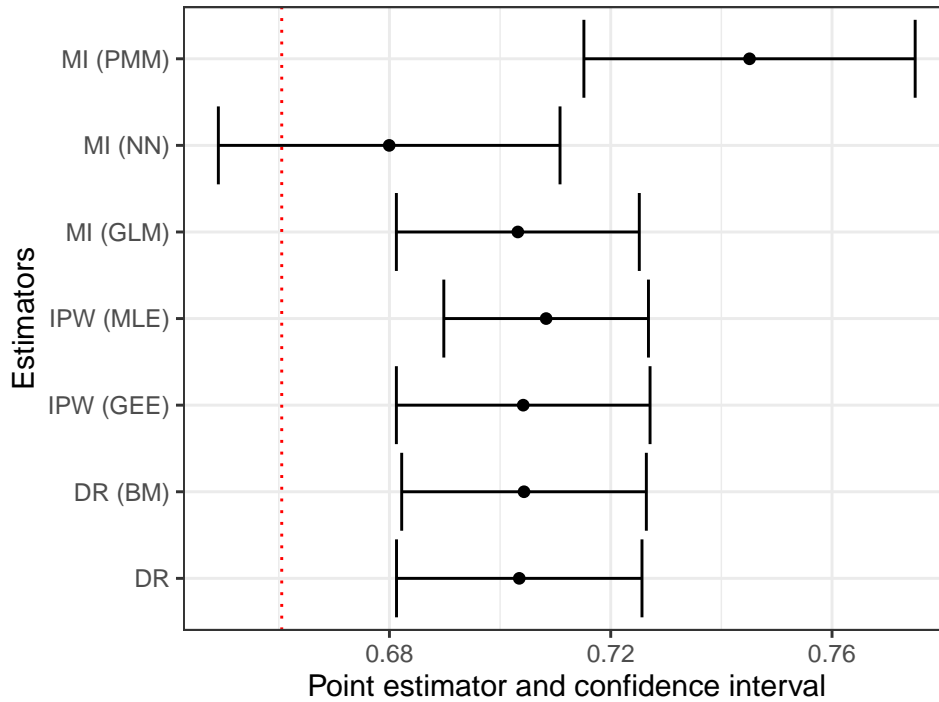


Figure 1: Comparison of estimates of the share of job vacancies offered on a single-shift

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 0.011179381
```

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,]      0.7203402
[2,]      0.7172369
[3,]      0.7180005
```

### *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  $\lambda$  values. Default is 50.
- `lambda_min` – The smallest value for  $\lambda$ , 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  $\lambda$  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

In the package we have created the main class `nonprobsvy` and a supplementary class `summary_nonprobsvy`. All `S3methods` implemented can be obtained by using `methods(class="nonprobsvy")`. For instance, the `check_balance` function already mentioned in the case study allows to assess the balance by checking how the PS weights reproduce known or estimated population totals, or the `nobs` function returns the sample size of the probability and non-probability samples:

```
R> nobs(dr_est1)

prob nonprob
6523    9344
```

Table 9 presents methods implemented for the `nonprobsy` class. On purpose we did not implement many methods as the goal of the package is to provide point and interval estimates. If a user is interested in assessing the quality of the models or covariate balance should use existing R packages.

## 6. Summary and future work

The `nonprobsvy` package provides a comprehensive R software for addressing inference challenges with non-probability samples by integrating them with probability samples or known

Function	Description
<code>check_balance</code>	aaa;
<code>confint</code>	aaa
<code>nobs</code>	
<code>pop_size</code>	
<code>summary</code>	
<code>logLik</code> , <code>AIC</code> , <code>BIC</code> ,	
<code>deviance</code>	
<code>residuals</code> , <code>hatvalues</code> ,	it works exactly like <code>glm</code> counterparts.
<code>cooks.distance</code> ,	
<code>print</code> , <code>vcov</code>	

Table 9: `S3Methods` implemented in the **nonprobsvy**

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, and doubly robust methods, each designed to correct selection bias by leveraging auxiliary data. By providing a unified framework and integration with the **survey** package, the **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 selections mechanism, developing theory for maintaining consistency with calibration weights, and supporting multiple non-probability samples for data integration from various sources. Further methodological extensions under consideration include empirical likelihood approaches for doubly/multiply robust estimation, integration of machine learning methods like debiased/double machine learning from causal inference, handling measurement error in big data variables, and expanding the bootstrap approach beyond simple random sampling with replacement.

The package will also be extended to work with the `svyrep.design` class from the **survey** package and the **svrep** package. These developments will enhance **nonprobsvy**'s capabilities for handling complex survey data structures and modern estimation challenges.

## 7. Acknowledgements

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

Łukasz Chrostowski is the main developer and maintainer of the package. Parts of this paper are based on Łukasz's Master's thesis (available at <https://github.com/ncn-foreigners/graduation-theses>). Piotr Chlebicki contributed to the package and implemented MI-PMM estimators. Maciej Beręsewicz was responsible for the initial idea and the design of the package, testing, reviewing and contributing to the source code and prepared the manuscript.

We would like to thank . . .

## 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
$\mathbf{x}_i$	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 $S_B$
$\pi_i^A$	Propensity score for unit $i$ in non-probability sample
$\pi_i^B$	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
$\mu$	Population mean of target variable $y$
$\mu_{\mathbf{x}}$	Population means of auxiliary variables $\mathbf{x}$
$m(\mathbf{x}_i, \boldsymbol{\beta})$	Semiparametric model for outcome variable
$\dot{m}(\mathbf{x}_i, \boldsymbol{\beta})$	First derivative of the $m(\mathbf{x}_i, \boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$
$\pi(\mathbf{x}_i, \boldsymbol{\gamma})$	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
$\hat{\mu}_{\mathbf{x}}$	Estimator for the population means of auxiliary variables $\mathbf{x}$
$\bar{\mathbf{x}}_A$	A vector of the sample means of the auxiliary variables $\mathbf{x}$ 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 10: 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  $\hat{\nu}(i) = \arg \min_{j \in S_A} d(\mathbf{x}_i, \mathbf{x}_j)$ .
- 2: If  $k > 1$ , then

$$\hat{\nu}(i, z) = \arg \min_{j \in S_A \setminus \bigcup_{t=1}^{z-1} \{\hat{\nu}(i, t)\}} d(\mathbf{x}_i, \mathbf{x}_j)$$

i.e.  $\hat{\nu}(i, z)$  is  $z$ -th nearest neighbour from the sample  $S_A$ ;

- 3: For each  $i \in S_B$ , calculate the imputed value as

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


---

---

**Algorithm 2:** Mass imputation using predictive mean matching variant:  $\hat{y} - \hat{y}$  matching

---

- 1: Estimate regression model  $m(\mathbf{x}, \beta)$  parameters;
- 2: Predict

$$\hat{y}_i = m(\mathbf{x}_i, \hat{\beta}), \hat{y}_j = m(\mathbf{x}_j, \hat{\beta})$$

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

$$\hat{\nu}(i) = \arg \min_{j \in S_A} d(\hat{y}_i, \hat{y}_j).$$

- 3: If  $k > 1$ , then:

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

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

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

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


---

---

**Algorithm 3:** Mass imputation using predictive mean matching variant:  $\hat{y} - y$  matching

---

- 1: Estimate regression model  $m(\mathbf{x}, \beta)$  parameters.;
- 2: Predict

$$\hat{y}_i = m(\mathbf{x}_i, \hat{\beta})$$

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

$$\hat{\nu}(i) = \arg \min_{j \in S_A} d(\hat{y}_i, y_j)$$

;

- 3: If  $k > 1$ , then:

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

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

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


---

## 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.” *Forthcomming to the 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 University.
- 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). “samplix: 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/jrssa/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](https://doi.org/10.1214/16-STS597). URL <https://doi.org/10.1214/16-STS597>.
- Statistics Poland (2021). “The Demand for labour: 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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1111/rssb.12354). URL <https://academic.oup.com/jrsssb/article/82/2/445/7056072>.

**Affiliation:**

Łukasz Chrostowski  
Adam Mickiewicz University  
First line  
Second line  
E-mail: [lukchr@st.amu.edu.pl](mailto: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](mailto: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. Niepodległości 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](mailto:maciej.beresewicz@ue.poznan.pl)  
URL: <https://github.com/BERENZ>, <https://ue.poznan.pl/en/people/dr-maciej-beresewicz/>