Journal of Statistical Software

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

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

Łukasz Chrostowski
Adam Mickiewicz University

Piotr Chlebicki
Stockholm University

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

Abstract

The abstract of the article.

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

1. Introduction

With the availability of large sets of administrative data, voluntary internet panels, social media and big data, inference with non-probability samples is being heavily studied in the statistical literature Beaumont (2020), Elliott and Valliant (2017), Beręsewicz (2017), Citro (2014). Because of their non-statistical character and unknown sampling mechanism, these sources cannot be used directly for estimating population characteristics.

Several inference approaches have been proposed in the literature with respect to data from non-probability samples, which either involve data integration with population level data or probability samples from the same population (for recent review see Wu (2022)).

Although probability samples are still the most popular standard among statisticians, the cost of obtaining them, in terms of time or capital, motivates the use of non-probability samples, which have to overcome other challenges. The first is that such samples generally do not represent the whole population, as can be said of probability samples. Another problem is the lack, or rather the ignorance, of the mechanism for selecting individuals for this type of sample, which does not allow the substantial use of existing statistical methods. For this reason, many different techniques have been proposed in the literature for integrating data in order to infer from available sources of different structural character.

It should be noted that there are several packages that allow the correction of selection bias in nonprobability samples, such as:

- **GJRM** (Marra and Radice (2023)) supports generalized joint regression modeling, enabling users to fit complex joint models for multivariate outcomes often needed in handling selection bias in nonprobability samples.
- **NonProbEst** (Luis Castro Martín and del Mar Rueda (2020)) provides methods specifically designed for estimating population parameters from nonprobability samples, employing approaches like propensity score adjustment and calibration weighting to correct for sample selection bias.
- **sampling** (Tillé and Matei (2021)) offers tools for implementing various sampling designs, including stratified, cluster, and unequal probability sampling, which are essential for creating representative samples and reducing selection bias in survey data analysis.

However, these packages do not implement state-of-the-art approaches recently proposed in the literature: Chen, Li, and Wu (2020), Yang, Kim, and Song (2020), Wu (2022) nor do they use the survey package Lumley (2004) for inference.

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

Table 1 shows the basic characteristics of each of the samples described. In particular, what are the advantages and disadvantages of each type of sample with respect to population coverage, bias, variance, costs, and the selection mechanism for observations into the samples.

2. Methods for non-probability samples

2.1. Basic setup

Let $U = \{1, ..., N\}$ denote the target population consisting of N labelled units. Each unit i has an associated vector of auxiliary variables $\boldsymbol{x}_i$ (a realisation of the random vector $\boldsymbol{X}_i$

Table 1: Probability and non-probability samples

Factor	**Probability sample**	**Non-probability sample**
Selection	Sampling design	Auto-selection
Coverage	Typically good	Certain groups are excluded
Bias	Typically smaller	Large or very large
Variance	Typically larger	Small, or very small
Cost	Large or very large	Typically small

Table 2: Two Sample Setting

Sample		Auxiliary Variables $\boldsymbol{X}$	Target Variable Y	Design (d) or Calibration (w) Weights
S_A (non-probability)	1	✓	✓	?
	...	✓	✓	?
	n_A	✓	✓	?
S_B (probability)	$n_A + 1$	✓	?	✓
	...	✓	?	✓
	$n_A + n_B$	✓	?	✓

in the super-population) and the study variable y_i (a realisation of the random variable Y_i in the super-population). Let $\{(y_i, \boldsymbol{x}_i), i \in S_A\}$ be a dataset of a non-probability sample of size n_A and let $\{(\boldsymbol{x}_i, \pi_i), i \in S_B\}$ be a dataset of a probability sample of size n_B, where only information about variables $\boldsymbol{X}$ and inclusion probabilities π (which in the super population model are also considered to be random variables) are available. Let R_i be an indicator of inclusion into non-probability sample. Each unit in the sample S_B has been assigned a design-based weight given by $d_i = 1/\pi_i$.

The goal is to estimate a finite population mean $\mu_y = \dfrac{1}{N}\sum_{i=1}^{N} y_i$ of the target variable Y. As values of y_i are not observed in the probability sample, it cannot be used to estimate the target quantity. Instead, one could try combining the non-probability and probability samples to estimate μ_y. In this paper we do not consider modifications for the possibly occurring overlap. The above description of the data is presented in a more concise form in Table 2.

2.2. Mass Imputation estimators

Imputation refers to the process of replacing missing or incomplete data with substituted values. The goal of imputation is to allow for more complete data analysis, as many statistical methods require complete datasets. Common imputation techniques include:

- Mean imputation: where missing values are replaced by the mean of the observed data for that variable.
- Median imputation: where the median value of the observed data is used.
- Regression imputation: where missing values are estimated based on a regression

model built from other available data.

Imputation helps prevent data bias and maintains dataset size, ensuring that missing data points do not skew analysis results. It is particularly useful when data is missing at random or when only a small portion of the data is missing. Mass imputation is the application of imputation techniques to an entire dataset where many observations have missing values for the given variable. Kim, Park, Chen, and Wu (2021), Yang, Kim, and Hwang (2021), Chlebicki, Chrostowski, and Beręsewicz (2024) propose the following imputation strategies as:

- Model based apprach (GLM),
- Nearest neigbour imputation (NN),
- Predictive mean mathing (PMM).

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

By assumptions (Table 2), we do not know the value of the dependent variable Y for the units in the probability sample. In this case, the method will be to impute the values of the explanatory variable for all units in the probability sample. We therefore treat the non-probability sample as a training set that is used to build the imputation model. In this subsection, we distinguish three main methods of mass imputation based on linear models and the k-nearest neighbours algorithm. Other popular methods for estimating the variable Y from the variable $\boldsymbol{X}$ can also be considered, e.g. machine learning models such as random forests or neural networks.

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

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

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

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

Generalized Linear Models

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

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

for a certain p–dimensional vector $\boldsymbol{\beta}_0$ and a known m function from a given class of mean functions for generalized linear models for which we can distinguish the following cases

1. **Y variable is continous (linear model):**

$$m\left(\boldsymbol{x}_i, \boldsymbol{\beta}_0\right) = \boldsymbol{x}_i^T \boldsymbol{\beta}_0$$

2. **The variable Y represents a discrete variable (*count data*):**

$$m\left(\boldsymbol{x}_i, \boldsymbol{\beta}_0\right) = \exp\left(\boldsymbol{x}_i^T \boldsymbol{\beta}_0\right)$$

3. **The variable Y is binary (logistic model):**

$$m\left(\boldsymbol{x}_i, \boldsymbol{\beta}_0\right) = \frac{\exp\left(\boldsymbol{x}_i^T \boldsymbol{\beta}_0\right)}{\exp\left(\boldsymbol{x}_i^T \boldsymbol{\beta}_0\right) + 1}$$

According to the model described, we have

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

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

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

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

Algorithm 1 Mass imputation based on a generalized linear model

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

2.3. Nearest Neighbour Algorithm

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

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

```
1: If $k = 1$, then for each $i \in S_B$ match $\hat{\nu}(i)$ such that $\hat{\nu}(i) = \arg\min_{j \in S_A} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$.
2: If $k > 1$, then
       $\hat{\nu}(i,z) = \arg\min_{j \in S_A \setminus \bigcup_{t=1}^{z-1} \{\hat{\nu}(i,t)\}} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$
   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
       $\hat{y}_i = \frac{1}{k} \sum_{t=1}^{k} y_{\hat{\nu}(i,t)}$.
```

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

2.4. Predictive Mean Matching

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

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

```
1: Estimate regression model $\mathbb{E}[Y|\boldsymbol{X}=\boldsymbol{x}]=m(\boldsymbol{x},\boldsymbol{\beta})$.
2: Impute
       $\hat{y}_i = m\left(\boldsymbol{x}_i,\hat{\boldsymbol{\beta}}\right), \hat{y}_j = m\left(\boldsymbol{x}_j,\hat{\boldsymbol{\beta}}\right)$
   for $i \in S_B, j \in S_A$ and assign each $i \in S_B$ to $\hat{\nu}(i)$, where
       $\hat{\nu}(i) = \underset{j \in S_A}{\arg\min} \|\hat{y}_i - \hat{y}_j\|$
   or
       $\hat{\nu}(i) = \underset{j \in S_A}{\arg\min}\, d\left(\hat{y}_i, \hat{y}_j\right)$
   if $d$ is not induced by the norm.
3: If $k > 1$, then:
       $\hat{\nu}(i,z) = \underset{j \in S_A \setminus \bigcup_{t=1}^{z-1}\{\hat{\nu}(i,t)\}}{\arg\min} d\left(\hat{y}_i, \hat{y}_j\right)$
   e.g., $\hat{\nu}(i,z)$ is $z$-th nearest neighbor from a sample.
4: For $i \in S_B$, calculate imputation value as
       $\hat{y}_i = \frac{1}{k}\sum_{t=1}^{k} y_{\hat{\nu}(i,t)}$.
```

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

```
1: Estimate regression $\mathbb{E}[Y|\boldsymbol{X}=\boldsymbol{x}]=m(\boldsymbol{x},\boldsymbol{\beta})$.
2: Impute $\hat{y}_i = m\left(\boldsymbol{x}_i,\hat{\boldsymbol{\beta}}\right)$ for $i \in S_B$ and assign each $i \in S_B$ do $\hat{\nu}(i)$, where $\hat{\nu}(i) = \underset{j \in S_A}{\arg\min} \|\hat{y}_i - y_j\|$ or $\hat{\nu}(i) = \underset{j \in S_A}{\arg\min}\, d\left(\hat{y}_i, y_j\right)$ if $d$ not induced by the norm.
3: If $k > 1$, then:
       $\hat{\nu}(i,z) = \underset{j \in S_A \setminus \bigcup_{t=1}^{z-1}\{\hat{\nu}(i,t)\}}{\arg\min} d\left(\hat{y}_i, y_j\right)$.
4: For each $i \in S_B$ calculate imputation value as
       $\hat{y}_i = \frac{1}{k}\sum_{t=1}^{k} y_{\hat{\nu}(i,t)}$.
```

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

2.5. Inverse Probability Weighting estimators

The main disadvantage of non-probability sampling is the unknown selection mechanism for a unit to be included in the sample. This is why we talk about the so-called "biased sample' ' problem. The inverse probability approach is based on the assumption that a reference probability sample is available and therefore we can estimate the propensity score of the selection mechanism. In recent years, a number of articles have addressed this issue. Chen *et al.* (2020) propose maximum likelihood estimation approach for estimating propensity scores for selection mechanism. Wu (2022) present the approach based on generalized estimating equations, this method is also mentioned in Yang *et al.* (2020). On the other hand calibration approach for quantiles was explained Beręsewicz and Szymkowiak (2024) and Sant'Anna, Song, and Xu (2022) present the approach based on maximize the covariate distribution balance among different treatment groups.

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

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

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

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

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

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

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

Maximum Likelihood Estimation

Consider the following likelihood function

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

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

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

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

Generalized Estimating Equations

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

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

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

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

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

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

The advantage of this method is the ability to estimate with global values of the variables (e.g. from external sources) instead of a probability sample. Note that this is allowed by the second form of the $\mathbf{G}$ function. Its second term is nothing more than the estimated sums of the $\boldsymbol{x}$ variables. On the other hand, empirical studies suggest that the process of solving this type of equation may be less stable than the maximum likelihood method. In other words, the iterative algorithm for finding zeros that satisfy equation (7) may not converge. In the `nonprobsvy` package the propensity scores can modelled using three different link functions: logistic, complementary log-log and probit. The logistic regression is the most commonly used for modelling probabilities. This method is based on the so-called sigmoidal function and for our settign it has the form

$$\pi_i^A = \pi(\boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{\exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}_0\right)}{1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}_0\right)} \tag{8}$$

It satisfies certain properties to model the probability. For our scheme, this will be the probability of belonging to the non-probability sample.

Another approach to modelling binary variables and also probabilities is probit regression. It is based on the standard normal distribution. Probability density function is given by the following function

$$\frac{\partial \Phi(t)}{\partial t} = \phi(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}t^2\right)$$

and the probability is modelled as

$$\pi_i^A = \pi(\boldsymbol{x}_i, \boldsymbol{\theta}) = \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta}).$$

Regression with the cloglog model is particularly useful if we are modelling rare phenomena, i.e. the probabilities will oscillate around the values 1 and 0. Compared to the sigmoidal function and the distribution, the cloglog function is more asymmetric towards the value 0.5. The model can be written as

$$\pi_i^A = \pi(\boldsymbol{x}_i, \boldsymbol{\theta}) = 1 - \exp\Big(-\exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big)\Big).$$

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

Table 3: MLE Functions and Gradients for Different Link Functions

Link	GEE Jacobian $h = \boldsymbol{x}_i$	GEE Jacobian $h = \boldsymbol{x}_i \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^{-1}$
`logit`	$-\sum_{i \in S_B} \frac{1}{\pi_i^B} \pi_i^A\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)\left(1-\pi_i^A\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)\right) \boldsymbol{x}_i \boldsymbol{x}_i^\top$	$-\sum_{i \in S_A} \frac{1-\pi_i^A\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)}{\pi_i^A\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)} \boldsymbol{x}_i \boldsymbol{x}_i^\top$
`probit`	$-\frac{1}{N} \sum_{i \in S_B} \frac{\dot{\pi}_i^A\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)}{\pi_i^B} \boldsymbol{x}_i \boldsymbol{x}_i^\top$	$-\frac{1}{N} \sum_{i \in S_A} \frac{\dot{\pi}_i^A\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)}{\left(\pi_i^A\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)\right)^2} \boldsymbol{x}_i \boldsymbol{x}_i^\top$
`cloglog`	$-\frac{1}{N} \sum_{i \in S_B} \frac{1-\pi_i^A\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)}{\pi_i^B} \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) \boldsymbol{x}_i \boldsymbol{x}_i^\top$	$-\frac{1}{N} \sum_{i \in S_A} \frac{1-\pi_i^A\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)}{\left(\pi_i^A\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)\right)^2} \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) \boldsymbol{x}_i \boldsymbol{x}_i^\top$

Table 4: GEE Jacobians for Different Link Functions

2.6. Doubly Robust estimators

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

The proposed estimation procedure addresses the challenge of combining data from non-probability and probability survey samples. Traditional semiparametric models, often applied to such problems, are not directly usable in this context due to the distinct nature of the two samples. Instead, a joint randomization framework is employed, integrating semiparametric models for propensity scores with outcome regression for the nonprobability sample and design-based inference from the probability sample. This framework leads to a doubly robust (DR) estimation approach, which is effective in the presence of model misspecifications.

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

Joint Randomization Approach

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

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

$$\hat{\mu}_{\mathrm{DR}} = \frac{1}{\hat{N}^{\mathrm{A}}} \sum_{i \in S_{\mathrm{A}}} d_i^{\mathrm{A}} \left\{ y_i - m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right)\right\} + \frac{1}{\hat{N}^{\mathrm{B}}} \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right), \tag{9}$$

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

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

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

Minimization of the bias for doubly robust methods

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

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

$$\begin{aligned}\operatorname{bias}(\hat{\mu}_{DR}) &= \mid \hat{\mu}_{DR} - \mu \mid \\ &= \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{R_i^A}{\pi_i^A\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}\right)} - 1\right\}\left\{y_i - m\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}\right)\right\} \\ &+ \frac{1}{N}\sum_{i=1}^{N}\left(R_i^B d_i^B - 1\right) m\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}\right)\end{aligned} \tag{10}$$

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

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

where

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

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

$$\begin{pmatrix} \sum_{i=1}^{N} R_i^A \left\{\frac{1}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} - 1\right\}\left\{y_i - m\left(\boldsymbol{x}_i, \boldsymbol{\beta}\right)\right\}\boldsymbol{x}_i \\ \sum_{i=1}^{N} \frac{R_i^A}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})}\frac{\partial m(\boldsymbol{x}_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} - \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} \frac{\partial m(\boldsymbol{x}_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \end{pmatrix} = \mathbf{0}, \tag{11}$$

which can be solved using Newton–Raphson optimization method.

3. Package contents and implementation

All of the methods described in this paper have been implemented in the R package **nonprobsvy** Łukasz Chrostowski and Beręsewicz (2024). In this chapter, we will show you how to use the main `nonprob` function of the package and what its main features are. The package has been written to be as compatible as possible with the survey package for probablistic inference. Namely, the first step to use the nonprobsvy package is to define an object using the svydesign function that stores the probability sample `data.frame` and other objects, such as design weights. This is a negligible step if, instead of the probability sample, we have access to the values of the vector of sums of variables in the population. It is also worth mentioning that in order to speed up the calculations in the case of variable selection, part of the package, or more precisely the whole variable selection algorithm, was written in `C++` using the **Rcpp** (Eddelbuettel, Francois, Allaire, Ushey, Kou, Russell, Ucar, Bates, and Chambers (2024)) package, which allows the C++ code to be called in the R environment. Moreover, the package is supported by other R packages such as **foreach** Folashade Daniel Hong Ooi and Weston (2023) (looping construct), **maxLik** Henningsen and Toomet (2023) (maximum likelihood estimation), **Matrix** Bates, Maechler *et al.* (2023) (matrix operations), **MASS** Ripley, Venables *et al.* (2023) (statistical functions and datasets), **ncvreg** Breheny and Huang (2023) (regularization methods), **mathjaxr** Epskamp (2023) (rendering equations in documentation), **nleqslv** Groemping (2023) (solving nonlinear equations), and **doParallel** Steve Weston and Tenenbaum (2022) (parallel computing).

3.1. Usage

```
R> nonprob(
+   data,
+   selection = NULL,
+   outcome = NULL,
+   target = NULL,
+   svydesign = NULL,
+   pop_totals = NULL,
+   pop_means = NULL,
+   pop_size = NULL,
+   method_selection = c("logit", "cloglog", "probit"),
+   method_outcome = c("glm", "nn", "pmm"),
+   family_outcome = c("gaussian", "binomial", "poisson"),
+   subset = NULL,
+   strata = NULL,
+   weights = NULL,
+   na_action = NULL,
+   control_selection = controlSel(),
+   control_outcome = controlOut(),
+   control_inference = controlInf(),
+   start_selection = NULL,
+   start_outcome = NULL,
+   verbose = FALSE,
+   x = TRUE,
+   y = TRUE,
+   se = TRUE,
+   ...
+ )
```

3.2. Arguments

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

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

Table 5: `nonprob` function arguments description

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

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

Suppose we have two data sets, the first *nonprob* containing individuals from the non-probability sample. As assumed, this set contains information on k variables $\mathbf{x}$, e.g. sex, income, etc., and the explanatory variable y. In addition, there is a probability sample defined using the survey package and the svydesign function, containing design weights and $\mathbf{x}$ variables, but no y variable.

In the case of a **mass imputation** estimator, the function should be defined as follows

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

As can be seen, we have defined a formula for the imputation of the explanatory variable similar to the function glm. We have also specified the datasets in the arguments data (non-probability sample) and svydesign (probability sample). Finally, we have provided information about the mass imputation method (glm) and the type of explanatory variable (continuous variable). Let us now look at the **propensity score** estimator

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

As you can see, three new arguments have appeared - `selection` and `target` are responsible for the formulas for modelling the inclusion model and for defining the variable for which we calculate the population mean. In addition, in the `method_selection` argument, we specify the name of the link function that models the probability of inclusion in the non-probability sample. For the **doubly robust estimator** call it is as follows

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

In this case the `target` is not needed as we define `selection` and `outcome` arguments. We also provide details on mass imputation and propensity score models to obtain a doubly robust estimator. Importantly, arguments such as `method_outcome`, `method_selection` or `family_outcome` (and a few others) take default values described in more detail in the package documentation. According to the description of the control functions, we can enforce that the estimation using the DR method is preceded by variable selection using the SCAD method for the IPW part and the MCP method for the MI part.

```
R> nonprob(
+   selection = ~ x1 + x2 + ... + xk,
+   outcome = y ~ x1 + x2 + ... + xk,
+   data = nonprob,
+   svydesign = prob,
+   method_outcome = "glm",
+   family_outcome = "gaussian",
+   method_selection = "logit",
+   control_selection = controlSel(penalty = "SCAD"),
+   control_outcome = controlSel(penalty = "MCP"),
+   control_inference = controlInf(vars_selection = TRUE),
+   verbose = TRUE
+ )
```

In the control function concerning the selection mechanism for the non-probabilistic sample, we can also choose the weighting estimation method. The default value mle can be changed to gee along with the appropriate h function (corresponding to $h(\boldsymbol{x}_i, \boldsymbol{\theta}) = \boldsymbol{x}_i$).

```
R> nonprob(
+   selection = ~ x1 + x2 + ... + xk,
+   outcome = y ~ x1 + x2 + ... + xk,
+   data = nonprob,
+   svydesign = prob,
+   method_outcome = "glm",
+   family_outcome = "gaussian",
+   method_selection = "logit",
+   control_selection = controlSel(est_method_sel = "gee", h = 2))
```

Mass imputation methods are defined in the argument `method_outcome`. In the control function, we can set the parameters of a given method, e.g., the number of nearest neighbours in the NN algorithm.

```
R> nonprob(
+   selection = ~ x1 + x2 + ... + xk,
+   outcome = y ~ x1 + x2 + ... + xk,
+   data = nonprob,
+   svydesign = prob,
+   method_outcome = "nn",
+   family_outcome = "gaussian",
+   method_selection = "logit",
+   control_outcome = controlOut(k = 3)
+ )
```

As the final example, we want to perform variable selection using the SCAD method for the IPW part, the default method for the MI part (also SCAD), choose bias minimization as the parameter estimation method for the DR estimator, and set the variance calculation method to bootstrap.

```
R> nonprob(
+   selection = ~ x1 + x2 + ... + xk,
+   outcome = y ~ x1 + x2 + ... + xk,
+   data = nonprob,
+   svydesign = prob,
+   method_outcome = "glm",
+   family_outcome = "gaussian",
+   method_selection = "logit",
+   control_selection = controlSel(penalty = "SCAD"),
+   control_inference = controlInf(vars_selection = TRUE,
+                                  bias_correction = TRUE,
+                                  var_method = "bootstrap"),
+   verbose = TRUE
+ )
```

The result of the function call will be an object of the class **nonprobsvy** containing a list of elements related to the estimation, i.e. the value of the estimated parameters, the mean and its standard deviation. On such an object, we can call the `summary` method, familiar to users of the R language. The result will look like this

```
R> Call:
+ nonprob(data = nonprob_df, selection = ~x1 + x2 + x3 + x4, outcome = y30 ~
+     x1 + x2 + x3 + x4, svydesign = svyprob, method_selection = "logit")
+
+ -------------------------
+ Estimated population mean: 9.374 with overall std.err of: 0.3955
+ And std.err for nonprobability and probability samples being respectively:
+ 0.364 and 0.1546
+
+ 95% Confidence interval for population mean:
+     lower_bound upper_bound
+ y30    8.598809    10.14916
+
+ Based on: Doubly-Robust method
+ For a population of estimate size: 20200.83
+ Obtained on a nonprobability sample of size: 950
+ With an auxiliary probability sample of size: 1001
+ -------------------------
+
+ Regression coefficients:
+ -----------------------
+ For glm regression on outcome variable:
+             Estimate Std. Error z value  P(>|z|)
+ (Intercept) -0.44155    1.05815  -0.417 0.676469
+ x1           1.20976    0.72852   1.661 0.096802 .
+ x2           1.50153    0.44714   3.358 0.000785 ***
+ x3           1.35890    0.28206   4.818 1.45e-06 ***
+ x4           1.17264    0.08029  14.606  < 2e-16 ***
+ ---
+ Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
+
+ -----------------------
+ For glm regression on selection variable:
+              Estimate Std. Error z value  P(>|z|)
+ (Intercept) -4.480634   0.113868 -39.349  < 2e-16 ***
+ x1          -0.028191   0.074889  -0.376    0.707
+ x2           0.277772   0.044721   6.211 5.26e-10 ***
+ x3           0.148772   0.029746   5.001 5.69e-07 ***
+ x4           0.172832   0.008622  20.046  < 2e-16 ***
+ -------------------------
+
+ Weights:
+    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
+   1.143  11.180  19.502  21.264  28.089  79.119
+ -------------------------
+
+ Covariate balance:
+ (Intercept)          x1          x2          x3          x4
+    72.18047  -425.80885  -584.02397  -351.61222  1508.57831
+ -------------------------
+
+ Residuals:
+     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
+ -0.31329 -0.04205 -0.01799  0.42335  0.94734  0.98736
+
+ AIC: 7255.979
+ BIC: 7283.86
+ Log-Likelihood: -3622.99 on 1946 Degrees of freedom
```

4. Practical examples

We apply our methods to integrate administrative and survey data about job vacancies in Poland. The goal is to estimate the percentage of single shift job vacancies according to available data sources. We defined our outcome variable Y as follows: *whether the vacancy notice was for single-shift work.* Now we present the description of data used.

The first source is the Job Vacancy Survey (JVS, also known as the Labour Demand Survey) with a sample of 6523 units. The data include "the number of employed persons, as well as the number and structure of job vacancies, including newly created and vacant positions reported to employment offices. Information on newly created and eliminated jobs" (Central Statistical Office of Poland). The survey is conducted using a representative method. The sample is drawn separately for units employing more than 9 people and for units employing up to 9 people. The sampling frame for this survey is the Statistical Units Database. It includes information about NACE (The Nomenclature of Economic Activities) (19 levels), region (16 levels), sector (2 levels), size (3 levels) and the number of employees according to administrative data integrated by Statistics Poland (RE).

```
R> jvs <- read.csv("data-raw/jvs.csv",
+                  colClasses = c("character", "numeric",
+                                 rep("character", 3), "numeric"))
R> head(jvs)
```

```
                                        id private size nace region weight
1 a9cc990df6a99ab215a1bc13f51d4825c7d52d18       0    L    O     14      1
2 c9dbaf50890165ebe810aa770de0e9df903dc35b       0    L    O     24      6
3 718e0bba42bcec6ed98f9690db6d26cb7b93c880       0    L  R.S     14      1
4 532a1879a692b9d7bbb7282ba757d028156ef341       0    L  R.S     14      1
5 0b6b623fa45e257284a3049d097af322841337e3       0    L  R.S     22      1
6 c855a825e80866c00c7513721d5fcb38929f3cd6       0    M  R.S     26      1
```

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

```
      total      SE
sizeL  8561  890.34
sizeM 13758 1199.67
sizeS 29551 2123.03
```

The second source is the Central Job Offers Database (CBOP), which is a register of all vacancies submitted to Public Employment Offices and can be accessed via CBOP API. It contains job offers submitted by employers looking for new employees. If an employer is seeking new workers for their business, they can approach the County Employment Office (PUP) and submit the appropriate application. CBOP also contains information about unit identifiers (REGON and NIP), so we were able to link units to the sampling frame to obtain auxiliary variables with the same definitions as those used in the survey. Beyond that it contains `single_shift` outcome variable.

```
R> admin <- read.csv("data-raw/admin.csv",
+                   colClasses = c("character", "numeric",
+                                  rep("character", 3), "logical")
+                   )
R> head(admin)
```

```
                                        id private size nace region
1 7fddce081cbb1dd5da072e1683e9bfd20acab593       0    L    P     30
2 3d40ca689a4dca7d774981dc7db408301bf7f192       0    L    O     14
3 2b5a57b0c2f03c2b252e559bcd77d52789d33d9c       0    L    O     04
4 f2b18f5ef4386e70206d64810b5d3b7e0654918b       0    L    O     24
5 bf17263f8fa3a9ff12d7ff60c10c4e426aeeb36e       0    L    O     04
6 10fe847b7d19284e9e310105c5acfaf8f1ccbc37       1    L    C     28
  single_shift
1        FALSE
```

```
2         TRUE
3         TRUE
4         TRUE
5         TRUE
6        FALSE
```

4.1. IPW

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

R> summary(est1_logit)


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

-------------------------
Estimated population mean: 0.7083 with overall std.err of: 0.01063
And std.err for nonprobability and probability samples being respectively:
0.0063 and 0.008567

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


Based on: Inverse probability weighted method
For a population of estimate size: 52898.13
Obtained on a nonprobability sample of size: 9344
With an auxiliary probability sample of size: 6523
-------------------------

Regression coefficients:
-----------------------
For glm regression on selection variable:
```

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

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

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
```

4.2. IPW CALIB

```
R> est2_logit <- nonprob(
+   selection = ~ region + private + nace + size,
+   target = ~ single_shift,
+   svydesign = jvs_svy,
+   data = admin,
+   method_selection = "logit",
+   control_selection = controlSel(h = 1, est_method_sel = "gee")
+ )
```

4.3. IPW bootstrap

```
R> est3_logit <- nonprob(
+   selection = ~ region + private + nace + size,
+   target = ~ single_shift,
+   svydesign = jvs_svy,
+   data = admin,
+   method_selection = "logit",
+   control_inference = controlInf(var_method = "bootstrap", num_boot = 50),
+   verbose = T,
+ )

  |                                                                              |
```

4.5. DR

```
R> est8_dr1 <- nonprob(
+   selection = ~ region + private + nace + size,
+   outcome = single_shift ~ region + private + nace + size,
+   svydesign = jvs_svy,
+   data = admin,
+   method_selection = "logit",
+   method_outcome = "glm",
+   family_outcome = "binomial",
+   pop_size = sum(weights(jvs_svy))
+ )
```

5. Classes and S3methods

6. Summary and future work

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

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

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

Overall, **nonprobsvy** is a comprehensive toolkit for researchers and practitioners working with mixed data sources. It simplifies the implementation of complex statistical methods for non-probability samples, making it easier to produce robust, reliable estimates even when working with non-representative data. By providing a unified framework for inference with non-probability samples, **nonprobsvy** contributes to the growing field of data integration and enhances the ability of statisticians to make informed decisions from diverse data sources.

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

References

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

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

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

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

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

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

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

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

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

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

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

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

Groemping U (2023). *nleqslv: Solve Systems of Nonlinear Equations*. R package version 3.3.3, URL https://CRAN.R-project.org/package=nleqslv.

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

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

Luis Castro Martín RFG, del Mar Rueda M (2020). *NonProbEst: Estimation in Nonprobability Sampling*.

Lumley T (2004). "survey R package."

Marra G, Radice R (2023). *GJRM: Generalised Joint Regression Modelling*.

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

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

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

Tillé Y, Matei A (2021). *sampling: Survey Sampling*. R package version 2.9, URL https://CRAN.R-project.org/package=sampling.

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

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

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

Łukasz Chrostowski, Beręsewicz M (2024). *nonprobsvy: Package for Inference Based on Non-Probability Samples.* R package version 0.1.0, https://ncn-foreigners.github.io/nonprobsvy/, URL https://github.com/ncn-foreigners/nonprobsvy.

Affiliation:

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

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

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

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

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

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