Adam Mickiewicz University
Faculty of Mathematics and Computer Science
Department of Mathematical Statistics and Data Analysis

Łukasz Chrostowski
Course of study: Data analysis and processing
Album no: 456584

Statistical inference with non-probability samples

Wnioskowanie statystyczne na podstawie prób nieprobablistycznych

Master's thesis
written under the supervision of
UAM Professor Łukasz Smaga, PhD, DSc

Poznań 2024

ABSTRACT

The purpose of this work is to present a statistical method for performing statistical inference with non-probability survey samples (including big data) when additional information is available from external sources such as probability samples or vector of population totals or means. In addition to presenting existing solutions, a number of extensions, additional proofs and implementation of methods in the R language package are presented. In particular, the methods were extended to include other linking functions, new methods for selecting variables for the model were added and an additional approach to one of the methods (mass imputation) was proposed. It is worth mentioning that some of the methods considered in the work can be found in a working paper Chlebicki, Chrostowski, and Beręsewicz (2024) of which the author is a co-author. First, basic assumptions about data integration are presented. A number of estimation methods are then discussed, including mass imputation. In the next chapter, the results obtained are used to describe and apply estimators of the population mean. The asymptotic properties of the estimates are also examined. Finally, the implementation of the considered theory in a statistical package is presented. Its main functionalities are shown, as well as its application to real data. The work is based on collaboration within the framework of a grant funded by National Science Centre grant entitled: Towards census-like statistics for foreign-born populations – quality, data integration and estimation (no. 2020/39/B/HS4/00941).

Key words: Data integration, Doubly robust estimation, Propensity score estimation, Mass imputation

STRESZCZENIE

Niniejsza praca ma na celu przedstawienie metod statystycznych, których zadaniem jest przeprowadzanie wnioskowania statystycznego z nieprobabilistycznymi próbami badawczymi (w tym big data), gdy dostępne są dodatkowe informacje ze źródeł zewnętrznych, takich jak próby probablistyczne lub sumy czy średnie z populacji. Oprócz przedstawienia istniejących rozwiązań, zaprezentowano szereg rozszerzeń, dodatkowych dowodów i implementacji metod w pakiecie języka R. W szczególności rozszerzono metody o inne funkcje linkujące, dodano nowe metody doboru zmiennych do modelu oraz zaproponowano dodatkowe podejście do jednej z metod (masowa imputacja). Warto wspomnieć, że niektóre z metod rozważanych w pracy można znaleźć w artykule roboczym Chlebicki, Chrostowski, and Beręsewicz (2024), którego autor pracy jest współtwórcą. Najpierw przedstawione zostały podstawowe założenia dotyczące intergracji danych. Następnie omówiono szereg metod dotyczących estymacji, a także tak zwanej masowej imputacji. Kolejny rozdział przedstawia opis i zastosowanie estymatorów średniej z populacji korzystając z uzyskanych rezultatów. Badane są również asymptotyczne własności szacunków. Ostatecznie, przedstawiono implementację rozważanej teorii w pakiecie statystycznym. Pokazano jego najważniejsze funkcjonalności oraz sposób zastosowania do rzeczywistych

danych. Praca oparta jest na współpracy w ramach grantu finansowanego przez Narodowe Centrum Nauki zatytułowanego: Towards census-like statistics for foreign-born populations - quality, data integration and estimation (nr 2020/39/B/HS4/00941).

Słowa kluczowe: Integracja danych, Podwójnie odporna estymacja, Estymacja prawdopodobieństwa inkluzji, Masowa imputacja

Contents

Introduction and Notation

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; Beręsewicz, 2017; Citro, 2014; Elliott & Valliant, 2017). 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. 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.

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

The thesis concerns the non-probability sample inference. It consists of the following chapters:

Introduction to the problem of estimation from integrated data

Mass imputation

Inverse Probability Weighting

Doubly robust methods

Techniques of variable selection with high-dimensional data

R `nonprobsvy` package

Simulation study

Empirical study

In the first chapter, we will describe what the data integration problem is in the context of drawing inferences about the whole population. In particular, we will explain how to perform inference using the advantages of probability and non-probability samples. We will then discuss a number of estimation methods derived from data integration models such as mass imputation, propensity score estimation (inverse probability weighting) and doubly robust methods. In the fifth chapter, we will explain variable selection techniques for multivariate data, followed by a description of the R (R Core Team, 2023) language package in which all the methods described are implemented. Finally, we will present the results of the simulations carried out and an example of use on real data related to job vacancies.

To sum up, I can summarise my contribution to the literature on non-probability samples as follows:

- Extension of the inverse probability weighting estimation method with additional linking functions, i.e. probit and complementary log-log along with the derivation of the log-likelihood function, gradient, hessian, jacobian and variance.
- Addition of additional variable selection methods when estimating with probability and non-probability sample integration methods such as Least Absolute Shrinkage and Selection Operator and Minimax Concave Penalty.
- Extending the estimations and variable selection to the case where, instead of a probability sample, we have access to the vector of total or mean values of the variables from the population.

- Adding a new method of mass imputation for the data integration problem, i.e., predictive mean matching, along with the derivation of variance and implementation. Details have been described in the article Chlebicki, Chrostowski, and Beręsewicz (2024).
- Writing the `nonprobsvy` package (Chrostowski, Beręsewicz, & Chlebicki, 2023) package in R, which includes the implementation of all the methods described in the paper: mass imputation, inverse probability weighting, doubly robust, bias minimization, variable selection and many others (available on CRAN).

In writing this thesis I have used the literature listed in the bibliography. It is the primary source for further information on the topics covered here. The numbering of definitions, theorems and examples is separate for each chapter. The first digit is the chapter number and the second digit is the consecutive number of the theorem, example (1.1), etc. Formulas are numbered similarly. Proofs end with the symbol □.

Notation

S_A, S_B - Non-probability and probability sample

$\boldsymbol{\theta}$ - Parameter vector for inverse probability weighted estimator

$\boldsymbol{\beta}$ - Parameter vector for mass imputation estimator

μ_y - Population mean

$R_i^A = I(i \in S_A)$ - Indicator function corresponding to the unit membership of the non-probability sample.

$R_i^B = I(i \in S_B)$ - Indicator function corresponding to the unit membership of the probability sample.

π_i^{A} - The probability of an unit i belonging to the non-probability sample.

π_i^{B} - The probability of an unit i belonging to the probability sample.

π_{ij} - The probability of an unit i and j belonging to the same sample.

$d_i^A = \frac{1}{\pi_i^A}, d_i^B = \frac{1}{\pi_i^B}$ - Sample weights

N, n_A, n_B - Population size, non-probability sample size and probability sample size

$R \xrightarrow{d} Q$ - Convergence in distribution

lim - Limit of a function

$\ell, \ell^*, \mathrm{U}, \mathrm{H}, \mathrm{J}$ - Likelihood, log-likelihood, gradient, hessian and jacobian function respectively

Φ, ϕ - Distribution function and density function of the standard normal distribution $N(0,1)$

$\mathrm{W}, \mathrm{W}_{Ap}$ - Matrix notation of the hessian and matrix notation of the hessian for the non-probability part for the logit link function.

X^T - Transposition of matrix X.

Chapter 1

Introduction to estimation from integrated data

In this chapter, we will consider how we can use the available data with the characteristics described in the introduction to adequately model what we are interested in, in particular the mean and the sum of the characteristic from the population. At the outset, it is worth spending a few words on the mathematical assumptions and definition of the statistical model we will be working with. This is important so that the relevant properties of the estimators are satisfied. We will also describe the most common methods available in the scientific literature (Yang, Kim, and Song (2020), Yang and Kim (2020), Wu and Thompson (2020), Kim et al. (2021)), for estimating the probability of inclusion, including maximum likelihood or calibration methods using generalized estimating equations. A widely used technique for missing data problems is mass imputation, which is also discussed in this chapter. This type of procedure is particularly useful in public statistics and in surveys where missing responses from respondents are common.

1.1 Fundamental assumptions

Let $\mathrm{U} = \{1, 2, \ldots, N\}$ be a finite population of size N. The population consists of the set of units $\mathcal{F}_\mathrm{N} = \{(\boldsymbol{x}_i, y_i), i \in \mathrm{U}\}$, and the parameter whose estimation we are interested in is the population mean given by the formula $\mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i$. Consider a non-probability sample S_A containing n_A units. Let $\{(\boldsymbol{x}_i, y_i), i \in S_\mathrm{A}\}$ be the realisation of this sample and $R_i^A = I(i \in S_A)$ be the indicator function corresponding to the unit membership of the sample. 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 i R_j^A are independent with $i \neq j$.

Perhaps we should explain in a few words what the assumptions described mean and what they imply in practice. Firstly, from the first assumption we immediately have $\pi_i^{\mathrm{A}} = P\left(R_i^A = 1 \mid \boldsymbol{x}_i, y_i\right) = P\left(R_i^A = 1 \mid \boldsymbol{x}_i\right)$ for all $i \in S_A$. In the literature, this is called the MAR (*missing at random*) assumption, which means that the absence of an individual depends on the observed characteristics but not on the characteristic we are studying. Together with assumption A2, it is a condition for so-called strong ignorability.

As this is a non-probability sample, the inclusion probabilities π_i^A are generally not known. Hence we need to estimate them. However, it should be noted that the estimation cannot be based directly on a sample S_A. An additional set of information about our population is needed. This can be a probability reference sample, but also a vector of total feature values $\boldsymbol{x}$, obtained for example from public statistics. Let $\left\{(\boldsymbol{x}_i, d_i^B), i \in S_B\right\}$ be the realisation of a probability sample S_B, where $d_i^B = \left(\pi_i^B\right)^{-1}$ for $\pi_i^B = P\left(i \in S_B\right)$. In statistical jargon, the inverse of the probability of inclusion is called the survey weight. These are used to adjust the parameters of variables such as means, totals or standard deviations. An implementation of these types of probability sampling methods can be found in the `survey` package Lumley (2004), which is discussed in more detail in Chapter 6. Note also that the variable y is not part of the reference sample. The above description of the data is presented in a more concise form in Table 1.1.

In the next three chapters, we will discuss three approaches to estimating the mean of a given variable in the population under study. We will show the disadvantages, advantages and basic properties of the estimators. We will also present the derivation of the asymptotic variance of the means. The first estimator will be based on the mass imputation approach. We will then discuss the inverse probability weighting estimator and finally present the doubly robust estimator, which is an integration of the described methods.

Table 1.1: 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)	$n_A + 1$	✓	?	✓
	...	✓	?	✓
	$n_A + n_B$	✓	?	✓

Chapter 2

Mass imputation

Imputation refers to the process of replacing missing or incomplete data with substituted values. The goal of imputation is to allow for more complete data analysis, as many statistical methods require complete datasets. 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 entire datasets where many observations have missing values for the given variable. Kim et al. (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.

In surveys, it is sometimes the case that a large proportion of respondents do not answer certain questions, which can lead to non-response bias. A good example is IPSOS surveys, which are designed to provide preliminary results of elections, such as general elections. IPSOS typically use a representative sample of the electorate. Respondents are selected using various sampling techniques, such as stratified random sampling, to ensure that the sample reflects the overall demographics of the electorate.

However, respondents may not answer all questions, such as age, place of residence or voting behaviour in previous and current elections. In such cases, survey organisations use statistical methods to deal with missing data. One such method is imputation, where missing values are estimated based on other information available from the respondent or from the rest of the dataset. However, mass imputation (as mentioned in the original text) is not the common term; a more accurate term is multiple imputation, where several plausible values for the missing data are generated and combined to improve the accuracy of the estimate.

This approach helps to minimise the bias that can arise from non-response, ensuring that the survey results remain as accurate and reflective of the population as possible.

2.1 Theory - point estimator

Note that, by assumptions (Table 1.1), 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.

Table 2.1: Data layout after mass imputation

Sample	$\boldsymbol{X}$	Y	d
S_A	$\boldsymbol{X}_A$	$\boldsymbol{Y}$	?
S_B	$\boldsymbol{X}_B$	$\hat{\boldsymbol{Y}}$	$\boldsymbol{d}_B$

Based on this approach, we can obtain an estimate of the population mean based on known design weights and an imputation model for units from the probability sample:

$$\hat{\mu}_{MI} = \frac{1}{\hat{N}^{\mathrm{B}}} \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} \hat{y}, \tag{2.1}$$

$\hat{N}^{\mathrm{B}} = \sum_{i \in S_B} d_i^B$ 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 variable y, we use its estimated equivalents.

2.2 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

$$\mathbb{E}\left(y_i \mid \boldsymbol{x}_i\right) = m\left(\boldsymbol{x}_i, \boldsymbol{\beta}_0\right) \tag{2.2}$$

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 interesting in finding such $\boldsymbol{\beta}$ that is the solution of the following equation

$$U(\boldsymbol{\beta}) = \frac{1}{n_A} \sum_{i \in S_A} \left\{y_i - m\left(\boldsymbol{x}_i; \boldsymbol{\beta}\right)\right\} h\left(\boldsymbol{x}_i; \boldsymbol{\beta}\right) = \mathbf{0}, \tag{2.3}$$

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

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 $\hat{\nu}(i) = \underset{j \in S_A}{\arg\min}\, d(\boldsymbol{x}_i, \boldsymbol{x}_j)$.
2: If $k > 1$, then

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

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. Examples of such functions are described in Subsection 2.2. 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 $\hat{y}_i = m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right), \hat{y}_j = m\left(\boldsymbol{x}_j, \hat{\boldsymbol{\beta}}\right)$ dla $i \in S_B, j \in S_A$ 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, Chrostowski, and Beręsewicz (2024).

2.5 Variance estimation

When looking for answers about what characterises a population, we are not always interested only in the estimation of means or totals, but also in their variability. Hence the concept of the variance of an estimator and the resulting confidence interval. In this subsection, we will present both an analytical and a bootstrap approach for counting this statistic. We will always rely on certain assumptions that are mathematically relevant to the consistency of the estimation. Also, the derivation of the variance based on non-probability samples is not straightforward, so the results will be asymptotic (we assume that the non-probability sample size goes to infinity). In this chapter, of course, we will focus only on mass imputation.

2.5.1 Analytical approach

The analytical approach involves some mathematical formulae to obtain the variance estimator. It relies on assumptions about the distribution of the underlying sample and the estimator itself. It is worth noting that the starting point for the analytical variance for both mass imputation and the other methods is the Taylor linearisation, which is based on a well-known mathematical concept - the Taylor expansion, what is a way to represent a smooth function as an infinite sum of terms, each derived from the function's derivatives at a single point. It approximates a function using polynomials that are based on the values of the function's derivatives at a specific point.

Generalized Linear Models

Recall that the mass imputation estimator is formulated as follows

$$\hat{\mu} = \frac{1}{N} \sum_{i \in S_B} d_i^B \hat{y}_i,$$

where N is the population size, d_i are the survey weights, and $\hat{y}_i$ are the imputed values for the study variable Y. Before describing the variance of this estimator, let us introduce the following assumptions:

(B1) $\boldsymbol{\beta}^* = p\lim \hat{\boldsymbol{\beta}}$, where $p\lim$ denotes the probability limit (convergence in probability) and $\hat{\boldsymbol{\beta}}$ is the solution of (2.3).

(B2) $\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^* = O_p\left(n_B^{-1/2}\right)$ under $V\left\{\widehat{U}\left(\boldsymbol{\beta}^*\right)\right\} = O_p\left(n_B^{-1}\right)$.

In the paper Kim et al. (2021), the authors prove that the mass imputation estimator has the following property. The note, theorem and proof are transcribed in full from the above paper with a small change regarding notation.

Note 2.1. *Suppose that (x, y) has bounded fourth moments over the sequence of finite populations and that condition (B2) holds. Under the regularity conditions, the mass imputation estimator $\hat{\mu}$ satisfies*

$$\hat{\mu} = \tilde{\mu} + o_p(n_A^{-1/2}),$$

where n_A is the size of the non-probability sample and

$$\tilde{\mu} = N^{-1} \sum_{i \in S_B} d_i m\left(\boldsymbol{x}_i; \boldsymbol{\beta}^*\right) + n_A^{-1} \sum_{i \in S_A} \left\{y_i - m\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}^*\right)\right\} g\left(\boldsymbol{x}_i^\top \boldsymbol{\beta}^*\right),$$

where $g\left(\boldsymbol{x}_i; \boldsymbol{\beta}^*\right) = \boldsymbol{x}_i^\top \mathbf{c}$ and $\boldsymbol{c} = \left\{n_A^{-1} \sum_{i \in S_A} \frac{\partial m\left(\boldsymbol{x}_i^\top \beta\right)}{\partial \beta} \boldsymbol{x}_i^\top \boldsymbol{\beta}^*\right\}^{-1} N^{-1} \sum_{i \in S_B} d_i \frac{\partial m\left(\boldsymbol{x}_i^\top \beta\right)}{\partial \beta}$.

Theorem 2.1. *The asymptotic variance of $\hat{\mu}$ is given by*

$$V(\hat{\mu}) = V_A + V_B \tag{2.4}$$

where

$$V_B = V\left(\frac{1}{n_B} \sum_{i=1}^{N} m\left(\boldsymbol{x}_i', \boldsymbol{\beta}\right)\right) \tag{2.5}$$

and

$$V_A = V\left(\frac{1}{N} \sum_{i=1}^{N} \left(R_i^A h\left(\boldsymbol{x}_i^\top, \boldsymbol{\beta}^*\right) \boldsymbol{c} \pi_i^{-1} - 1\right) e_i\right) \cong E\left(\frac{1}{n_A^2} \sum_{i \in S_A} \left(h\left(\boldsymbol{x}_i^\top, \boldsymbol{\beta}^*\right) \boldsymbol{c}\right) e_i^2\right) \tag{2.6}$$

and $\boldsymbol{e} = y_i - m\left(\boldsymbol{x}^\top \boldsymbol{\beta}\right)$, $c = \left(\frac{1}{n_A} \sum_{i \in S_A} \dot{m}\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right) h\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right)'\right)^{-1} \frac{1}{N} \sum_{i \in S_B} d_i \dot{m}\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right)$ and h is an objective function for the given regression problem.

Proof. *Let*

$$Cov_N(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y}), \text{where } \bar{x} \text{ and } \bar{y} \text{ are the finite population means.}$$

Following Note 2.1, we can express

$$\tilde{\mu} - \mu = \frac{1}{N}\left(\sum_{i=1}^{N} d_i m_i^* - \sum_{i=1}^{N} m_i^*\right) + \left(\frac{1}{N} \sum_{i=1}^{N} R_i^A g_i^* e_i^* - \frac{1}{N} \sum_{i=1}^{N} e_i^*\right), \tag{A1}$$

where $m_i^* = m(x_i, \beta^*)$, $e_i^* = y_i - m(x_i, \beta^*)$, and $g_i = g(x_i, \beta^*)$. Further, ignoring the higher terms, the asymptotic bias can be given as

$$E(\tilde{\mu} - \mu) = E\left(\frac{1}{N} \sum_{i=1}^{N} R_i^A g_i^* e_i^* - \frac{1}{N} \sum_{i=1}^{N} e_i^*\right) = -E\left(\frac{1}{N} \sum_{i=1}^{N} e_i^*\right) = E\left(Cov_N\left(R_i^A \pi_A^{-1}, e_i^*\right)\right),$$

where $\pi_A^{-1} = \frac{n_A}{N}$. *Second equality comes from* $E\left[\sum_{i=1}^N R_i^A y_i - m(\boldsymbol{x}_i, \boldsymbol{\beta}^*) g_i^*\right] = 0$ *by definition of* β^*. *If* $\beta^* = \beta_0$, *then* $e_i^* = y_i - m(x_i, \beta^*) = e_i$, *and mass impuatation estimator (2.1) is unbiased. Otherwise, the bias is not zero. By equation (A1), we have*

$$V(\tilde{\mu} - \mu) = V_A + V_B, \text{ where}$$

$$V_B = V\left(\frac{1}{N}\sum_{i=1}^N d_i m_i^* - \frac{1}{N}\sum_{i=1}^N m_i^*\right)$$

is the variance component for sample S_B *and*

$$V_A = V\left[E\left\{N^{-1}\sum_{i=1}^N \left(R_i^A g_i \pi_A^{-1} - 1\right) e_i^* \mid \boldsymbol{x}, \boldsymbol{R}^A\right\}\right] + E\left[V\left\{N^{-1}\sum_{i=1}^N \left(R_i^A g_i \pi_A^{-1} - 1\right) e_i^* \mid \boldsymbol{x}, \boldsymbol{R}^A\right\}\right]$$

is the variance component for sample S_A.

If the sampling mechanism for sample S_A *is ignorable (*$P(R_i^A = 1|\boldsymbol{x}, y) = P(R = 1|\boldsymbol{x})$*), we have* $\beta^* = \beta_0$ *and* $e_i^* = e_i$. *In this case, the first term of equation (3) disappears because*

$$E[e_i|\boldsymbol{x}_i R_i^A] = E[y_i - m(\boldsymbol{x}_i, \beta_0)|\boldsymbol{x}_i, R_i^A] - E[y_i - m(\boldsymbol{x}_i, \beta_0)|X] = 0.$$

Thus can simplify V_A *as*

$$V_A = E\left[V\left(\frac{1}{N}\sum_{i=1}^N (R_i^A g_i \pi_A^{-1} - 1) e_i \middle| \boldsymbol{x}, R_i^A\right)\right] = E\left[\frac{1}{N^2}\left(\sum_{i=1}^N \left(R_i^A g_i \pi_A^{-1} - 1\right) e_i^2\right)\right].$$

The second equality comes from the independence of e_i *in the superpopulation model. If* $\frac{n_A}{N} = o(1)$, *than we can use*

$$V_A = E\left[\frac{1}{n_A^2}\sum_{i \in S_A} (e_i g_i)^2\right].$$

Finally, if the probability sample S_B *is selected by simple random sampling (SRS), the asymptotic variance of* $\hat{\mu}$ *is given by*

$$V(\hat{\mu} - \mu) = V_A + V_B,$$

where V_A *and* V_B *are specified as in Theorem 2.1.* □

Theorem 2.2. *Estimators of (2.5) and (2.6) are given by*

$$\hat{V}_B = \frac{n}{N^2}\sum_{i \in S_B}\sum_{j \in S_B} \frac{\pi_{ij}^B - \pi_i^B \pi_j^B}{\pi_{ij}^B} \frac{m\left(\boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}\right)}{\pi_i^A} \frac{m\left(\boldsymbol{x}_j^\top \hat{\boldsymbol{\beta}}\right)}{\pi_j^B} \tag{2.7}$$

and

$$\hat{V}_A = \frac{1}{n_A^2}\sum_{i \in S_A} \hat{e}_i^2 \left[h\left(\boldsymbol{x}_i^\top, \hat{\boldsymbol{\beta}}\right) \hat{\boldsymbol{c}}\right]. \tag{2.8}$$

Proof. Assuming $\hat{e}_i = y_i - m(\boldsymbol{x}_i, \hat{\beta})$ and $\hat{\boldsymbol{c}} = \left(\frac{1}{n_A}\sum_{i \in S_A} \dot{m}\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right) h\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right)'\right)^{-1} \frac{1}{N}\sum_{i \in S_B} d_i \dot{m}\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right)$, the plug-in estimators of V_A and V_B are given as in theorem (2.2). □

Nearest Neighbour Imputation

Let us introduce some notation before moving on to the assumptions. Let $\hat{\mu}_{g,\text{nni}}$ denote the MI estimator of NN for $k = 1$ and $\hat{\mu}_{g,\text{knn}}$ the estimator for $k > 1$. On the other hand, let $\hat{\mu}_{g,\text{HT}}$ denote the Horvitz-Thompson estimator (i.e. the MI estimator if y were known for the probability sample), where g denotes a known function, e.g. an identity function (we are then investigating properties for the standard estimator of the population mean). Let us also present the assumptions that will accompany us as we discuss the reasoning about the variance of the k-nearest neighbour estimator. These assumptions have been thoroughly presented and described in the article Yang, Kim, and Hwang (2021). To study the asymptotic properties of $\hat{\mu}_{g,\text{nni}}$, we impose the following regularity conditions.

(C1) (i) $f(\boldsymbol{X})$ and $\mu_g(\boldsymbol{X}) = \mathbb{E}[g(Y)|\boldsymbol{X}]$ are continuously differentiable for any continuous and bounded $g(Y)$.

(ii) $\mathbb{E}[g_\beta(Y)|\boldsymbol{X}]$ is bounded for $\beta = 1, 2$.

(C2) (i) There exist positive constants C_1 and C_2 such that $C_1 \leq n\pi_i \leq C_2$ for $i = 1, \ldots, N$.

(ii) The sampling fraction for Sample S_B is negligible, $(n_B/N) = o(1)$.

(iii) The sequence of the Horvitz–Thompson estimators $\hat{\mu}_{g,\text{HT}}$ satisfies $\text{var}_p(\hat{\mu}_{g,\text{HT}}) = O(n^{-1})$ and $n^{1/2}(\hat{\mu}_{g,\text{HT}} - \mu_g) \xrightarrow{d} N(0, \sigma_g^2)$ as $n \to \infty$, where var_p denotes the variance under the sampling design for Sample S_B.

Assumption C1 is a technical condition for functional continuity and finite moments that holds for common models. Assumption C2 applies to standard sampling designs in survey practice. It requires that the sample weights behave well in the sense that there are no extremely large weights that dominate other weights. According to the paper by Yang, Kim, and Hwang (2021), the following theorem is satisfied. Let us now state Slutsky theorem (without proof), which we will use in describing the variance of the estimator.

Theorem 2.3. *Let X_n be a sequence of random variables that converges in distribution to X, and let Y_n be a sequence of random variables that converges in probability to a constant c. Then, the following hold:*

1. $X_n + Y_n \xrightarrow{d} X + c$
2. $X_n Y_n \xrightarrow{d} X \cdot c$
3. *If $c \neq 0$, then* $\frac{X_n}{Y_n} \xrightarrow{d} \frac{X}{c}$

We will first present the variance of the nearest neighbour estimator and then generalise this approach to the case of $k > 1$.

Theorem 2.4. *Under Assumptions C1–C2 and $N/N_B = O(1)$, $\hat{\mu}_{g,nni}$ has the same distribution as $\hat{\mu}_{g,HT}$ as $n_A \to \infty$. Furthermore, under Assumption 4, $\hat{\mu}_{g,nni}$ is consistent for μ_g, and*

$$n^{1/2}(\hat{\mu}_{g,nni} - \mu_g) \xrightarrow{d} N(0, V_{nni})$$

where

$$V_{nni} = \lim_{n\to\infty} \frac{n}{N^2} E\left[\text{var}_p\left\{\sum_{i\in S_B} \pi_i^{-1} g\left(Y_i\right)\right\}\right].$$

which can be estimated as

$$\hat{V}_{nni} = \frac{n}{N^2} \sum_{i\in S_B} \sum_{j\in A} \frac{\pi_{ij} - \pi_i\pi_j}{\pi_i\pi_j} \frac{g\left(Y_{i1}\right)}{\pi_i} \frac{g\left(Y_{j1}\right)}{\pi_j}$$

where $\hat{V}_{nni}$ is consistent etimator of V_{nni} and π_{ij} are the joint inclusion probability for units i and j.

Similarly, for the k-nearest neighbours estimator with $k > 1$, Yang, Kim, and Hwang (2021) proved the following.

Theorem 2.5. *Under Assumptions C1–C2, $\hat{\mu}_{g,knn}$ is consistent for μ_g, and*

$$\sqrt{n}(\hat{\mu}_{g,knn} - \mu_g) \xrightarrow{d} N(0, V_{knn}),$$

where

$$V_{knn} = \lim_{n\to\infty} \frac{n}{N^2} \left(E\left[\text{var}_p\left\{\sum_{i\in S_B} \pi_i^{-1} \mu_g\left(\mathbf{X}_i\right)\right\}\right] + E\left\{\frac{1-\pi_i^A(\mathbf{X})}{\pi_i^A(\mathbf{X})}\sigma_g^2(\mathbf{X})\right\}\right),$$

and $\sigma_g^2(\boldsymbol{X}) = \mathbb{E}[(g(Y) - \mu_g(\boldsymbol{X}))^2|\boldsymbol{X}]$.

Note that if non–probability sample us a large fraction of the target population (π_i^A goes to 1), V_{knn} can be smaller than V_{nni} suggesting $\hat{\mu}_{g,\text{knn}}$ more stable estimator than $\hat{\mu}_{g,\text{nni}}$. As mentioned, the proofs of the theorems presented above can be found in the supplementary material of Yang, Kim, and Hwang (2021).

Predictive Mean Matching

In the working paper Chlebicki, Chrostowski, and Beręsewicz, 2024, the authors show how to estimate the variance of predictive mean matching estimator in respect to known or estimated value of population size. We will now present the theorems described there, as the proofs are quite computationally complex those interested in their details are referred to the mentioned article.

Theorem 2.6. *If the population size N is known then the variance of pmm estimators is given by*

$$\mathbb{V}\left(\hat{\mu}\right) = V_1 + V_2, \tag{2.9}$$

where

$$V_1 = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\pi_{ij}-\pi_i\pi_j}{\pi_i\pi_j\pi_{ij}}\mathbb{E}\left[\frac{1}{k}\sum_{t=1}^{k}y_{\hat{\nu}(i,t)}\cdot\frac{1}{k}\sum_{t'=1}^{k}y_{\hat{\nu}(j,t')}\right], \tag{2.10}$$

and

$$V_2 = \mathbb{V}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{k}\sum_{t=1}^{k}y_{\hat{\nu}(i,t)}\right]. \tag{2.11}$$

Theorem 2.7. *Estimators of* (2.10) *and* (2.11) *are given by*

$$\hat{V}_1 = \frac{1}{N^2}\sum_{i\in S_B}\sum_{j\in S_B}\frac{\pi_{ij}^B-\pi_i^B\pi_j^B}{\pi_{ij}^B}\frac{\frac{1}{k}\sum_{t=1}^{k}y_{\hat{\nu}(i,t)}}{\pi_i^B}\frac{\frac{1}{k}\sum_{t'=1}^{k}y_{\hat{\nu}(j,t')}}{\pi_j^B}, \tag{2.12}$$

and

$$\hat{V}_2 = \frac{1}{N^2}\sum_{i\in S_B}\sum_{j\in S_B}\pi_{ij}^{-1}\widehat{cov}\left(\frac{1}{k}\sum_{t=1}^{k}y_{\hat{\nu}(i,t)},\frac{1}{k}\sum_{t'=1}^{k}y_{\hat{\nu}(j,t')}\right), \tag{2.13}$$

respectively.

According to Chlebicki, Chrostowski, and Beręsewicz, 2024 The V_1 term is just a variance of the Horvits–Thompson estimator for the mean of imputed values and the V_2 term can be seen as a compensation for inherent randomness (induced into S_A by unknown sampling, regression estimation etc.) resulting from PMM imputation. Authors show that V_2 can be omitted for large S_A samples and point that it cannot be estimated directly from samples S_B and S_A proposing "mini-bootstrap" approach to estimate $\text{cov}(\frac{1}{k}\sum_{t=1}^{k}y_{\hat{\nu}(i,t)},\frac{1}{k}\sum_{t'=1}^{k}y_{\hat{\nu}(j,t')}$ explained in algorithm 5.

Algorithm 5 Non-parametric mini-bootstrap estimator for covariance terms

1: Sample n_A units from S_A with replacement to create S'_A (if pseudo-weights are present inclusion probabilities should be proportional to their inverses).
2: Estimate regression model $\mathbb{E}[Y|\boldsymbol{X}] = m(\boldsymbol{X},\boldsymbol{\beta})$ based on $j \in S'_A$ from step 1
3: Compute $\hat{\nu}'(i,t)$ for $t = 1,\ldots,k, i \in S_B$ using estimated $m(\boldsymbol{x}',\cdot)$ and $\{(y_j,\boldsymbol{x}_j)|j \in S'_A\}$
4: Compute $\frac{1}{k}\sum_{t=1}^{k}y_{\hat{\nu}'(i)}$ using Y values from S'_A.
5: Repeat steps 1-4 M times (we set $M = 50$ in our simulations)
6: Estimate covariance between imputed values for each pair $i,j \in S_B$ using constructed pseudo-sample with values from 4.

2.5.2 Bootstrap approach

As the computing power of our computers increases, bootstrap statistical methods are becoming increasingly popular for estimating sample parameters of interest. Bootstrap is a powerful method for estimating the sample mean or estimator variance using resampling methods. It involves repeatedly sampling the observations from the sample with replacement and calculating

statistics for each bootstrap sample. This method allows the variance or mean to be calculated without making strong assumptions about the sample distribution. In the context of estimating variance, the non–parametric bootstrap approach provides a straightforward way to measure the variability of the estimator across different bootstrap samples from the same population. Of course, it is worth noting that the process of computing bootstrap variance is usually much more time-consuming than the analytical approach.

In this section we describe the bootstrap procedures for estimating the variance of mass imputation estimators. As we will see, the general procedure is the same regardless of the mass imputation method chosen. Only the technical nuances relating to the estimation of the population mean itself in each iteration differ.

Algorithm 6 Bootstrap variance estimator – GLM

1: Sample n_A units from S_A with replacement to create S'_A (if pseudo-weights are present inclusion probabilities should be proportional to their inverses).
2: Sample n_B units from S_B according to the design weights for probability sample to create S'_B.
3: Estimate regression model $\mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}'] = m(\boldsymbol{x}', \boldsymbol{\beta})$ based on $j \in S'_A$ from step 1.
4: Compute bootstrap mean as
$$\hat{\mu} = \frac{1}{\hat{N}^{\mathrm{B}}} \sum_{i \in S'_{\mathrm{B}}} d_i^{\mathrm{B}} m\left(\boldsymbol{x}'_i, \hat{\boldsymbol{\beta}}\right).$$
5: Repeat steps 1-4 L times (we set $L = 500$ in our simulations).
6: Estimate variance term as $\hat{V} = \frac{1}{L-1} \sum_{i=1}^{L} (\hat{\mu}_i - \bar{\mu})^2$ where $\bar{\mu} = \frac{1}{L} \sum_{i=1}^{L} \hat{\mu}_i$.

Algorithm 7 Bootstrap variance estimator – NN

1: Sample n_A units from S_A with replacement to create S'_A (if pseudo-weights are present inclusion probabilities should be proportional to their inverses).
2: Sample n_B units from S_B according to the design weights for probability sample to create S'_B.
3: Obtain $\hat{y}_i$ for each $i \in S'_B$ basing in Algorithm 4 and compute estimation mean as
$$\hat{\mu} = \frac{1}{\hat{N}^{\mathrm{B}}} \sum_{i \in \mathrm{S}'_{\mathrm{B}}} d_i^{\mathrm{B}} \hat{y}_i.$$
4: Repeat steps 1-4 L times (we set $L = 500$ in our simulations).
5: Estimate variance term as $\hat{V} = \frac{1}{L-1} \sum_{i=1}^{L} (\hat{\mu}_i - \bar{\mu})^2$ where $\bar{\mu} = \frac{1}{L} \sum_{i=1}^{L} \hat{\mu}_i$.

Algorithm 8 Bootstrap variance estimator – PMM

1: Sample n_A units from S_A with replacement to create S'_A (if pseudo-weights are present inclusion probabilities should be proportional to their inverses).
2: Sample n_B units from S_B according to the design weights for probability sample to create S'_B.
3: Estimate regression model $\mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}'] = m(\boldsymbol{x}', \boldsymbol{\beta})$ based on $j \in S'_A$ from step 1
4: Follow steps 2-4 from Algorithm 6 or 7 depending on the imputation method ($\hat{y} - \hat{y}$ or $\hat{y} - y$) to compute $\hat{\mu}$ in current iteration.
5: Repeat steps 1-4 L times (we set $L = 500$ in our simulations).
6: Estimate V_2 term using algorithm 5
7: Estimate variance V_1 term as $\hat{V}_1 = \frac{1}{L-1}\sum_{i=1}^{L}(\hat{\mu}_i - \bar{\mu})^2$ where $\bar{\mu} = \frac{1}{L}\sum_{i=1}^{L}\hat{\mu}_i$ and final bootstrap variance as $\hat{V} = \hat{V}_1 + \hat{V}_2$

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

2.6 Comparison of the proposed methods

In this section we will compare three different approaches for estimating the population mean based on mass imputation. In particular, we will focus on the mathematical properties and the stability or precision of these methods. Although they are all used to estimate the same information about the population under study, the mathematics or even the methodology is quite different. In a nutshell, the first method (GLM) is the so-called parametric method, because we simply look for linear (or non-linear) relationships between the explanatory and predictor variables and find the parameters that maximise the log-likelihood function or minimize the objective function. The nearest neighbour method, on the other hand, simply searches two samples (probability and non-probability) for mutual 'neighbours' and imputes the corresponding values of the explanatory variable (or their mean if $k > 1$). There is no wider mathematical theory here, although we are concerned with the analysis of the results themselves, which can be more precise for non-linear relationships. The predictive mean matching method is a combination of the other two in that it uses both methods to find the MI estimator. It is also the most recent approach in the field, and additionally allows some leeway in the choice of the type of predictive matching, as described in the previous section. The following conclusions and observations follow from the simulations carried out in Chapter 7 and the papers as Kim et al. (2021), Yang, Kim, and Hwang (2021) and Chlebicki, Chrostowski, and Beręsewicz (2024).

The k-NN estimator is consistent under certain conditions and converges to the true popu-

lation parameter as the sample size increases. However, its efficiency depends on the choice of the k parameter, which can significantly affect the level of bias and variance in the estimator. The PMM estimator is also consistent even in cases where the model is partially misspecified, making it more robust compared to k-NN. The GLM estimator, used in the context of mass imputation, ensures consistency provided that the model is correctly specified and can be transferred to the probability sample. The GLM is also characterised by a well-defined asymptotic variance, which makes it stable in large samples, although it is sensitive to model misspecification, which can introduce bias.

The stability of the k-NN estimator is highly dependent on the choice of k and the sample size. Larger k values generally increase stability, but may also introduce greater bias. The k-NN estimator is also susceptible to the curse of dimensionality, which reduces its stability as the number of variables increases. The PMM, on the other hand, is highly stable due to its robustness to partial model misspecification and its immunity to the curse of dimensionality, making it more adaptable in practice. The stability of the GLM estimator depends on the correct specification of the model and the size of the non-probability sample used for training. Larger non-probability samples improve stability, but poor covariate selection can reduce the stability of this estimator.

The k-NN estimator performs well with small values of k, but its precision may decrease with a large number of variables or an inappropriate choice of k. PMM provides high precision, especially for large datasets, and often outperforms k-NN in terms of MSE (mean squared error) due to its better handling of model misspecification and dimensionality issues. GLM generally provides accurate estimates, especially when the model is correctly specified and covariate selection is optimised. Its precision is enhanced by effective variance estimation techniques, making it more efficient than k-NN, especially in larger samples with more complex structures.

In summary, PMM stands out as the most robust and generally applicable estimator, especially when the correctness of the model is uncertain. GLM offers high efficiency and precision in well-specified models, while k-NN is useful for its simplicity but less robust in more complex scenarios.

Chapter 3

Inverse probability weighting

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

3.1 Theory – point estimator

As mentioned above, there are a number of methods in the scientific literature on data integration that aim to estimate the propensity score. In general, these are based on parametric modelling of the inclusion probabilities. So let us say that $P\left(R_i^A = 1 \mid \boldsymbol{x}_i\right) = \pi(\boldsymbol{x}_i, \boldsymbol{\theta}_0)$ where $\boldsymbol{\theta}_0$ is the true value of the parameter vector we want to estimate. There are three binding link functions that can be used to regress a dichotomous dependent variable: logistic, complementary log-log and probit functions.

The procedures associated with these functions are implemented in the `stats` package, which contains a wide range of procedures associated with generalized linear models. The same applies to the `nonprobsvy` package (Chapter 6).

In general, we are interested in obtaining, after the estimation, the layout of the data shown in Table 3.1.

The estimated propensity score is used to construct an inverse probability weighting esti-

Table 3.1: Data layout after estimation of inclusion probabilities

Sample	$\boldsymbol{X}$	Y	d	R_i^A
S_A	$\boldsymbol{X}_A$	$\boldsymbol{Y}_A$	$\hat{\boldsymbol{d}}_A$	**1**
S_B	$\boldsymbol{X}_B$	?	$\boldsymbol{d}_B$	**0**

mator of the population mean of the form

$$\hat{\mu}_{IPW} = \frac{1}{\hat{N}^A} \sum_{i \in S_A} \frac{y_i}{\hat{\pi}_i^A}. \tag{3.1}$$

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

3.2 Estimation methods

A number of approaches to probability estimation are distinguished in the literature. We will focus on the two most commonly considered. First, there is the maximum likelihood estimation method that was presented in Chen, Li, and Wu (2020) and whose main objective is to find the parameters that maximise the likelihood of the observed data under the model. Secondly, we will present the approach from Wu (2022) of estimating equation method, which is based on the calibration, i.e. we estimate the parameters so that the sums of the dependent variables in the non-probability sample reproduce the sums of the variables in the probability sample or their corresponding vector of sums from some external sources.

3.2.1 Maximum likelihood method

Consider the following likelihood function

$$\begin{aligned} \ell(\boldsymbol{\theta}) &= \sum_{i=1}^{N} \left\{ R_i^A \log \pi_i^{\mathrm{A}} + \left(1 - R_i^A\right) \log\left(1 - \pi_i^{\mathrm{A}}\right) \right\} \\ &= \sum_{i \in S_{\mathrm{A}}} \log \left\{ \frac{\pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)}{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)} \right\} + \sum_{i=1}^{N} \log \left\{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\right\} \end{aligned} \tag{3.2}$$

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

$$\ell^*(\boldsymbol{\theta}) = \sum_{i \in S_{\mathrm{A}}} \log \left\{ \frac{\pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)}{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)} \right\} + \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} \log \left\{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\right\}. \tag{3.3}$$

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.

3.2.2 Generalized Estimating Equations

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

$$\mathbf{G}(\boldsymbol{\theta}) = \sum_{i \in S_A} h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) - \sum_{i \in S_B} d_i^B \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \mathbf{0}, \tag{3.4}$$

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

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

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

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

The advantage of this method is the ability to estimate with global values of the variables (e.g. from external sources) instead of a probability sample. Note that this is allowed by the second form of the $\mathbf{G}$ function. Its second term is nothing more than the estimated sums of the $\boldsymbol{x}$ variables. On the other hand, empirical studies suggest that the process of solving this type of equation may be less stable than the maximum likelihood method. In other words, the iterative algorithm for finding zeros that satisfy equation (3.4) may not converge.

3.3 Logistic regression

Logistic regression has become one of the primary methods for binary classification, i.e. assigning an observation $\boldsymbol{x}$ to one of the two classes considered in a given problem. This is based on modelling the probability of belonging to a given class and then assigning it to the class for which it is (usually) higher. This method is based on the so-called sigmoidal function of the form

$$\pi_i^A = \pi(\boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{\exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}_0\right)}{1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}_0\right)} \tag{3.5}$$

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

3.3.1 MLE

By applying the logistic regression model on the variable R_i^A we obtain the following.

$$\ell^*(\boldsymbol{\theta}) = \sum_{i \in S_{\mathrm{A}}} \boldsymbol{x}_i^\top \boldsymbol{\theta} - \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} \log\left\{1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)\right\}. \tag{3.6}$$

Using the standard algorithm for finding the extremum of a function, first determine the gradient of the function $l^*(\boldsymbol{\theta})$.

Note 3.1. *Let* $\mathrm{U}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} l^*(\boldsymbol{\theta})$. *We have then*

$$\mathrm{U}(\boldsymbol{\theta}) = \sum_{i \in S_A} \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \pi(\boldsymbol{x}_i, \boldsymbol{\theta}) \boldsymbol{x}_i \tag{3.7}$$

Proof.

$$\begin{aligned}
\mathrm{U}(\boldsymbol{\theta}) &= \sum_{i \in S_A} \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \left\{ \frac{1}{1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)} \right\} \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) \boldsymbol{x}_i \\
&= \sum_{i \in S_A} \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \left\{ \frac{\exp\left(\boldsymbol{x}_i^T \boldsymbol{\theta}\right)}{1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)} \right\} \boldsymbol{x}_i \\
&= \sum_{i \in S_A} \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \pi(\boldsymbol{x}_i, \boldsymbol{\theta}) \boldsymbol{x}_i
\end{aligned}$$

□

The maximum likelihood estimator is obtained by solving the equation $\mathrm{U}(\boldsymbol{\theta}) = \mathbf{0}$. Finding the explicit form of a vector $\boldsymbol{\theta}_0$ solving the above equation is not possible. It is therefore necessary to use iterative algorithms for finding zeroes of functions. For example, the Newton-Raphson method can be used. For this purpose, let us determine the Hessian of the maximum-likelihood function.

Note 3.2. *Let* $\mathrm{H}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \mathrm{U}(\boldsymbol{\theta})$. *We then have*

$$\mathrm{H}(\boldsymbol{\theta}) = -\sum_{i \in S_B} d_i^B \pi(\boldsymbol{x}_i, \boldsymbol{\theta})(1 - \pi(\boldsymbol{x}_i, \boldsymbol{\theta})) \boldsymbol{x}_i \boldsymbol{x}_i^\top. \tag{3.8}$$

Proof.

$$\begin{aligned}
\frac{\partial \mathrm{U}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= -\sum_{i \in S_B} d_i^B \left\{ \frac{\exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) \boldsymbol{x}_i (1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)) - \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) \boldsymbol{x}_i}{(1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right))^2} \right\} \boldsymbol{x}_i \\
&= -\sum_{i \in S_B} d_i^B \left\{ \frac{\exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) \boldsymbol{x}_i + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) \boldsymbol{x}_i \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) - \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) \boldsymbol{x}_i}{(1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right))^2} \right\} \boldsymbol{x}_i \\
&= -\sum_{i \in S_B} d_i^B \left\{ \frac{\exp\left(\boldsymbol{x}_i^T \boldsymbol{\theta}\right)}{1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)} \frac{(1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right) - \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)) \boldsymbol{x}_i}{1 + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}\right)} \right\} \boldsymbol{x}_i \\
&= -\sum_{i \in S_B} d_i^B \pi(\boldsymbol{x}_i, \boldsymbol{\theta})(1 - \pi(\boldsymbol{x}_i, \boldsymbol{\theta})) \boldsymbol{x}_i \boldsymbol{x}_i^\top = \boldsymbol{X}_B^\top \boldsymbol{W}_{lB} \boldsymbol{X}_B,
\end{aligned}$$

where

$$\boldsymbol{W}_{lB} = diag\left(-d_1^B\pi(\boldsymbol{x}_1,\boldsymbol{\theta})(1-\pi(\boldsymbol{x}_1,\boldsymbol{\theta})), -d_2^B\pi(\boldsymbol{x}_2,\boldsymbol{\theta})(1-\pi(\boldsymbol{x}_2,\boldsymbol{\theta})),\right.$$
$$\left.\ldots, -d_{n_B}^B\pi(\boldsymbol{x}_{n_B},\boldsymbol{\theta})(1-\pi(x_{n_B},\boldsymbol{\theta}))\right).$$

□

Finally, the iterative formula for $\boldsymbol{\theta}$ looks as follows

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} - \left\{H\left(\boldsymbol{\theta}^{(m)}\right)\right\}^{-1} U\left(\boldsymbol{\theta}^{(m)}\right),$$

where $H(\boldsymbol{\theta}) = \frac{\partial}{\partial\boldsymbol{\theta}}\mathrm{U}(\boldsymbol{\theta})$ and the start vector is given by $\boldsymbol{\theta}^0 = \mathbf{0}$.

3.3.2 Generalized Estimating Equations

The equation (3.4) from Section 3.2.2 can be solved using the Newton-Raphson method (or other similar methods). However, we will need the derivative of the function G (the Jacobian). In this section, we will derive its form for different types of the function h.

Note 3.3. *Let* $\mathrm{J}(\boldsymbol{\theta}) = \frac{\partial\mathbf{G}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$. *Then for* $h(\boldsymbol{x}_i,\boldsymbol{\theta}) = \boldsymbol{x}_i\pi(\boldsymbol{x}_i,\boldsymbol{\theta})^{-1}$, *we have*

$$\mathrm{J}(\boldsymbol{\theta}) = -\sum_{i\in S_A}\frac{1-\pi_i^A\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}\right)}{\pi_i^A\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}\right)}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}. \tag{3.9}$$

and for $h(\boldsymbol{x}_i,\boldsymbol{\theta}) = \boldsymbol{x}_i$ *we have*

$$\mathrm{J}(\boldsymbol{\theta}) = -\sum_{i\in S_B}\frac{1}{\pi_i^B}\pi_i^A\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}\right)\left(1-\pi_i^A\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}\right)\right)\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}} \tag{3.10}$$

Proof. Case 1: $h(\boldsymbol{x}_i,\boldsymbol{\theta}) = \boldsymbol{x}_i\pi(\boldsymbol{x}_i,\boldsymbol{\theta})^{-1}$

For this case, $\mathbf{G}(\boldsymbol{\theta})$ can be rewritten as:

$$\mathbf{G}(\boldsymbol{\theta}) = \sum_{i\in S_A}\boldsymbol{x}_i\pi(\boldsymbol{x}_i,\boldsymbol{\theta})^{-1} - \sum_{i\in S_B}d_i^B\boldsymbol{x}_i$$

To find the Jacobian $\mathrm{J}(\boldsymbol{\theta})$, we differentiate $\mathbf{G}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$:

$$\mathrm{J}(\boldsymbol{\theta}) = \frac{\partial\mathbf{G}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$$

Differentiating the first sum:

$$\frac{\partial}{\partial\boldsymbol{\theta}}\left(\sum_{i\in S_A}\boldsymbol{x}_i\pi(\boldsymbol{x}_i,\boldsymbol{\theta})^{-1}\right) = -\sum_{i\in S_A}\frac{\partial\pi(\boldsymbol{x}_i,\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\cdot\frac{\boldsymbol{x}_i}{\pi(\boldsymbol{x}_i,\boldsymbol{\theta})^2}$$

Using the logit link function for the propensity score from 3.5 we have:

$$\frac{\partial \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)}{\partial \boldsymbol{\theta}} = \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\left(1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\right)\boldsymbol{x}_i$$

Substituting this into the expression:

$$\mathrm{J}(\boldsymbol{\theta}) = -\sum_{i \in S_A} \frac{\pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\left(1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\right)\boldsymbol{x}_i \boldsymbol{x}_i^\top}{\pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^2} = -\sum_{i \in S_A} \frac{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)}{\pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)} \boldsymbol{x}_i \boldsymbol{x}_i^\top$$

This completes the proof for the first case.

Case 2: $h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \boldsymbol{x}_i$

For this case, $\mathbf{G}(\boldsymbol{\theta})$ can be rewritten as:

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

To find the Jacobian $\mathrm{J}(\boldsymbol{\theta})$, differentiate $\mathbf{G}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$:

$$\mathrm{J}(\boldsymbol{\theta}) = \frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

Differentiating the second sum:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \left(\sum_{i \in S_B} d_i^B \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) \boldsymbol{x}_i \right) = \sum_{i \in S_B} d_i^B \frac{\partial \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)}{\partial \boldsymbol{\theta}} \boldsymbol{x}_i$$

Substituting the expression for $\frac{\partial \pi(\boldsymbol{x}_i, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$:

$$\mathrm{J}(\boldsymbol{\theta}) = \sum_{i \in S_B} d_i^B \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\left(1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\right)\boldsymbol{x}_i \boldsymbol{x}_i^\top$$

Given that $d_i^B = \frac{1}{\pi_i^B}$, we have:

$$\mathrm{J}(\boldsymbol{\theta}) = -\sum_{i \in S_B} \frac{1}{\pi_i^B} \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\left(1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\right)\boldsymbol{x}_i \boldsymbol{x}_i^\top$$

This completes the proof for the second case.

□

3.4 Probit regression

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

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

3.4.1 MLE

Asumming probit regression for modelling of the inclusion probability to non-probability sample we derive MLE function as

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

For clarity of calculation in the case of the probit model, we will use the so-called chain rule for counting derivatives described in the following note.

Note 3.4. *Notice that*

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}} = \frac{\partial \ell}{\partial p} \frac{\partial p}{\partial \eta} \frac{\partial \eta}{\partial \boldsymbol{\theta}} = \frac{\partial \ell}{\partial p} \frac{\partial p}{\partial \eta} \boldsymbol{X}$$

where $\eta = \boldsymbol{X}^\top \boldsymbol{\theta}$ *and* $\frac{\partial \ell}{\partial \boldsymbol{\theta}} = \boldsymbol{U}\boldsymbol{X}$ *for* $\boldsymbol{U} = \frac{\partial \ell}{\partial p} \frac{\partial p}{\partial \eta} = \frac{\partial \ell}{\partial \eta}$. *Similarly we have*

$$\frac{\partial^2 \ell}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} = \boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X},$$

where $\boldsymbol{W} = \frac{\partial^2 \ell}{\partial \eta}$.

Since we operate on the probability function of the normal distribution, let us present the use of this distribution, which will be useful in the detailed discussion steps on maximum likelihood estimation.

Note 3.5. *Notice that*

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

Note 3.6. *Let* $\mathrm{U}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} l^*(\boldsymbol{\theta})$ *and* $\mathbf{U} = \frac{\partial \ell}{\partial \boldsymbol{\eta}}$. *Then we have*

$$\mathbf{U} = \sum_{i \in S_A} \frac{\phi(\eta)}{\Phi(\eta)(1 - \Phi(\eta))} - \sum_{i \in S_B} d_i^B \frac{\phi(\eta)}{1 - \Phi(\eta)},$$

where $\eta = \boldsymbol{x}^T \boldsymbol{\theta}$ *and*

$$\mathrm{U}(\boldsymbol{\theta}) = \sum_{i \in S_A} \frac{\phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})}{\Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})(1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta}))} \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \frac{\phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})}{1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})} \boldsymbol{x}_i.$$

Proof. The gradient of the log-likelihood function $l^*(\boldsymbol{\theta})$ is obtained by differentiating the two sums directly.

For the first sum:

$$\sum_{i \in S_A} \log \left\{ \frac{\Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})}{1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})} \right\} \quad \text{we have} \quad \sum_{i \in S_A} \frac{\phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})}{\Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})(1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta}))} \boldsymbol{x}_i$$

For the second sum:

$$\sum_{i\in S_B} d_i^B \log\{1-\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})\} \quad \text{we have} \quad -\sum_{i\in S_B} d_i^B \frac{\phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})}{1-\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})}\boldsymbol{x}_i$$

Thus, the gradient is:

$$\frac{\partial l^*(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = \sum_{i\in S_A} \frac{\phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})}{\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})(1-\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta}))}\boldsymbol{x}_i - \sum_{i\in S_B} d_i^B \frac{\phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})}{1-\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})}\boldsymbol{x}_i$$

□

Note 3.7. *Let* $\mathrm{H}(\boldsymbol{\theta}) = \frac{\partial}{\partial\boldsymbol{\theta}}\mathrm{U}(\boldsymbol{\theta})$ *and* $\boldsymbol{W} = \frac{\partial^2\ell}{\partial\eta}$. *Then we have*

$$\mathbf{W} = \sum_{i\in S_A} \frac{\phi\left(\eta\right)^2\left(2\Phi\left(\eta\right)-1\right)}{\left(\Phi\left(\eta\right)-1\right)^2\Phi\left(\eta\right)^2} - \sum_{i\in S_B} d_i^B \frac{\phi(\eta)^2}{\left(\Phi(\eta)-1\right)^2}$$

and finally

$$\begin{aligned}
\mathrm{H}_p(\boldsymbol{\theta}) &= \sum_{i\in S_A} \frac{-\boldsymbol{x}_i^\top\phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})}{\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})(1-\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta}))}\boldsymbol{x}_i\boldsymbol{x}_i^\top - \sum_{i\in S_A} \frac{\phi(\boldsymbol{x}_i^\top\boldsymbol{\theta}))^2\left(1-2\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta}))\right)}{\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})^2(1-\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta}))^2}\boldsymbol{x}_i\boldsymbol{x}_i^\top \\
&- \sum_{i\in S_B} \frac{\boldsymbol{x}_i^\top\boldsymbol{\theta}\phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})}{(1-\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta}))}\boldsymbol{x}_i\boldsymbol{x}_i^\top - \sum_{i\in S_B} \frac{\phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})^2}{\left(1-\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})\right)} \\
&= \boldsymbol{X}_A^\top\boldsymbol{W}_{Ap}\boldsymbol{X}_A - \boldsymbol{X}_B^\top\boldsymbol{W}_{Bp}\boldsymbol{X}_B,
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{W}_{Ap} = diag\Bigg(&\frac{-\boldsymbol{x}_1^\top\phi(\boldsymbol{x}_1^\top\boldsymbol{\theta})}{\Phi(\boldsymbol{x}_1^\top\boldsymbol{\theta})(1-\Phi(\boldsymbol{x}_1^\top\boldsymbol{\theta}))}\boldsymbol{x}_1\boldsymbol{x}_1^\top - \frac{\phi(\boldsymbol{x}_1^\top\boldsymbol{\theta}))^2\left(1-2\Phi(\boldsymbol{x}_1^\top\boldsymbol{\theta}))\right)}{\Phi(\boldsymbol{x}_1^\top\boldsymbol{\theta})^2(1-\Phi(\boldsymbol{x}_1^\top\boldsymbol{\theta}))^2}\boldsymbol{x}_1\boldsymbol{x}_1^\top, \\
&\frac{-\boldsymbol{x}_2^\top\phi(\boldsymbol{x}_2^\top\boldsymbol{\theta})}{\Phi(\boldsymbol{x}_2^\top\boldsymbol{\theta})(1-\Phi(\boldsymbol{x}_2^\top\boldsymbol{\theta}))}\boldsymbol{x}_2\boldsymbol{x}_2^\top - \frac{\phi(\boldsymbol{x}_2^\top\boldsymbol{\theta}))^2\left(1-2\Phi(\boldsymbol{x}_2^\top\boldsymbol{\theta}))\right)}{\Phi(\boldsymbol{x}_2^\top\boldsymbol{\theta})^2(1-\Phi(\boldsymbol{x}_2^\top\boldsymbol{\theta}))^2}\boldsymbol{x}_2\boldsymbol{x}_2^\top, \\
&\ldots, \\
&\frac{-\boldsymbol{x}_{n_A}^\top\phi(\boldsymbol{x}_{n_A}^\top\boldsymbol{\theta})}{\Phi(\boldsymbol{x}_{n_A}^\top\boldsymbol{\theta})(1-\Phi(\boldsymbol{x}_{n_A}^\top\boldsymbol{\theta}))}\boldsymbol{x}_{n_A}\boldsymbol{x}_{n_A}^\top - \frac{\phi(\boldsymbol{x}_{n_A}^\top\boldsymbol{\theta}))^2\left(1-2\Phi(\boldsymbol{x}_{n_A}^\top\boldsymbol{\theta}))\right)}{\Phi(\boldsymbol{x}_{n_A}^\top\boldsymbol{\theta})^2(1-\Phi(\boldsymbol{x}_{n_A}^\top\boldsymbol{\theta}))^2}\boldsymbol{x}_{n_A}\boldsymbol{x}_{n_A}^\top\Bigg).
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{W}_{Bp} = diag\Bigg(&\frac{\boldsymbol{x}_1^\top\boldsymbol{\theta}\phi(\boldsymbol{x}_1^\top\boldsymbol{\theta})}{(1-\Phi(\boldsymbol{x}_1^\top\boldsymbol{\theta}))}\boldsymbol{x}_1\boldsymbol{x}_1^\top - \frac{\phi(\boldsymbol{x}_1^\top\boldsymbol{\theta})^2}{\left(1-\Phi(\boldsymbol{x}_1^\top\boldsymbol{\theta})\right)}, \\
&\frac{\boldsymbol{x}_2^\top\boldsymbol{\theta}\phi(\boldsymbol{x}_2^\top\boldsymbol{\theta})}{(1-\Phi(\boldsymbol{x}_2^\top\boldsymbol{\theta}))}\boldsymbol{x}_2\boldsymbol{x}_2^\top - \frac{\phi(\boldsymbol{x}_2^\top\boldsymbol{\theta})^2}{\left(1-\Phi(\boldsymbol{x}_2^\top\boldsymbol{\theta})\right)}, \\
&\ldots, \\
&\frac{\boldsymbol{x}_{N_B}^\top\boldsymbol{\theta}\phi(\boldsymbol{x}_{N_B}^\top\boldsymbol{\theta})}{(1-\Phi(\boldsymbol{x}_{N_B}^\top\boldsymbol{\theta}))}\boldsymbol{x}_{N_B}\boldsymbol{x}_{N_B}^\top - \frac{\phi(\boldsymbol{x}_{N_B}^\top\boldsymbol{\theta})^2}{\left(1-\Phi(\boldsymbol{x}_{N_B}^\top\boldsymbol{\theta})\right)}\Bigg).
\end{aligned}$$

Proof. First, let's define the Hessian as:

$$\mathrm{H}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \mathrm{U}(\boldsymbol{\theta})$$

where $\mathrm{U}(\boldsymbol{\theta})$ is the gradient of the log-likelihood. Additionally, we define:

$$\mathbf{W} = \frac{\partial^2 \ell}{\partial \eta}$$

The Hessian matrix for the log-likelihood function $\ell(\boldsymbol{\theta})$ is given by:

$$\mathbf{W} = \sum_{i \in S_A} \frac{\phi\left(\eta\right)^2 \left(2\Phi\left(\eta\right) - 1\right)}{\left(\Phi\left(\eta\right) - 1\right)^2 \Phi\left(\eta\right)^2} - \sum_{i \in S_B} d_i^B \frac{\phi(\eta)^2}{\left(\Phi(\eta) - 1\right)^2}$$

Finally, the Hessian matrix $\mathrm{H}_p(\boldsymbol{\theta})$ is given by:

$$\begin{aligned}
\mathrm{H}_p(\boldsymbol{\theta}) &= \sum_{i \in S_A} \frac{-\boldsymbol{x}_i^\top \phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})}{\Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})(1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta}))} \boldsymbol{x}_i \boldsymbol{x}_i^\top - \sum_{i \in S_A} \frac{\phi(\boldsymbol{x}_i^\top \boldsymbol{\theta}))^2 \left(1 - 2\Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta}))\right)}{\Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})^2 (1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta}))^2} \boldsymbol{x}_i \boldsymbol{x}_i^\top \\
&- \sum_{i \in S_B} \frac{\boldsymbol{x}_i^\top \boldsymbol{\theta} \phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})}{(1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta}))} \boldsymbol{x}_i \boldsymbol{x}_i^\top - \sum_{i \in S_B} \frac{\phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})^2}{\left(1 - \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})\right)} \\
&= \boldsymbol{X}_A^\top \boldsymbol{W}_{Ap} \boldsymbol{X}_A - \boldsymbol{X}_B^\top \boldsymbol{W}_{Bp} \boldsymbol{X}_B,
\end{aligned}$$

This completes the proof of the Hessian matrix $\mathrm{H}(\boldsymbol{\theta})$.

□

3.4.2 GEE

As in Section 3.3.2, we will show the formulas for the derivative of the function G for the probit link function.

Note 3.8. *Let* $\mathrm{J}(\boldsymbol{\theta}) = \frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$. *Then for* $h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \boldsymbol{x}_i \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^{-1}$, *we have*

$$\mathrm{J}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i \in S_A} \frac{\dot{\pi_i^A}\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\theta}\right)}{\pi_i^A \left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\theta}\right)^2} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} \tag{3.11}$$

and for $h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \boldsymbol{x}_i$ *we have*

$$\mathrm{J}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i \in S_B} \frac{\dot{\pi_i^A}\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\theta}\right)}{\pi_i^B} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} \tag{3.12}$$

Proof. Case 1: $h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \boldsymbol{x}_i \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^{-1}$

For this case, $\mathbf{G}(\boldsymbol{\theta})$ can be rewritten as:

$$\mathbf{G}(\boldsymbol{\theta}) = \sum_{i \in S_A} \boldsymbol{x}_i \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^{-1} - \sum_{i \in S_B} d_i^B \boldsymbol{x}_i$$

To find the Jacobian $\mathrm{J}(\boldsymbol{\theta})$, we differentiate $\mathbf{G}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$:

$$\mathrm{J}(\boldsymbol{\theta}) = \frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

Differentiating the first sum:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \left(\sum_{i \in S_A} \boldsymbol{x}_i \pi \left(\boldsymbol{x}_i, \boldsymbol{\theta} \right)^{-1} \right) = - \sum_{i \in S_A} \frac{\partial \pi \left(\boldsymbol{x}_i, \boldsymbol{\theta} \right)}{\partial \boldsymbol{\theta}} \cdot \frac{\boldsymbol{x}_i}{\pi \left(\boldsymbol{x}_i, \boldsymbol{\theta} \right)^2}$$

Using the probit link function, where $\pi(\boldsymbol{x}_i, \boldsymbol{\theta}) = \Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})$, the derivative is:

$$\frac{\partial \pi \left(\boldsymbol{x}_i, \boldsymbol{\theta} \right)}{\partial \boldsymbol{\theta}} = \phi(\boldsymbol{x}_i^\top \boldsymbol{\theta}) \boldsymbol{x}_i$$

Substituting this into the expression:

$$\mathrm{J}(\boldsymbol{\theta}) = - \sum_{i \in S_A} \frac{\phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})}{\Phi(\boldsymbol{x}_i^\top \boldsymbol{\theta})^2} \boldsymbol{x}_i \boldsymbol{x}_i^\top = -\frac{1}{N} \sum_{i \in S_A} \frac{\dot{\pi_i^A} \left(\boldsymbol{x}_i^\top \boldsymbol{\theta} \right)}{\pi_i^A \left(\boldsymbol{x}_i^\top \boldsymbol{\theta} \right)^2} \boldsymbol{x}_i \boldsymbol{x}_i^\top$$

This completes the proof for the first case.

Case 2: $h \left(\boldsymbol{x}_i, \boldsymbol{\theta} \right) = \boldsymbol{x}_i$

For this case, $\mathbf{G}(\boldsymbol{\theta})$ can be rewritten as:

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

To find the Jacobian $\mathrm{J}(\boldsymbol{\theta})$, differentiate $\mathbf{G}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$:

$$\mathrm{J}(\boldsymbol{\theta}) = \frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

Differentiating the second sum:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \left(\sum_{i \in S_B} d_i^B \pi \left(\boldsymbol{x}_i, \boldsymbol{\theta} \right) \boldsymbol{x}_i \right) = \sum_{i \in S_B} d_i^B \frac{\partial \pi \left(\boldsymbol{x}_i, \boldsymbol{\theta} \right)}{\partial \boldsymbol{\theta}} \boldsymbol{x}_i$$

Substituting the expression for $\frac{\partial \pi(\boldsymbol{x}_i, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$:

$$\mathrm{J}(\boldsymbol{\theta}) = - \sum_{i \in S_B} d_i^B \phi(\boldsymbol{x}_i^\top \boldsymbol{\theta}) \boldsymbol{x}_i \boldsymbol{x}_i^\top$$

Given that $d_i^B = \frac{1}{\pi_i^B}$, we have:

$$\mathrm{J}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i \in S_B} \frac{\dot{\pi_i^A} \left(\boldsymbol{x}_i^\top \boldsymbol{\theta} \right)}{\pi_i^B} \boldsymbol{x}_i \boldsymbol{x}_i^\top$$

This completes the proof for the second case.

□

3.5 Complementary log-log regression

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

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

3.5.1 MLE

Likelihood function can be easily obtained as

$$\ell^*(\boldsymbol{\theta}) = \sum_{i \in S_A} \Big\{ \log\{1 - \exp\Big(-\exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big)\Big)\} + \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \Big\} - \sum_{i \in S_B} d_i^B \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big)$$

Note 3.9. *Let* $\mathrm{U}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} l^*(\boldsymbol{\theta})$ *and* $\mathbf{U} = \frac{\partial \ell}{\partial \boldsymbol{\eta}}$. *Then we have*

$$\mathrm{U}_c(\boldsymbol{\theta}) = \sum_{i \in S_A} \frac{\exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \boldsymbol{x}_i}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} - \sum_{i \in S_B} d_i^B \exp\Big(\boldsymbol{x}_i^T \boldsymbol{\theta}\Big) \boldsymbol{x}_i.$$

Proof.

$$\begin{aligned}
\mathrm{U}_c(\boldsymbol{\theta}) &= \sum_{i \in S_A} \frac{1}{1 - \exp\Big(-\exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big)\Big)} \exp\Big(-\exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big)\Big) \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \boldsymbol{x}_i \\
&\quad + \sum_{i \in S_A} \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \boldsymbol{x}_i \\
&= \sum_{i \in S_A} \left\{ \frac{\exp\Big(-\exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big)\Big)}{1 - \exp\Big(-\exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big)\Big)} + 1 \right\} \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) - \sum_{i \in S_B} d_i^B \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \boldsymbol{x}_i \\
&= \sum_{i \in S_A} \left\{ \frac{1 - \pi(\boldsymbol{x}_i, \boldsymbol{\theta})}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} + 1 \right\} \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \boldsymbol{x}_i \\
&= \sum_{i \in S_A} \frac{1}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \boldsymbol{x}_i \\
&= \sum_{i \in S_A} \frac{\exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \boldsymbol{x}_i}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} - \sum_{i \in S_B} d_i^B \exp\Big(\boldsymbol{x}_i^T \boldsymbol{\theta}\Big) \boldsymbol{x}_i.
\end{aligned}$$

□

Note 3.10. *Let* $\mathrm{H}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \mathrm{U}(\boldsymbol{\theta})$.

$$\begin{aligned}
\mathrm{H}(\boldsymbol{\theta}) &= \sum_{i \in S_A} \frac{\exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big)}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} \left\{ 1 - \frac{\exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big)}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} + \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \right\} \boldsymbol{x}_i \boldsymbol{x}_i^\top - \sum_{i \in S_B} d_i^B \exp\Big(\boldsymbol{x}_i^\top \boldsymbol{\theta}\Big) \boldsymbol{x}_i \boldsymbol{x}_i^\top \\
&= \boldsymbol{X}_A^\top \boldsymbol{W}_{Ac} \boldsymbol{X}_A - \boldsymbol{X}_B^\top \boldsymbol{W}_{Bc} \boldsymbol{X}_B,
\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{W}_{Ac} = diag\Bigg(&\frac{\exp\left(\boldsymbol{x}_1^\top\boldsymbol{\theta}\right)}{\pi(\boldsymbol{x}_1,\boldsymbol{\theta})}\left\{1-\frac{\exp\left(\boldsymbol{x}_1^\top\boldsymbol{\theta}\right)}{\pi(\boldsymbol{x}_1,\boldsymbol{\theta})}+\exp\left(\boldsymbol{x}_1^\top\boldsymbol{\theta}\right)\right\},\\ &\frac{\exp\left(\boldsymbol{x}_2^\top\boldsymbol{\theta}\right)}{\pi(\boldsymbol{x}_2,\boldsymbol{\theta})}\left\{1-\frac{\exp\left(\boldsymbol{x}_2^\top\boldsymbol{\theta}\right)}{\pi(\boldsymbol{x}_2,\boldsymbol{\theta})}+\exp\left(\boldsymbol{x}_2^\top\boldsymbol{\theta}\right)\right\},\\ &\ldots,\\ &\frac{\exp\left(\boldsymbol{x}_{n_A}^\top\boldsymbol{\theta}\right)}{\pi(\boldsymbol{x}_{n_A},\boldsymbol{\theta})}\left\{1-\frac{\exp\left(\boldsymbol{x}_{n_A}^\top\boldsymbol{\theta}\right)}{\pi(\boldsymbol{x}_{n_A},\boldsymbol{\theta})}+\exp\left(\boldsymbol{x}_{n_A}^\top\boldsymbol{\theta}\right)\right\}\Bigg)\end{aligned}$$

and

$$\boldsymbol{W}_{Bc} = diag\left(d_1^B\exp\left(\boldsymbol{x}_1^\top\boldsymbol{\theta}\right), d_2^B\exp\left(\boldsymbol{x}_2^\top\boldsymbol{\theta}\right), \ldots, d_{n_B}^B\exp\left(\boldsymbol{x}_{n_B}^\top\boldsymbol{\theta}\right)\right).$$

Proof.

$$\begin{aligned}\frac{\partial\,\mathrm{U}_c(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} &= \sum_{i\in S_A}\frac{\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top(1-\exp\left(-\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\right)) - \exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\boldsymbol{x}_i\exp\left(-\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\right)\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\boldsymbol{x}_i}{(1-\exp\left(-\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\right))^2}\\ &- \sum_{i\in S_B} d_i^B\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top\\ &= \sum_{i\in S_A}\frac{\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)(1-\exp\left(-\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\right)) - \exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\exp\left(-\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\right)\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)}{(1-\exp\left(-\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\right))^2}\boldsymbol{x}_i\boldsymbol{x}_i^\top\\ &- \sum_{i\in S_B} d_i^B\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top\\ &= \sum_{i\in S_A}\frac{\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\left\{\pi(\boldsymbol{x}_i,\boldsymbol{\theta}) - \exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)(1-\pi(\boldsymbol{x}_i,\boldsymbol{\theta}))\right\}}{\pi(\boldsymbol{x}_i,\boldsymbol{\theta})^2}\boldsymbol{x}_i\boldsymbol{x}_i^\top - \sum_{i\in S_B} d_i^B\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top\\ &= \sum_{i\in S_A}\frac{\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\left\{\pi(\boldsymbol{x}_i,\boldsymbol{\theta}) - \exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right) + \exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\pi(\boldsymbol{x}_i,\boldsymbol{\theta})\right\}}{\pi(\boldsymbol{x}_i,\boldsymbol{\theta})^2}\boldsymbol{x}_i\boldsymbol{x}_i^\top - \sum_{i\in S_B} d_i^B\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top\\ &= \sum_{i\in S_A}\frac{\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)}{\pi(\boldsymbol{x}_i,\boldsymbol{\theta})}\left\{1-\frac{\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)}{\pi(\boldsymbol{x}_i,\boldsymbol{\theta})}+\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\right\}\boldsymbol{x}_i\boldsymbol{x}_i^\top - \sum_{i\in S_B} d_i^B\exp\left(\boldsymbol{x}_i^\top\boldsymbol{\theta}\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top\\ &= \boldsymbol{X}_A^\top\boldsymbol{W}_{Ac}\boldsymbol{X}_A - \boldsymbol{X}_B^\top\boldsymbol{W}_{Bc}\boldsymbol{X}_B,\end{aligned}$$

□

3.5.2 Generalized Estimating Equations

And similarly as in Section 3.3.2, we can derive the expressions for the derivative of the function G, which will follow a similar form but adjusted for the characteristics of the complementary log–log transformation.

Note 3.11. *Let* $\mathrm{J}(\boldsymbol{\theta}) = \frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$. *Then for* $h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \boldsymbol{x}_i \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^{-1}$, *we have*

$$\mathrm{J}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i \in S_A} \frac{1 - \pi_i^A\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\theta}\right)}{\pi_i^A\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\theta}\right)^2} \exp\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\theta}\right) \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} \tag{3.13}$$

and for $h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \boldsymbol{x}_i$ *we have*

$$\mathrm{J}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i \in S_B} \frac{1 - \pi_i^A\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\theta}\right)}{\pi_i^B} \exp\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\theta}\right) \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} \tag{3.14}$$

Proof. Case 1: $h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \boldsymbol{x}_i \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^{-1}$

For this case, $\mathbf{G}(\boldsymbol{\theta})$ can be rewritten as:

$$\mathbf{G}(\boldsymbol{\theta}) = \sum_{i \in S_A} \boldsymbol{x}_i \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^{-1} - \sum_{i \in S_B} d_i^B \boldsymbol{x}_i$$

To find the Jacobian $\mathrm{J}(\boldsymbol{\theta})$, we differentiate $\mathbf{G}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$:

$$\mathrm{J}(\boldsymbol{\theta}) = \frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

Differentiating the first sum:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \left(\sum_{i \in S_A} \boldsymbol{x}_i \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^{-1} \right) = - \sum_{i \in S_A} \frac{\partial \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)}{\partial \boldsymbol{\theta}} \cdot \frac{\boldsymbol{x}_i}{\pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^2}$$

Using the cloglog link function, where $\pi(\boldsymbol{x}_i, \boldsymbol{\theta}) = 1 - \exp\left(-\exp\left(\boldsymbol{x}_i^{\top} \boldsymbol{\theta}\right)\right)$, the derivative is:

$$\frac{\partial \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)}{\partial \boldsymbol{\theta}} = \exp\left(\boldsymbol{x}_i^{\top} \boldsymbol{\theta}\right) \left(1 - \pi(\boldsymbol{x}_i, \boldsymbol{\theta})\right) \boldsymbol{x}_i$$

Substituting this into the expression:

$$\mathrm{J}(\boldsymbol{\theta}) = - \sum_{i \in S_A} \frac{1 - \pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)}{\pi\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)^2} \exp\left(\boldsymbol{x}_i^{\top} \boldsymbol{\theta}\right) \boldsymbol{x}_i \boldsymbol{x}_i^{\top}$$

Thus, we have:

$$\mathrm{J}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i \in S_A} \frac{1 - \pi_i^A\left(\boldsymbol{x}_i^{\top} \boldsymbol{\theta}\right)}{\pi_i^A\left(\boldsymbol{x}_i^{\top} \boldsymbol{\theta}\right)^2} \exp\left(\boldsymbol{x}_i^{\top} \boldsymbol{\theta}\right) \boldsymbol{x}_i \boldsymbol{x}_i^{\top}$$

This completes the proof for the first case.

Case 2: $h\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) = \boldsymbol{x}_i$

For this case, $\mathbf{G}(\boldsymbol{\theta})$ can be rewritten as:

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

To find the Jacobian $\mathrm{J}(\boldsymbol{\theta})$, differentiate $\mathbf{G}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$:

$$\mathrm{J}(\boldsymbol{\theta}) = \frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

Differentiating the second sum:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \left(\sum_{i \in S_B} d_i^B \pi \left(\boldsymbol{x}_i, \boldsymbol{\theta} \right) \boldsymbol{x}_i \right) = \sum_{i \in S_B} d_i^B \frac{\partial \pi \left(\boldsymbol{x}_i, \boldsymbol{\theta} \right)}{\partial \boldsymbol{\theta}} \boldsymbol{x}_i$$

Substituting the expression for $\frac{\partial \pi(\boldsymbol{x}_i, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$:

$$\mathrm{J}(\boldsymbol{\theta}) = - \sum_{i \in S_B} d_i^B \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta} \right) \left(1 - \pi(\boldsymbol{x}_i, \boldsymbol{\theta}) \right) \boldsymbol{x}_i \boldsymbol{x}_i^\top$$

Given that $d_i^B = \frac{1}{\pi_i^B}$, we have:

$$\mathrm{J}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i \in S_B} \frac{1 - \pi_i^A \left(\boldsymbol{x}_i^\top \boldsymbol{\theta} \right)}{\pi_i^B} \exp\left(\boldsymbol{x}_i^\mathrm{T} \boldsymbol{\theta} \right) \boldsymbol{x}_i \boldsymbol{x}_i^\mathrm{T}$$

This completes the proof for the second case.

□

Tables 3.2, 3.3, 3.4 provide a summary of the methods described in this chapter. It includes the form of the likelihood function, its gradient and the Hessian for the maximum likelihood method, and the objective function and its derivative for the GEE method, with a distinction for the form of the h function.

Table 3.2: Summary of logit estimation method

Component	**GEE**	**MLE**
Log-likelihood function	Not defined	$\sum_{i \in S_\mathrm{A}} \log\left(\frac{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})}{1 - \pi(\boldsymbol{x}_i, \boldsymbol{\theta})} \right)$ $+ \sum_{i \in S_\mathrm{B}} d_i^\mathrm{B} \log\left(1 - \pi(\boldsymbol{x}_i, \boldsymbol{\theta}) \right)$
Objective function / Gradient	$\sum_{i \in S_A} \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \pi(\boldsymbol{x}_i, \boldsymbol{\theta}) \boldsymbol{x}_i$	$\sum_{i \in S_A} \frac{\boldsymbol{x}_i}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} - \sum_{i \in S_B} d_i^B \boldsymbol{x}_i$
Hessian	$- \sum_{i \in S_B} \frac{1}{\pi_i^B} \pi_i^A(\boldsymbol{x}_i^\mathrm{T} \boldsymbol{\theta})$ x $(1 - \pi_i^A(\boldsymbol{x}_i^\mathrm{T} \boldsymbol{\theta})) \boldsymbol{x}_i \boldsymbol{x}_i^\mathrm{T}$	$\boldsymbol{X}_A^\top \boldsymbol{W}_{Ap} \boldsymbol{X}_A - \boldsymbol{X}_B^\top \boldsymbol{W}_{Bp} \boldsymbol{X}_B$

Table 3.3: Summary of cloglog estimation method

Component	**GEE**	**MLE**
Log-likelihood function	Not defined	$\sum_{i \in S_A} \left[\log\left(1 - \exp\left(- \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta} \right) \right) \right) \right.$ $\left. + \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta} \right) \right] - \sum_{i \in S_B} d_i^B \exp\left(\boldsymbol{x}_i^\top \boldsymbol{\theta} \right)$
Objective function / Gradient	$\sum_{i \in S_A} \boldsymbol{x}_i - \sum_{i \in S_B} d_i^B \pi(\boldsymbol{x}_i, \boldsymbol{\theta}) \boldsymbol{x}_i$	$\sum_{i \in S_A} \frac{\boldsymbol{x}_i}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} - \sum_{i \in S_B} d_i^B \boldsymbol{x}_i$
Hessian	$-\frac{1}{N} \sum_{i \in S_B} \frac{1 - \pi_i^A(\boldsymbol{x}_i^\mathrm{T} \boldsymbol{\theta})}{\pi_i^B} \exp\left(\boldsymbol{x}_i^\mathrm{T} \boldsymbol{\theta} \right) \boldsymbol{x}_i \boldsymbol{x}_i^\mathrm{T}$	$\boldsymbol{X}_A^\top \boldsymbol{W}_{Ac} \boldsymbol{X}_A - \boldsymbol{X}_B^\top \boldsymbol{W}_{Bc} \boldsymbol{X}_B$

Table 3.4: Summary of probit estimation method

Component	**GEE**	**MLE**
Log-likelihood function	Not defined	$\sum_{i\in S_A} \log\left(\frac{\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})}{1-\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})}\right)$ $+\sum_{i\in S_B} d_i^B \log\left(1-\Phi(\boldsymbol{x}_i^\top\boldsymbol{\theta})\right)$
Objective function / Gradient	$\sum_{i\in S_A}\boldsymbol{x}_i - \sum_{i\in S_B} d_i^B\pi(\boldsymbol{x}_i,\boldsymbol{\theta})\boldsymbol{x}_i$	$\sum_{i\in S_A}\frac{\boldsymbol{x}_i}{\pi(\boldsymbol{x}_i,\boldsymbol{\theta})} - \sum_{ß\in S_B} d_i^B\boldsymbol{x}_i$
Hessian	$-\frac{1}{N}\sum_{i\in S_B}\frac{\pi_i^A(\boldsymbol{x}_i^\mathrm{T}\boldsymbol{\theta})}{\pi_i^B}\boldsymbol{x}_i\boldsymbol{x}_i^\mathrm{T}$	$\boldsymbol{X}_A^\top\boldsymbol{W}_{Ap}\boldsymbol{X}_A - \boldsymbol{X}_B^\top\boldsymbol{W}_{Bp}\boldsymbol{X}_B$

3.6 Variance estimation

In this section, we look at variance estimation for the Inverse Probability Weighting (IPW) estimator. Variance estimation is critical to understanding the precision and reliability of an estimator. By estimating the variance, we can construct confidence intervals and perform hypothesis testing, which are essential for statistical inference. This section will explore both analytical and bootstrap approaches to variance estimation, providing a comprehensive framework for assessing the variability of the IPW estimator. We will focus on both the maximum likelihood and estimating equations methods. The idea and solution method will be the same in both cases, as we will be using the Taylor expansion of the respective functions, which will differ in form in the dependence and link function. Therefore, for each combination of methods we will obtain a different form of variance estimate.

3.6.1 Analytical approach

We begin by presenting the form of the analytical variance of the IPW estimator as a function of the chosen link function describing the inclusion probability. This approach was first presented in Chen, Li, and Wu (2020) and mentioned in Wu (2022). However, the authors demonstrated estimation for a logit link function, while we were able to extend this approach with complementary log–log and probit versions. First, we present the general reasoning behind the derivation, and then we present its form for each link function. Let us start with the basic assumptions.

(D1) The population size N and the sample sizes n_A and n_B satisfy $\lim_{N\to\infty} n_A/N = f_A \in (0,1)$ and $\lim_{N\to\infty} n_B/N = f_B \in (0,1)$.

(D2) For each $\boldsymbol{x}, \partial m(\boldsymbol{x},\boldsymbol{\beta})/\partial\boldsymbol{\beta}$ is continuous in $\boldsymbol{\beta}$ and $|\partial m(\boldsymbol{x},\boldsymbol{\beta})/\partial\boldsymbol{\beta}| \leq h(\boldsymbol{x},\boldsymbol{\beta})$ for $\boldsymbol{\beta}$ in the neighborhood of $\boldsymbol{\beta}_0$, and $N^{-1}\sum_{i=1}^N h\left(\boldsymbol{x}_i,\boldsymbol{\beta}_0\right) = O(1)$.

(D3) For each $\boldsymbol{x}, \partial^2 m(\boldsymbol{x},\boldsymbol{\beta})/\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^\top$ is continuous in $\boldsymbol{\beta}$ and $\max_{j,l}|\partial^2 m(\boldsymbol{x},\boldsymbol{\beta})/\partial\beta_j\partial\beta_l| \leq k(\boldsymbol{x},\boldsymbol{\beta})$ for $\boldsymbol{\beta}$ in the neighborhood of $\boldsymbol{\beta}_0$, and $N^{-1}\sum_{i=1}^N k\left(\boldsymbol{x}_i,\boldsymbol{\beta}_0\right) = O(1)$.

(D4) The finite population and the sampling design for S_B satisfy $N^{-1}\sum_{i\in S_B} d_i^B \boldsymbol{u}_i - N^{-1}\sum_{i=1}^N \boldsymbol{u}_i = O_p\left(n_B^{-1/2}\right)$ for $\boldsymbol{u}_i = \boldsymbol{x}_i, y_i$ and $m(\boldsymbol{x}_i, \boldsymbol{\beta})$.

(D5) There exist c_1 and c_2 such that $0 < c_1 \leq N\pi_i^A/n_A \leq c_2$ and $0 < c_1 \leq N\pi_i^B/n_B \leq c_2$ for all units i.

(D6) The finite population and the propensity scores satisfy $N^{-1}\sum_{i=1}^N y_i^2 = O(1)$, $N^{-1}\sum_{i=1}^N \|\boldsymbol{x}_i\|^3 = O(1)$, and $N^{-1}\sum_{i=1}^N \pi_i^A\left(1-\pi_i^A\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top$ is a positive definite matrix.

Generally we need to find such $\hat{\boldsymbol{\eta}}^\top = \left(\hat{\mu}, \hat{\boldsymbol{\theta}}^\top\right)$ that is solution to the following system of equations:

$$\boldsymbol{\Phi}_n(\boldsymbol{\eta}) = \begin{pmatrix} \frac{1}{N}\sum_{i=1}^N \left[\frac{R_i^A(y_i-\mu)}{\pi_i^A} + \Delta\frac{R_i^A-\pi_i^A}{\pi_i^A}\right] \\ \frac{1}{N}\,\mathrm{U}(\boldsymbol{\theta}) \end{pmatrix} = \mathbf{0}, \tag{3.15}$$

where $\mathrm{U}(\boldsymbol{\theta})$ is one of the objective functions derived from the estimation methods described before (e.g. gradient of maximum likelihood function or estimating function from 3.4).

For a well-specified model for propensity score, we have $E\{\boldsymbol{\Phi}_n(\boldsymbol{\eta})\} = \mathbf{0}$. If conditions D1-D6 hold then we can write that $\boldsymbol{\Phi}_n(\hat{\boldsymbol{\eta}}) = \mathbf{0}$ and $\boldsymbol{\Phi}_n(\boldsymbol{\eta}_0) = O_p\left(n_A^{-1/2}\right)$. To derive the variance from the IPW estimator under asymptotic conditions we will use the first-order Taylor expansion of the function $\boldsymbol{\Phi}_n(\hat{\boldsymbol{\eta}})$ around $\boldsymbol{\eta}_0$.

In the following steps, we will dissect the process of calculating the variance of the estimator. Importantly, we will divide it into a part calculated on the basis of a non-probability sample as well as a probability sample. We will then apply the general formula to the various parameter estimation methods. We will also distinguish the variance of the estimated parameters $\hat{\boldsymbol{\theta}}$.

Applying first-order Taylor expansion we have

$$\begin{aligned} \hat{\boldsymbol{\eta}} - \boldsymbol{\eta}_0 &= \left[\boldsymbol{\phi}_n(\boldsymbol{\eta}_0)\right]^{-1}\boldsymbol{\Phi}_n(\boldsymbol{\eta}_0) + o_p\left(n_A^{-1/2}\right) \\ &= \left[E\left\{\boldsymbol{\phi}_n(\boldsymbol{\eta}_0)\right\}\right]^{-1}\boldsymbol{\Phi}_n(\boldsymbol{\eta}_0) + o_p\left(n_A^{-1/2}\right), \end{aligned} \tag{3.16}$$

where $\boldsymbol{\phi}_n(\boldsymbol{\eta}) = \partial\boldsymbol{\Phi}_n(\boldsymbol{\eta})/\partial\boldsymbol{\eta}$. It follows that

$$\begin{aligned} \mathrm{Var}(\hat{\boldsymbol{\eta}}) &= E\left[(\hat{\boldsymbol{\eta}} - \boldsymbol{\eta}_0)^2\right] \\ &= \left[E\left\{\boldsymbol{\phi}_n(\boldsymbol{\eta}_0)\right\}\right]^{-1}\mathrm{Var}\left\{\boldsymbol{\Phi}_n(\boldsymbol{\eta}_0)\right\}\left[E\left\{\boldsymbol{\phi}_n(\boldsymbol{\eta}_0)\right\}^\top\right]^{-1} + o\left(n_A^{-1}\right). \end{aligned} \tag{3.17}$$

with $\boldsymbol{\phi}_n(\boldsymbol{\eta})$ varying depending on the chosen link function. In the following subsections, we will continue the variance calculations, taking into account the technical differences.

3.6.2 Variance estimation for Maximum Likelihood method

Logit link function

$$\boldsymbol{\phi}_n(\boldsymbol{\eta}) = \frac{1}{N}\begin{pmatrix} -\sum_{i=1}^N\left\{\left(1-\frac{\Delta}{\mu}\right)\frac{R_i^A}{\pi_i^A} + \frac{\Delta}{\mu}\right\} & -\sum_{i=1}^N \frac{1-\pi_i^A}{\pi_i^A}R_i^A\left(y_i - \mu + \Delta\right)\boldsymbol{x}_i^\top \\ \mathbf{0} & \mathrm{H}(\boldsymbol{\theta}) \end{pmatrix},$$

where $\mathrm{H}(\boldsymbol{\theta})$ regards to hessian of MLE approach or Jacobian of GEE approach. Further it can be shown that

$$E(\phi_n(\boldsymbol{\eta})) = \begin{pmatrix} -1 & -\frac{1}{N}\sum_{i=1}^{N} \\ \mathbf{0} & \frac{1}{N}\,\mathrm{E}\left(\mathrm{H}_p(\boldsymbol{\theta})\right) \end{pmatrix}$$

what leads to

$$\left[E\left\{\boldsymbol{\phi}_n(\boldsymbol{\eta})\right\}\right]^{-1} = \begin{pmatrix} -1 & \frac{\Delta}{\mu}\mathbf{b}_1^\top + \left(1-\frac{\Delta}{\mu}\right)\mathbf{b}_2^\top \\ \mathbf{0} & \frac{1}{N}\,\mathrm{E}\left(\mathrm{H}(\boldsymbol{\theta})^{-1}\right) \end{pmatrix}, \tag{3.18}$$

where

$$\mathbf{b}_1^\top = \left\{N^{-1}\sum_{i=1}^{N}\left(1-\pi_i^{\mathrm{A}}\right)y_i\boldsymbol{x}_i^\top\right\}\left\{N^{-1}\sum_{i=1}^{N}\pi_i^{\mathrm{A}}\left(1-\pi_i^{\mathrm{A}}\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top\right\}^{-1}$$

$$\mathbf{b}_2^\top = \left\{N^{-1}\sum_{i=1}^{N}\left(1-\pi_i^{\mathrm{A}}\right)(y_i-\mu_y)\,\boldsymbol{x}_i^\top\right\}\left\{N^{-1}\sum_{i=1}^{N}\pi_i^{\mathrm{A}}\left(1-\pi_i^{\mathrm{A}}\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top\right\}^{-1}.$$

Note that we can decompose the system of equations from (3.15) into a sum of two systems $\boldsymbol{A}_1 + \boldsymbol{A}_2$. Let us also assume that we can write the $\mathrm{U}(\boldsymbol{\theta})$ function as the sum of two functions, one based on units from a probability sample and the other on units from a non-probability sample. We then have

$$\mathbf{A}_1 = \begin{pmatrix} \frac{1}{N}\sum_{i=1}^{N}\left[\frac{R_i^A(y_i-\mu)}{\pi_i^A} + \Delta\frac{R_i^A-\pi_i^A}{\pi_i^A}\right] \\ \mathrm{u}_1(\boldsymbol{\theta}) \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} 0 \\ \mathrm{u}_2(\boldsymbol{\theta}) \end{pmatrix},$$

where $\mathrm{U}(\boldsymbol{\theta}) = \mathrm{u}_1(\boldsymbol{\theta}) + \mathrm{u}_2(\boldsymbol{\theta})$.

Let us therefore calculate the variance of the two components and then derive the asymptotic variance of μ_{IPW} estimator. Let $\boldsymbol{V}_1 = \mathrm{Var}\left(\boldsymbol{A}_1\right)$ and $\boldsymbol{V}_2 = \mathrm{Var}\left(\boldsymbol{A}_2\right)$. Then we have

$$\mathrm{Var}\left(\boldsymbol{A}_1\right) = \frac{1}{N^2}\sum_{i=1}^{N}\begin{pmatrix} (y_i-\mu+\Delta)^2\left(\frac{1-\pi_i^A}{\pi_i^A}\right) & \mathrm{v}_{12}\,\boldsymbol{x}_i^\top \\ \mathrm{v}_{21}\,\boldsymbol{x}_i & \mathrm{v}_{22}\,\boldsymbol{x}_i\boldsymbol{x}_i^\top \end{pmatrix}$$

where

$$\mathrm{v}_{12} = \mathrm{v}_{21} = \left(1-\pi_i^A\right)(y_i-\mu+\Delta)$$

$$\mathrm{v}_{22} = \pi_i^A\left(1-\pi_i^A\right)$$

and

$$\mathrm{Var}\left(\boldsymbol{A}_2\right) = \begin{pmatrix} 0 & \mathbf{0}^\top \\ \mathbf{0} & \boldsymbol{D} \end{pmatrix},$$

where $\mathbf{D} = N^{-2}V_p\left(\sum_{i\in S_B} d_i^B\pi_i^A\boldsymbol{x}_i\right)$ is the design–based variance–covariance matrix under the probability sampling p for sample S_B.

Probit link function

$$\boldsymbol{\phi}_n(\boldsymbol{\eta}) = \frac{1}{N}\begin{pmatrix} -\sum_{i=1}^{N}\left\{\left(1-\frac{\Delta}{\mu}\right)\frac{R_i^A}{\pi_i^A}+\frac{\Delta}{\mu}\right\} & -\sum_{i=1}^{N}\frac{R_i^A\left(y_i-\mu+\frac{\Delta}{\mu}\right)\dot{\pi}_i^A}{\left(\pi_i^A\right)^2}\boldsymbol{x}_i^\top \\ \mathbf{0} & \mathrm{H}(\boldsymbol{\theta}) \end{pmatrix}.$$

Further we have

$$E(\phi_n(\boldsymbol{\eta})) = \begin{pmatrix} -1 & -\frac{1}{N}\sum_{i=1}^{N}\frac{\left(y_i-\mu+\frac{\Delta}{\mu}\right)\dot{\pi}_i^A}{\pi_i^A}\boldsymbol{x}_i^\top \\ \mathbf{0} & \frac{1}{N}\,\mathrm{E}\left(\mathrm{H}_p(\boldsymbol{\theta})\right) \end{pmatrix}$$

$$\Downarrow$$

$$\left[E(\phi_n(\boldsymbol{\eta}))\right]^{-1} = \begin{pmatrix} -1 & -\sum_{i=1}^{N}\frac{\left(y_i-\mu+\frac{\Delta}{\mu}\right)\dot{\pi}_i^A}{\pi_i^A}\boldsymbol{x}_i^\top\left[\mathrm{E}\left(\mathrm{H}_p(\boldsymbol{\theta})\right)\right]^{-1} \\ \mathbf{0} & \left[\frac{1}{N}\,\mathrm{E}\left(\mathrm{H}_p(\boldsymbol{\theta})\right)\right]^{-1} \end{pmatrix}.$$

$\mathrm{Var}\left\{\boldsymbol{\Phi}_n\left(\boldsymbol{\eta}_0\right)\right\}$ can be found using decomposition of $\boldsymbol{\Phi}_n(\boldsymbol{\eta}) = \boldsymbol{A}_1 + \boldsymbol{A}_2$. Then $\mathrm{Var}\left\{\boldsymbol{\Phi}_n\left(\boldsymbol{\eta}_0\right)\right\} = \mathrm{Var}\left(\boldsymbol{A}_1\right) + \mathrm{Var}\left(\boldsymbol{A}_2\right)$. Let

$$\boldsymbol{A}_1 = \frac{1}{N}\sum_{i=1}^{N}\begin{pmatrix} \frac{R_i(y_i-\mu)}{\pi_i^A}+\Delta\frac{R_i^A-\pi_i^A}{\pi_i^A} \\ R_i^A\frac{\dot{\pi}_i^A}{\pi_i^A\left(1-\pi_i^A\right)}\boldsymbol{x}_i - \pi_i^A\frac{\dot{\pi}_i^A}{\pi_i^A\left(1-\pi_i^A\right)}\boldsymbol{x}_i \end{pmatrix}$$

and

$$\boldsymbol{A}_2 = \frac{1}{N}\begin{pmatrix} 0 \\ \sum_{i=1}^{N}\pi_i^A\frac{\dot{\pi}_i^A}{\pi_i^A\left(1-\pi_i^A\right)} - \sum_{i\in S_B} d_i^B\frac{\dot{\pi}_i^A}{1-\pi_i^A}\boldsymbol{x}_i \end{pmatrix}$$

Further we have

$$\mathrm{Var}\left(\boldsymbol{A}_1\right) = \frac{1}{N^2}\sum_{i=1}^{N}\begin{pmatrix} \left(y_i-\mu+\Delta\right)^2\left(\left(\pi_i^A\right)^{-1}-1\right) & \frac{\dot{\pi}_i^A(y_i-\mu+\Delta)}{\pi_i^A}\boldsymbol{x}_i^\top \\ \frac{\dot{\pi}_i^A(y_i-\mu+\Delta)}{\pi_i^A}\boldsymbol{x}_i & \frac{\dot{\pi}_i^A}{\pi_i^A\left(1-\pi_i^A\right)}\boldsymbol{x}_i\boldsymbol{x}_i^\top \end{pmatrix}$$

and

$$\mathrm{Var}\left(\boldsymbol{A}_2\right) = \begin{pmatrix} 0 & \mathbf{0}^\top \\ \mathbf{0} & \boldsymbol{D} \end{pmatrix},$$

where

$$\boldsymbol{D} = \frac{1}{N^2}V_p\left(\sum_{i\in S_B} d_i^B\frac{\dot{\pi}_i^A}{1-\pi_i^A}\boldsymbol{x}_i\right) \tag{3.19}$$

is variance covariance matrix.

Complementary log-log link function

$$\boldsymbol{\phi}_n(\boldsymbol{\eta}) = \frac{1}{N}\begin{pmatrix} -\sum_{i=1}^{N}\left\{\left(1-\frac{\Delta}{\mu}\right)\frac{R_i^A}{\pi_i^A}+\frac{\Delta}{\mu}\right\} & \sum_{i=1}^{N}\frac{R_i^A(y_i-\mu+\Delta)\left(1-\pi_i^A\right)\log\left(1-\pi_i^A\right)}{\left(\pi_i^A\right)^2}\boldsymbol{x}_i^\top \\ \mathbf{0} & \mathrm{H}(\boldsymbol{\theta}) \end{pmatrix}$$

$$[E(\phi_n(\boldsymbol{\eta}))]^{-1} = \begin{pmatrix} -1 & \sum_{i=1}^{N} \frac{(y_i-\mu+\Delta)\left(1-\pi_i^A\right)\log\left(1-\pi_i^A\right)}{\pi_i^A} \boldsymbol{x}_i^\top \left[\mathrm{E}\left(\mathrm{H}_c(\boldsymbol{\theta})\right)\right]^{-1} \\ \mathbf{0} & \left[\frac{1}{N}\mathrm{E}\left(\mathrm{H}_c(\boldsymbol{\theta})\right)\right]^{-1} \end{pmatrix}.$$

Further we have

$$\boldsymbol{A}_1 = \frac{1}{N}\sum_{i=1}^{N} \begin{pmatrix} \frac{R_i^A(y_i-\mu)}{\pi_i^A} + \Delta\frac{R_i^A-\pi_i^A}{\pi_i^A} \\ \pi_i^A \frac{\log\left(1-\pi_i^A\right)}{\pi_i^A}\boldsymbol{x}_i - R_i^A\frac{\log\left(1-\pi_i^A\right)}{\pi_i^A}\boldsymbol{x}_i \end{pmatrix}$$

and

$$\boldsymbol{A}_2 = \frac{1}{N} \begin{pmatrix} 0 \\ \sum_{i\in S_B} d_i^B \log\left(1-\pi_i^A\right)\boldsymbol{x}_i - \sum_{i=1}^{N}\pi_i^A\frac{\log\left(1-\pi_i^A\right)}{\pi_i^A}\boldsymbol{x}_i \end{pmatrix}$$

$$\Downarrow$$

$$\mathrm{Var}\left(\boldsymbol{A}_1\right) = \frac{1}{N^2}\sum_{i=1}^{N} \begin{pmatrix} (y_i-\mu+\Delta)^2\left((\pi_i^A)^{-1}-1\right) & -\frac{\left(1-\pi_i^A\right)}{\pi_i^A}\log\left(1-\pi_i^A\right)(y_i-\mu+\Delta)\boldsymbol{x}_i^\top \\ -\frac{\left(1-\pi_i^A\right)}{\pi_i^A}\log\left(1-\pi_i^A\right)(y_i-\mu+\Delta)\boldsymbol{x}_i & \frac{\left(1-\pi_i^A\right)\log^2\left(1-\pi_i^A\right)}{\pi_i^A}\boldsymbol{x}_i\boldsymbol{x}_i^\top \end{pmatrix}$$

and

$$\mathrm{Var}\left(\boldsymbol{A}_2\right) = \begin{pmatrix} 0 & \mathbf{0}^\top \\ \mathbf{0} & \boldsymbol{D} \end{pmatrix},$$

where

$$\boldsymbol{D} = \frac{1}{N^2}V_p\left(\sum_{i\in S_B} d_i^B\log\left(1-\pi_i^A\right)\boldsymbol{x}_i\right) \tag{3.20}$$

is variance-covariance matrix.

3.6.3 Variance estimation for Generalized Estimating Equations method

In this method, the system shown in (3.14) can be written as

$$\boldsymbol{\Phi}_n(\boldsymbol{\eta}) = \begin{pmatrix} \frac{1}{N}\sum_{i=1}^{N}\left[\frac{R_i^A(y_i-\mu)}{\pi_i^A} + \Delta\frac{R_i^A-\pi_i^A}{\pi_i^A}\right] \\ \frac{1}{N}\sum_{i=1}^{N} R_i^A h\left(\boldsymbol{x}_i\right) - \frac{1}{N}\sum_{i\in S_B} d_i^B\pi_i^A h\left(\boldsymbol{x}_i\right) \end{pmatrix} = \mathbf{0}. \tag{3.21}$$

Using the same approach as in Section 3.6.2 we can show that the decomposition of $\boldsymbol{\Phi}_n(\boldsymbol{\eta})$ may consist of

$$\mathbf{A}_1 = \frac{1}{N}\sum_{i=1}^{N}\begin{pmatrix} \frac{R_i^A(y_i-\mu)}{\pi_i^A} + \Delta\frac{R_i-\pi_i^A}{\pi_i^A} \\ R_i^A h\left(\boldsymbol{x}_i\right)\end{pmatrix}, \quad \mathbf{A}_2 = \frac{1}{N}\begin{pmatrix} 0 \\ -\sum_{i\in S_B} d_i^B\pi_i^A h\left(\boldsymbol{x}_i\right)\end{pmatrix}. \tag{3.22}$$

It can be shown that for $h\left(\boldsymbol{x}_i\right) = \boldsymbol{x}_i$

$$\mathbf{V}_1 = \frac{1}{N^2}\sum_{i=1}^{N}\begin{pmatrix} \left\{\left(1-\pi_i^A\right)/\pi_i^A\right\}(y_i-\mu+\Delta)^2 & \left(1-\pi_i^A\right)(y_i-\mu+\Delta)\boldsymbol{x}_i^\top \\ \left(1-\pi_i^A\right)(y_i-\mu+\Delta)\boldsymbol{x}_i & \pi_i^A\left(1-\pi_i^A\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top \end{pmatrix}$$

and

$$\mathbf{V}_2 = \begin{pmatrix} 0 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{D} \end{pmatrix},$$

where

$$\mathbf{D} = N^{-2} V_p \left(\sum_{i \in S_B} d_i^B \pi_i^A \boldsymbol{x}_i \right)$$

and for $h(\boldsymbol{x}_i) = \boldsymbol{x}_i \pi_i^A(\boldsymbol{x}_i, \boldsymbol{\theta})^{-1}$

$$\mathbf{V}_1 = \frac{1}{N^2} \sum_{i=1}^{N} \begin{pmatrix} \left\{ \left(1 - \pi_i^A\right) / \pi_i^A \right\} (y_i - \mu + \Delta)^2 & \frac{\left(1-\pi_i^A\right)}{\pi_i^A} (y_i - \mu + \Delta) \boldsymbol{x}_i^\top \\ \frac{\left(1-\pi_i^A\right)}{\pi_i^A} (y_i - \mu + \Delta) \boldsymbol{x}_i & \left(1 - \pi_i^A\right) \boldsymbol{x}_i \boldsymbol{x}_i^\top \end{pmatrix}$$

and

$$\mathbf{V}_2 = \begin{pmatrix} 0 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{D} \end{pmatrix}$$

where

$$\mathbf{D} = N^{-2} V_p \left(\sum_{i \in S_B} d_i^B \boldsymbol{x}_i \right).$$

The missing element to obtain the values of function (3.15) is the derivative of $\boldsymbol{\Phi}_n(\boldsymbol{\eta})$ and the inverse of its expected value, which varies depending on the assumptions made about the linking function and the form of the function h.

In general, for all linking functions we can write $\boldsymbol{\phi}_n(\boldsymbol{\eta})$ as

$$\phi_n(\boldsymbol{\eta}) = \frac{1}{N} \begin{pmatrix} -\sum_{i=1}^{N} \frac{R_i^A}{\pi_i^A} & \phi_{12} \\ 0 & \mathrm{J}(\boldsymbol{\theta}) \end{pmatrix} \tag{3.23}$$

and inverse of its expectation values is given as

$$\left[E \left\{ \boldsymbol{\phi}_n(\boldsymbol{\eta}) \right\} \right]^{-1} = \begin{pmatrix} -1 & \mathbf{b}^\top \\ \mathbf{0} & \mathrm{J}(\boldsymbol{\theta})^{-1} \end{pmatrix}, \tag{3.24}$$

where for **logit** we have

$$\phi_{12} = -\sum_{i=1}^{N} \frac{1 - \pi_i^A}{\pi_i^A} R_i (y_i - \mu) \boldsymbol{x}_i^\top,$$

$$\mathbf{b}^\top = \begin{cases} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(1 - \pi_i^A\right) (y_i - \mu) \boldsymbol{x}_i^\top \right\} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(1 - \pi_i^A\right) \boldsymbol{x}_i \boldsymbol{x}_i^\top \right\}^{-1} & \text{if } h(\boldsymbol{x}_i) = \boldsymbol{x}_i \pi_i^A \left(\boldsymbol{x}_i^\mathrm{T} \boldsymbol{\theta} \right)^{-1}, \\ \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(1 - \pi_i^A\right) (y_i - \mu) \boldsymbol{x}_i^\top \right\} \left\{ \frac{1}{N} \sum_{i=1}^{N} \pi_i^A \left(1 - \pi_i^A\right) \boldsymbol{x}_i \boldsymbol{x}_i^\top \right\}^{-1} & \text{if } h(\boldsymbol{x}_i) = \boldsymbol{x}_i. \end{cases}$$

for **cloglog** we have

$$\phi_{12} = \sum_{i=1}^{N} \frac{R_i^A (y_i - \mu) \left(1 - \pi_i^A\right) \log \left(1 - \pi_i^A\right)}{\left(\pi_i^A\right)^2} \boldsymbol{x}_i^\top,$$

$$\mathbf{b}^\top = \sum_{i=1}^{N} \frac{R_i^A \left(y_i - \mu\right) \left(1 - \pi_i^A\right) \log\left(1 - \pi_i^A\right)}{\left(\pi_i^A\right)^2} \boldsymbol{x}_i^\top$$

and for **probit** we have

$$\phi_{12} = -\sum_{i=1}^{N} \frac{R_i^A \left(y_i - \mu\right) \dot{\pi}_i^A}{\left(\pi_i^A\right)^2} \boldsymbol{x}_i^\top,$$

$$\mathbf{b}^\top = \left\{ \frac{1}{N} \sum_{i=1}^{N} \frac{\dot{\pi_i^A}}{\pi_i^A} \left(y_i - \mu\right) \boldsymbol{x}_i^\top \right\} \left\{ \frac{1}{N} \sum_{i=1}^{N} \pi_i^A \boldsymbol{x}_i \boldsymbol{x}_i^\top \right\}^{-1}.$$

As you can be seen, the vector $\mathbf{b}^\top$ is only variable for the logit link function. In the other cases it has the same form regardless of the chosen function h. From equation 3.15 we know that the asymptotic variance of $\hat{\boldsymbol{\eta}}$ is equal to $\left[E\left\{\boldsymbol{\phi}_n\left(\boldsymbol{\eta}_0\right)\right\}\right]^{-1} \left(\mathrm{Var}\left(\boldsymbol{A}_1\right) + \mathrm{Var}\left(\boldsymbol{A}_2\right)\right) \left[E\left\{\boldsymbol{\phi}_n\left(\boldsymbol{\eta}_0\right)\right\}^\top\right]^{-1}$, where the first diagonal element is the variance of $\hat{\mu}_{IPW}$ and the rest are variances of $\hat{\boldsymbol{\theta}}$.

3.6.4 Bootstrap approach

This approach helps to approximate the sampling distribution of the IPW estimator and provides an estimate of its variance. By generating multiple bootstrap samples, we can better understand the variability and reliability of the estimator, which is particularly useful when dealing with complex survey data or non-probability samples. The algorithm 9 shows a scheme for calculating the variance based on the bootstrap samples generated. Briefly, in each iteration we artificially generate probability and non-probability samples corresponding to the weights of the units, count the propensity score estimator, and then estimate its variance based on L estimated population means.

Algorithm 9 Bootstrap variance estimator

1: Sample n_A units from S_A with replacement to create S'_A (if pseudo-weights are present inclusion probabilities should be proportional to their inverses).
2: Sample n_B units from S_B according to the design weights for probability sample to create S'_B.
3: Estimate propensity model $\pi_i^A = P\left(R_i^A = 1 \mid \boldsymbol{x}_i\right)$ based on sampled units from step 1
4: Compute bootstrap mean as

$$\hat{\mu} = \frac{1}{\hat{N}^{\mathrm{A}}} \sum_{i \in S'_{\mathrm{A}}} \frac{y_i}{\hat{\pi}_i^A}.$$

5: Repeat steps 1-4 L times (we set $L = 500$ in our simulations).
6: Estimate variance term as $\hat{V} = \frac{1}{L-1} \sum_{i=1}^{L} \left(\hat{\mu}_i - \bar{\mu}\right)^2$ where $\bar{\mu} = \frac{1}{L} \sum_{i=1}^{L} \hat{\mu}_i$.

Chapter 4

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.

4.1 Joint Randomization Approach

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

The selection mechanism for the non–probability sample is modeled as in chapter 3. 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 as explained in Chapter 2. With known design

weights d_i^B for $i \in S_B$ we can define the DR estimator as

$$\hat{\mu}_{\mathrm{DR}} = \frac{1}{\hat{N}^{\mathrm{A}}} \sum_{i \in S_{\mathrm{A}}} d_i^{\mathrm{A}} \left\{ y_i - m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right) \right\} + \frac{1}{\hat{N}^{\mathrm{B}}} \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}\right), \tag{4.1}$$

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.

4.2 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, Kim, and Song, 2020, the idea is to determine the equations leading to the estimation of the $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ parameters based on the bias of the population mean estimator. In contrast to the joint randomization approach, this method allows for the estimation of the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ in a single step, rather than in two separate steps.

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

$$\begin{aligned} \operatorname{bias}\left(\hat{\mu}_{DR}\right) =& \mid \hat{\mu}_{DR} - \mu \mid \\ &= \frac{1}{N} \sum_{i=1}^{N} \left\{ \frac{R_i^A}{\pi_i^A\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\theta}\right)} - 1 \right\} \left\{ y_i - m\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta}\right) \right\} \\ &+ \frac{1}{N} \sum_{i=1}^{N} \left(R_i^B d_i^B - 1 \right) m\left(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{\beta}\right) \end{aligned} \tag{4.2}$$

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

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

where

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

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

$$\begin{pmatrix} \sum_{i=1}^{N} R_i^A \left\{ \frac{1}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} - 1 \right\} \left\{ y_i - m\left(\boldsymbol{x}_i, \boldsymbol{\beta}\right) \right\} \boldsymbol{x}_i \\ \sum_{i=1}^{N} \frac{R_i^A}{\pi(\boldsymbol{x}_i, \boldsymbol{\theta})} \frac{\partial m(\boldsymbol{x}_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} - \sum_{i \in S_{\mathrm{B}}} d_i^{\mathrm{B}} \frac{\partial m(\boldsymbol{x}_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \end{pmatrix} = \mathbf{0}, \tag{4.3}$$

which can ve solved using Newton–Raphson optimization method.

4.3 Variance Estimation

In this section, we will describe the asymptotic properties of the estimators depending on the method used to estimate their parameters.

4.3.1 Analytical approach

Recall that we have two ways of counting the parameters $(\boldsymbol{\beta}, \boldsymbol{\theta})$, corresponding to the parametric methods – MI and IPW respectively. The first results from the chapters 2 and 1. Here the parameters are counted separately. The second follows from the method described in this chapter, where we rely on minimising the bias, which leads us to solve a system of equations with respect to the $(\boldsymbol{\beta}, \boldsymbol{\theta})$ vector.

Therefore, we will also present two variance estimators corresponding to each method.

Variance estimator for the joint randomisation approach

As far as doubly robust estimators are concerned, a natural approach is to estimate the parameters separately for the part that includes the inverse probability weighting method, then similarly for the mass imputation method, and in the last step calculate the value of the estimator from formula 4.1. We call such a method a joint randomisation approach because we are combining two separate methods. This allows the estimator to be robust to misspecification of a given part of the estimator. Such a method yields the most effective results in terms of estimation bias. let us look at the variance form of this estimator. First, we introduce some definitions.

Definition 4.1. *Let* $\mathrm{W} = \frac{1}{N^2}\mathrm{V}_p\left(\sum_{i\in S_B} d_i^B t_i\right)$ *be the design-based variance derived from probability sample, where*

1. *For the logit model,*

$$t_i = \pi_i^{\mathrm{A}} \boldsymbol{x}_i^\top \mathbf{b}_3 + m\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right) - N^{-1}\sum_{i=1}^{N} m\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right). \tag{4.4}$$

2. *For the probit model,*

$$t_i = \frac{\dot{\pi}_i^A}{1-\pi_i^A}\boldsymbol{x}_i^\top \mathbf{b}_3 + \mathrm{m}\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right) - \frac{1}{N}\sum_{i=1}^{N} \mathrm{m}\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right). \tag{4.5}$$

3. *For the complementary log-log model,*

$$t_i = \log\left(1-\pi_i^A\right)\boldsymbol{x}_i^\top \mathbf{b}_3 + \mathrm{m}\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right) - \frac{1}{N}\sum_{i=1}^{N} \mathrm{m}\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right). \tag{4.6}$$

Definition 4.2. *Let* $h_{\mathrm{N}} = N^{-1}\sum_{i=1}^{N}\left(y_i - m\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right)\right)$ *be the vector of residuals.*

Let us also make the following note, proof of which can be found in the supplementary materials of Chen, Li, and Wu (2020)

Note 4.1. . *The outcome regression model parameters* $\boldsymbol{\beta}$ *have no impact on the asymptotic variance of* μ_{DR}*.*

In the following steps, we will present the asymptotic form of the variance of the DR estimator depending on the assumed link function for the modelled inclusion probability. Similar to the propensity score approach, the main idea comes from Chen, Li, and Wu (2020) and we extend this approach to additional link functions.

Theorem 4.1. *Suppose the* $\boldsymbol{\theta}$ *vector has been estimated by maximum likelihood and* π_i^A *is modelled by logistic regression. Then we have*

$$\begin{aligned}\operatorname{Var}\left(\hat{\mu}_{DR}\right) = \frac{1}{N^2}\sum_{i=1}^{N}\left(1-\pi_i^{\mathrm{A}}\right)\pi_i^{\mathrm{A}}\left[\left\{y_i - m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right) - h_{\mathrm{N}}\right\}/\right.\\ \left.\pi_i^{\mathrm{A}} - \mathbf{b}_3^{\top}\boldsymbol{x}_i\right]^2 + W + o\left(n_A^{-1}\right),\end{aligned} \tag{4.7}$$

where $\mathbf{b}_3^{\top} = \left[N^{-1}\sum_{i=1}^{N}\left(1-\pi_i^{\mathrm{A}}\right)\left\{y_i - m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right) - h_{\mathrm{N}}\right\}\boldsymbol{x}_i^{\top}\right]\left\{N^{-1}\sum_{i=1}^{N}\pi_i^{\mathrm{A}}\left(1-\pi_i^{\mathrm{A}}\right)\boldsymbol{x}_i\boldsymbol{x}_i^{\top}\right\}^{-1}$.

Proof. Notice that the first part of $\hat{\mu}_{DR}$ *given in (4.1) is the IPW estimator (*$\hat{\mu}_{IPW}$ *given in (3.1) with* y_i *replaced with* $y_i - m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)$*. Using the asymptotic expansion derived in (3.15), we have*

$$\begin{aligned}\frac{1}{\hat{N}^A}\sum_{i=1}^{N}\frac{R_i\left\{y_i - m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)\right\}}{\hat{\pi}_i^A} =& h_N + \frac{1}{N}\sum_{i=1}^{N}R_i\left\{\frac{y_i - m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right) - h_N}{\pi_i^A} - \mathbf{b}_3^{\top}\boldsymbol{x}_i\right\}\\ &+ \mathbf{b}_3^{\top}\frac{1}{N}\sum_{i\in S_B}d_i^B\pi_i^A\boldsymbol{x}_i + o_p\left(n_A^{-1/2}\right)\end{aligned}$$

where $h_N = N^{-1}\sum_{i=1}^{N}\left\{y_i - m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)\right\}$ *and* $\mathbf{b}_3^{\top}$ *as in theorem. The second part of the DR estimator is the Hajek estimator under the sampling design for* S_B*, which has the following expansion*

$$\frac{1}{\hat{N}^B}\sum_{i\in S_B}d_i^B m_i = \frac{1}{N}\sum_{i=1}^{N}m_i + \frac{1}{N}\sum_{i\in S_B}d_i^B\left\{m_i - \frac{1}{N}\sum_{i=1}^{N}m_i\right\} + O_p\left(n_B^{-1}\right),$$

where $m_i = m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)$*. Merging the two parts leads us to*

$$\hat{\mu}_{DR2} - \mu_y = \frac{1}{N}\sum_{i=1}^{N}R_i\left\{\frac{y_i - m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right) - h_N}{\pi_i^A} - \mathbf{b}_3^{\top}\boldsymbol{x}_i\right\} + \frac{1}{N}\sum_{i\in S_B}d_i^B t_i + o_p\left(n_A^{-1/2}\right),$$

where $t_i = \pi_i^A\boldsymbol{x}_i^{\top}\mathbf{b}_3 + m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right) - N^{-1}\sum_{i=1}^{N}m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)$.

Deriving the asymptotic variance as $\mathrm{E}\left[\left(\hat{\mu}_{DR2}-\mu_y\right)^2\right]$ *we obtain the formula as in the theorem where*

$$W = N^{-2}V_p\left(\sum_{i\in S_{\mathrm{s}}} d_i^{\mathrm{B}} t_i\right)$$

□

Theorem 4.2. *Suppose the* $\boldsymbol{\theta}$ *vector has been estimated by the method of maximum likelihood and* π_i^A *is modelling by probit regression. Then we have*

$$\begin{aligned}\mathrm{Var}\left(\hat{\mu}_{DR}\right) =& \frac{1}{N^2}\sum_{i=1}^{N}\left(1-\pi_i^A\right)\pi_i^A\left[\frac{y_i-\mathrm{m}\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)-h_n}{\pi_i^A}-\mathbf{b}_3^T\frac{\dot{\pi}_i^A}{\pi_i^A\left(1-\pi_i^A\right)}\boldsymbol{x}_i\right]^2 \\ &+\mathrm{W}+o\left(n_A^{-1}\right),\end{aligned} \tag{4.8}$$

where $\mathbf{b}_3^\top = -\sum_{i=1}^N \frac{\left(y_i-\mathrm{m}(\boldsymbol{x}_i,\boldsymbol{\beta}^*)-h_n\right)\dot{\pi}_i^A}{\pi_i^A}\boldsymbol{x}_i^\top\left[\mathrm{E}\left(\mathrm{H}_p(\boldsymbol{\theta})\right)\right]^{-1}$.

Proof. By analogy to logit approach we can show that for probit we have

$$\begin{aligned}\frac{1}{\hat{N}^A}\sum_{i=1}^{N}\frac{R_i\left\{y_i-m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)\right\}}{\hat{\pi}_i^A} =& h_N+\frac{1}{N}\sum_{i=1}^{N}R_i\left\{\frac{y_i-m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)-h_N}{\pi_i^A}-\mathbf{b}_3^\top\frac{\dot{\pi}_i^A}{\pi_i^A\left(1-\pi_i^A\right)}\boldsymbol{x}_i\right\} \\ &+\mathbf{b}_3^\top\frac{1}{N}\sum_{i\in S_B}d_i^B\frac{\dot{\pi}_i^A}{\left(1-\pi_i^A\right)}\boldsymbol{x}_i+o_p\left(n_A^{-1/2}\right),\end{aligned}$$

where $h_n=\frac{1}{N}\sum_{i=1}^N\left(y_i-m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)\right)$ *and*

$$\mathbf{b}_3^\top = -\sum_{i=1}^{N}\frac{\left(y_i-m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)-h_n\right)\dot{\pi}_i^A}{\pi_i^A}\boldsymbol{x}_i^\top\left[\mathrm{E}\left(\mathrm{H}_p(\boldsymbol{\theta})\right)\right]^{-1}.$$

The second part of the estimator has the same expansion as logit what implies that:

$$\begin{aligned}\hat{\mu}_{DR2}-\mu_y =& \frac{1}{N}\sum_{i=1}^{N}R_i\left\{\frac{y_i-m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)-h_N}{\pi_i^A}-\mathbf{b}_3^\top\frac{\dot{\pi}_i^A}{\pi_i^A\left(1-\pi_i^A\right)}\boldsymbol{x}_i\right\} \\ &+\frac{1}{N}\sum_{i\in S_B}d_i^B t_i+o_p\left(n_A^{-1/2}\right)\end{aligned}$$

It follows that the variance has the formula introduced in the theorem

$$\mathrm{W}=\frac{1}{N^2}\,\mathrm{V}_p\left(\sum_{i\in S_B}d_i^B t_i\right)$$

and $t_i=\frac{\dot{\pi}_i^A}{1-\pi_i^A}\boldsymbol{x}_i^\top\mathbf{b}_3+m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)-\frac{1}{N}\sum_{i=1}^N m\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)$. □

Theorem 4.3. *Suppose the* $\boldsymbol{\theta}$ *vector has been estimated by the method of maximum likelihood and* π_i^A *is modelling by complementary log-log link function. Then we have*

$$\begin{aligned}\mathrm{Var}\left(\hat{\mu}_{DR}\right) =& \frac{1}{N^2}\sum_{i=1}^{N}\left(1-\pi_i^A\right)\pi_i^A\left[\frac{y_i-\mathrm{m}\left(\boldsymbol{x}_i,\boldsymbol{\beta}^*\right)-h_n}{\pi_i^A}-\mathbf{b}_3^T\frac{\log\left(1-\pi_i^A\right)}{\pi_i^A}\boldsymbol{x}_i\right]^2 \\ &+\mathrm{W}+o\left(n_A^{-1}\right),\end{aligned} \tag{4.9}$$

where $\mathbf{b}_3^\top = -\sum_{i=1}^N \frac{(y_i - \mathrm{m}(\boldsymbol{x}_i, \boldsymbol{\beta}^*) - h_n)\left(1-\pi_i^A\right)\log\left(1-\pi_i^A\right)}{\pi_i^A}\boldsymbol{x}_i^\top \left[\mathrm{E}\left(\mathrm{H}_l(\boldsymbol{\theta})\right)\right]^{-1}$.

Proof. Similarly we can show that for complementary log-log

$$\frac{1}{\hat{N}^A}\sum_{i=1}^N \frac{R_i\left\{y_i - m\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right)\right\}}{\hat{\pi}_i^A} = h_N + \frac{1}{N}\sum_{i=1}^N R_i\left\{\frac{y_i - m\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right) - h_N}{\pi_i^A} - \mathbf{b}_3^\top \frac{\log\left(1-\pi_i^A\right)}{\pi_i^A}\boldsymbol{x}_i\right\} + \mathbf{b}_3^\top \frac{1}{N}\sum_{i\in S_B} d_i^B \log\left(1-\pi_i^A\right)\boldsymbol{x}_i + o_p\left(n_A^{-1/2}\right),$$

The second part of the $\hat{\mu}_{DR}$ *ca be derived the same way as for logit and probit what implies that:*

$$\hat{\mu}_{DR2} - \mu_y = \frac{1}{N}\sum_{i=1}^N R_i\left\{\frac{y_i - m\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right) - h_N}{\pi_i^A} - \mathbf{b}_3^\top \frac{\log\left(1-\pi_i^A\right)}{\pi_i^A}\boldsymbol{x}_i\right\} + \frac{1}{N}\sum_{i\in S_B} d_i^B t_i + o_p\left(n_A^{-1/2}\right),$$

and the variance can be written as in theorem, where

$$\mathrm{W} = \frac{1}{N^2}\,\mathrm{V}_p\left(\sum_{i\in S_B} d_i^B t_i\right)$$

and $t_i = \log\left(1-\pi_i^A\right)\boldsymbol{x}_i^\top \mathbf{b}_3 + m\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right) - \frac{1}{N}\sum_{i=1}^N m\left(\boldsymbol{x}_i, \boldsymbol{\beta}^*\right)$. □

It needs to be noted that V_p refers to the design-based variance under the probability sampling design for S_B and this approach for variance estimation requires correctly specified model for propensity scores.

Estimator of the variance for bias minimization approach

As described in the subsection , the parameters $\boldsymbol{\beta}, \boldsymbol{\theta}$, i.e. for the mass imputation and propensity score methods respectively, can be estimated jointly using a bias-minimisation approach. To repeat, this method involves deriving the bias of the DR estimator and then finding parameters that minimise it. Let us now make the following assumptions:

(C1) **Parameter Compactness**: The parameter α belongs to a compact subset in $\mathbb{R}^{2p}$, and the true parameter α^* lies in the interior of this subset.

(C2) **Covariate Boundedness**: The covariates $\boldsymbol{x}_i$ for $i \in U$ are fixed and uniformly bounded.

(C3) **Matrix Eigenvalue Conditions**: There exist constants c_1 and c_2 such that

$$0 < c_1 \leq \lambda_{\min}\left(\frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i^T \boldsymbol{x}_i\right) \leq \lambda_{\max}\left(\frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i^T \boldsymbol{x}_i\right) \leq c_2 < \infty,$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalues of a matrix, respectively.

(C1) **Residual Conditions**: For the residuals $\epsilon_i(\beta) = y_i - m(\boldsymbol{x}_i^T\beta)$, there exist constants c_3, c_4, and c_5 such that

$$E[|\epsilon_i(\beta^*)|^{2+\delta}] \leq c_3 \quad \text{and} \quad E[\exp\{c_4|\epsilon_i(\beta^*)|\}|\boldsymbol{x}_i] \leq c_5,$$

for all $1 \leq i \leq N$ and some $\delta > 0$.

(C4) **Model Function Boundedness**: The first three derivatives of the model function $m(\cdot)$, evaluated at $\boldsymbol{x}_i^T\beta$, are uniformly bounded away from infinity on the set

$$N_{\alpha,\tau} = \left\{\alpha \in \mathbb{R}^{2p} : \|\alpha_{M_\alpha} - \alpha^*_{M_\alpha}\| \leq \tau\sqrt{\frac{s_\alpha}{n}}, \alpha M^c_\alpha = 0\right\},$$

for some $\tau > 0$.

(C5) **Non-zero Coefficient Conditions**: The minimum absolute values of the non-zero coefficients in θ^* and β^*, denoted by λ_θ and λ_β, respectively, tend to infinity as $n \to \infty$.

(C6) **Sparsity Conditions**: The sparsity measure $s_\alpha = s_\theta + s_\beta$ satisfies

$$s_\alpha = o(n^{1/3}), \quad \lambda_\theta, \lambda_\beta \to 0, \quad \log(n)^2 = o(n\lambda_\theta^2),$$

and other similar conditions that restrict the dimensions of covariates p and the true non-zero coefficients s_α.

Below is the theorem described in the article Yang and Kim (2020).

Theorem 4.4. *Under the assumptions that either the sampling score model $\pi_B(\boldsymbol{x}^T\theta)$ or the outcome model $m(\boldsymbol{x}^T\beta)$ is correctly specified, the doubly robust estimator $\hat{\mu}_{dr}(\hat{\theta}, \hat{\beta})$ is asymptotically normal with variance $V = \lim_{n\to\infty}(V_1 + V_2)$, where:*

$$V_1 = \mathbb{E}\left\{\frac{n}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\pi^B_{ij} - \pi^B_i\pi^B_j)\frac{m(\boldsymbol{x}_i^T\beta^*)}{\pi^B_i}\frac{m(\boldsymbol{x}_j^T\beta^*)}{\pi^B_j}\right\}$$

$$V_2 = \frac{n}{N^2}\sum_{i=1}^{N}\mathbb{E}\left[\left(\frac{R^A_i}{\pi^A_i(\boldsymbol{x}_i^T\theta^*)} - 1\right)^2(y_i - m(\boldsymbol{x}_i^T\beta^*))^2\right]$$

Proof. We start with the doubly robust estimator $\hat{\mu}_{\mathrm{dr}}(\hat{\theta}, \hat{\beta})$:

$$\hat{\mu}_{\mathrm{dr}}(\hat{\theta}, \hat{\beta}) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{R^A_i}{\pi_B(\boldsymbol{x}_i^T\hat{\alpha})}\left(y_i - m(\boldsymbol{x}_i^T\hat{\beta})\right) + R^B_i d^B_i m(\boldsymbol{x}_i^T\hat{\beta})\right]$$

Using the Taylor series expansion around the true parameters α^* and β^*, we have:

$$\sqrt{n}\left(\hat{\mu}_{\mathrm{dr}}(\hat{\theta}, \hat{\beta}) - \mu\right) = \sqrt{n}\left(\hat{\mu}_{\mathrm{dr}}(\theta^*, \beta^*) - \mu\right) + \sqrt{n}\left(\frac{\partial\hat{\mu}_{\mathrm{dr}}(\theta, \beta)}{\partial(\theta, \beta)}\bigg|_{(\theta^*,\beta^*)}\right)(\hat{\theta} - \theta^*, \hat{\beta} - \beta^*)^T + o_p(1)$$

Since $\hat{\mu}_{\text{dr}}(\theta^*, \beta^*)$ is unbiased if either the model $\pi_A(X^T\theta)$ or $m(X^T\beta)$ is correctly specified, the first term $\sqrt{n}(\hat{\mu}_{\text{dr}}(\theta^*, \beta^*) - \mu)$ converges to a normal distribution:

$$\sqrt{n}(\hat{\mu}_{\text{dr}}(\theta^*, \beta^*) - \mu) \xrightarrow{d} N(0, V_1 + V_2)$$

We now derive the components V_1 and V_2. V_1 is the variance component related to the sampling variance of the Horvitz-Thompson estimator:

$$V_1 = \mathbb{E}\left\{ \frac{n}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\pi_{A,ij} - \pi_i^B \pi_j^B) \frac{m(\boldsymbol{x}_i^T \beta^*)}{\pi_i^B} \frac{m(\boldsymbol{x}_j^T \beta^*)}{\pi_j^B} \right\}$$

The second term V_2 captures the variance due to the estimation error in the outcome model:

$$V_2 = \frac{n}{N^2} \sum_{i=1}^{N} \mathbb{E}\left[\left(\frac{R_i^A}{\pi_i^A(\boldsymbol{x}_i^T \theta^*)} - 1 \right)^2 (y_i - m(\boldsymbol{x}_i^T \beta^*))^2 \right]$$

To estimate V_1 and V_2, we use the following consistent estimators:

$$\hat{V}_1 = \frac{n}{N^2} \sum_{i \in S_B} \sum_{j \in S_B} \frac{\pi_{ij}^B - \pi_i^A \pi_j^B}{\pi_{ij}^B} \frac{m(\boldsymbol{x}_i^T \hat{\beta})}{\pi_i^B} \frac{m(\boldsymbol{x}_j^T \hat{\beta})}{\pi_j^B}$$

$$\hat{V}_2 = \frac{n}{N^2} \sum_{i=1}^{N} \left[\left(\frac{R_i^A}{\pi_A(\boldsymbol{x}_i^T \hat{\theta})} - 1 \right)^2 (y_i - m(\boldsymbol{x}_i^T \hat{\beta}))^2 \right]$$

Combining these estimators, we obtain the overall estimator for the variance V:

$$\hat{V} = \hat{V}_1 + \hat{V}_2$$

Thus, under the assumptions that either $\pi_B(\boldsymbol{x}^T\theta)$ or $m(\boldsymbol{x}^T\beta)$ is correctly specified, the doubly robust estimator $\hat{\mu}_{\text{dr}}(\hat{\theta}, \hat{\beta})$ is asymptotically normal with variance $V = \lim_{n\to\infty}(V_1 + V_2)$. □

Theorem 4.5. *Estimators of V_1 and V_2 terms are given by*

$$\hat{V}_1 = \frac{n}{N^2} \sum_{i \in S_B} \sum_{j \in S_B} \frac{\pi_{ij}^B - \pi_i^B \pi_j^B}{\pi_{ij}^B} \frac{m\left(\boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}\right)}{\pi_i^B} \frac{m\left(\boldsymbol{x}_j^\top \hat{\boldsymbol{\beta}}\right)}{\pi_j^B} \tag{4.10}$$

and

$$\hat{V}_2 = \frac{n}{N^2} \sum_{i=1}^{N} \left[\left\{ \frac{R_i^A}{\pi_i^A \left(\boldsymbol{x}_i^\top \hat{\boldsymbol{\theta}}\right)^2} - \frac{2R_i^A}{\pi_i^A \left(\boldsymbol{x}_i^\top \hat{\boldsymbol{\theta}}\right)} \right\} \left\{ y_i - m\left(\boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}\right) \right\}^2 + R_i^B d_i^B \hat{\sigma}^2 \left(\boldsymbol{x}_i\right) \right] \tag{4.11}$$

respectively.

Proof. Let $\sigma^2\left(\boldsymbol{x}_i^{\mathrm{T}} \beta^*\right) = E\left[\left\{ y_i - m\left(\boldsymbol{x}_i^{\mathrm{T}} \beta^*\right) \right\}^2\right]$, and let $\hat{\sigma}^2\left(\boldsymbol{x}_i\right)$ be a consistent estimator of $\sigma^2\left(\boldsymbol{x}_i^{\mathrm{T}} \beta^*\right)$. □

It should be noted that the variance estimator has certain limitations. First, it is derived for a point estimator assuming that the population size N is known, which limits its applicability when N is unknown or needs to be estimated. Secondly, the estimator is constructed under the logistic regression model and does not need to be valid if the propensity score model is specified using a different link function (probit or complementary log–log). Thus, using alternative link functions requires a different variance estimation approach. Finally, solutions to the system of equations 4.3.1 may not exist unless the two sets of covariates used in the outcome regression model and the propensity score model have the same dimensions. This is why bootstrap approach can be used as an alternative method.

4.3.2 Bootstrap approach

As with the previous methods, we can use bootstrap variance with bootstrap sampling from probability and non-probability samples. For the probability sample, we will use design weights and an assumed sampling scheme. For the non-probability sample, it will be the imposed weights or the vector **1**. The rest of the scheme involves estimating the parameters in each bootstrap iteration and then calculating the bootstrap mean.

Algorithm 10 Bootstrap variance estimator

1: Sample n_A units from S_A with replacement to create S'_A (if pseudo-weights are present inclusion probabilities should be proportional to their inverses).

2: Sample n_B units from S_B according to the design weights for probability sample to create S'_B.

3: Estimate regression model $\mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}'] = m(\boldsymbol{x}', \boldsymbol{\beta})$ based on $j \in S'_A$ from step 1

4: Estimate propensity model $\pi_i^A = P\left(R_i = 1 \mid \boldsymbol{x}_i\right)$ based on sampled units from step 1

5: Compute bootstrap mean as

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

6: Repeat steps 1-4 L times (we set $L = 500$ in our simulations).

7: Estimate variance term as $\hat{V} = \dfrac{1}{L-1} \sum_{i=1}^{L} (\hat{\mu}_i - \bar{\mu})^2$ where $\bar{\mu} = \dfrac{1}{L} \sum_{i=1}^{L} \hat{\mu}_i$.

Chapter 5

Techniques of variable selection with high-dimensional data

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

In the case of data integration based on probability and non-probability samples, the selection of variables is part of a two-step process leading to the estimation of the mean, where in the first step statistically significant variables are selected and in the second step the model is rebuilt. For the first step, a penalized logistic regression model has been proposed for estimating propensity scores (Yang, Kim, and Song (2020)), but this approach can be extended

to other linking models such as clog-log and probit functions. For mass imputation based on a parametric model, penalized OLS (*Ordinary Least Squared*) is considered. It is worth mentioning that Yang and Kim (2020), in their article on this topic, used the SCAD method (*Smoothly Clipped Absolute Deviation*), but we extend it on other selection techniques such as LASSO and MCP. Let us discuss each of them.

To begin with, it is useful to make a few assumptions. For any vector $\boldsymbol{\theta} \in \mathbb{R}^p$, denote the number of non-zero elements $\boldsymbol{\theta}$ as $\|\boldsymbol{\theta}\|_0 = \Sigma_{j=1}^p I\left(\theta_j \neq 0\right)$, L_1-norm as $\|\boldsymbol{\theta}\|_1 = \Sigma_{j=1}^p |\theta_j|$, the L_2-norm as $\|\boldsymbol{\theta}\|_2 = \sqrt{\Sigma_{j=1}^p \theta_j^2}$ and L_∞-norm as $\|\boldsymbol{\theta}\|_\infty = \max_{1 \leqslant j \leqslant p} |\theta_j|$. For any $\mathcal{J} \subseteq \{1, \ldots, p\}$, let $\boldsymbol{\theta}_\mathcal{J}$ be subvector $\boldsymbol{\theta}$, elements whose indices are in $\mathcal{J}$. Let $\mathcal{J}^{\mathrm{c}}$ be complement $\mathcal{J}$. For any $\mathcal{J}_1, \mathcal{J}_2 \subseteq \{1, \ldots, p\}$ and matrix $\Sigma \in \mathbb{R}^{p \times p}$, let $\Sigma_{\mathcal{J}_1, \mathcal{J}_2}$ be submatrix Σ, formed by the rowsn in $\mathcal{J}_1$ and columns in $\mathcal{J}_2$. According to the literature on variable selection, the dependent variables should first be standardised so that they have variances approximately equal to 1, which makes the variable selection procedure more stable. Let us also make the following assumptions:

(E1) Selection mechanism to S_A sample is consistent with the logistic (or probit, complementary log–log) model, i.e. for $i \in S_A$ have $\pi_i^A = \frac{\exp\left\{\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}_0\right)\right\}}{1+\exp\left\{\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}_0\right)\right\}}$.

(E2) $m\left(\boldsymbol{x}\right)$ function is consistent with one of the generalised linear models described on 2.2.

Let $\mathrm{U}(\boldsymbol{\theta}, \boldsymbol{\beta})$ will be one of the equations described in subsection 3.2.2. Let p will be a large integer, consider penalized estimating functions for $(\boldsymbol{\theta}, \boldsymbol{\beta})$ as

$$\mathrm{U}^p(\boldsymbol{\theta}, \boldsymbol{\beta}) = \mathrm{U}(\boldsymbol{\theta}, \boldsymbol{\beta}) - \begin{pmatrix} q_{\lambda_\theta}(|\boldsymbol{\theta}|)\,\mathrm{sgn}(\boldsymbol{\theta}) \\ q_{\lambda_\beta}(|\boldsymbol{\beta}|)\,\mathrm{sgn}(\boldsymbol{\beta}) \end{pmatrix} \tag{5.1}$$

where q_{λ_θ} and q_{λ_β} are some smooth functions. Let $q_\lambda(x) = \frac{\partial p_\lambda}{\partial x}$, where p_λ is a certain penalising function.

Before moving on to describe the second step, i.e. modelling on matched variables, let us discuss each of the penalization techniques. We will also point out their properties, similarities and differences between them.

5.1 LASSO

The Least Absolute Shrinkage and Selection Operator (LASSO) is a regularization technique that not only helps in reducing overfitting but also in variable selection by imposing a penalty on the absolute size of the regression coefficients. The penalty parameter λ controls the strength of the penalty, where larger values of λ increase the number of coefficients pushed to zero, thus simplifying the model.

LASSO is particularly useful in high-dimensional settings where the number of predictors exceeds the number of observations, as it performs both variable selection and regularization

simultaneously. The LASSO penalization approach tends to produce sparse models, where only a subset of the predictors have non-zero coefficients, making it easier to interpret the model. However, one of the limitations of LASSO is that it can shrink large coefficients too much, potentially leading to biased estimates.

Definition

The LASSO penalty is expressed as

$$p_\lambda(x) = \lambda|x|$$

and its derivative as

$$q_\lambda(x) = \begin{cases} -\lambda & \text{if } x < 0, \\ [-\lambda, \lambda] & \text{if } x = 0 \\ \lambda & \text{if } x > 0 \end{cases}$$

The LASSO penalization effectively adds a linear constraint to the optimization problem, which forces some of the coefficients to be exactly zero. This linear penalty is what distinguishes LASSO from other regularization techniques like Ridge regression, which uses a quadratic penalty and does not perform variable selection.

5.2 SCAD

The Smoothly Clipped Absolute Deviation (SCAD) is another non-convex penalty function designed to overcome the limitations of LASSO by allowing significant coefficients to remain large. Unlike LASSO, which applies a constant penalty regardless of the size of the coefficient, SCAD introduces a penalty that decreases as the coefficient value increases, thus reducing the bias for large coefficients.

SCAD's penalization approach is designed to retain the advantages of LASSO (such as sparsity) while mitigating its tendency to overshrink large coefficients. The non-convex nature of the SCAD penalty helps in achieving more accurate coefficient estimates in cases where some predictors have genuinely large effects.

Definition

The SCAD penalty is expressed as

$$p_\lambda(x;\gamma) = \begin{cases} \lambda|x| & \text{if } |x| \leq \lambda, \\ \frac{2\gamma\lambda|x| - x^2 - \lambda^2}{2(\gamma-1)} & \text{if } \lambda < |x| < \gamma\lambda \\ \frac{\lambda^2(\gamma+1)}{2} & \text{if } |x| \geq \gamma\lambda \end{cases}$$

and its derivative as

$$q_\lambda(x;\gamma) = \begin{cases} \lambda & \text{if } |x| \leq \lambda \\ \frac{\gamma\lambda - |x|}{\gamma - 1} & \text{if } \lambda < |x| < \gamma\lambda \\ 0 & \text{if } |x| \geq \gamma\lambda \end{cases}$$

The SCAD penalization approach is advantageous in that it keeps the penalty constant for large coefficients, reducing bias compared to LASSO. This approach is particularly effective in situations where the true underlying model has coefficients of varying magnitudes, as SCAD is less likely to shrink large coefficients to zero.

5.3 MCP

The Minimax Concave Penalty (MCP) is another non-convex penalty function that, like SCAD, addresses the limitations of LASSO by applying less shrinkage to larger coefficients. The MCP penalty is designed to decrease the penalty applied as the absolute value of the coefficient increases, leading to a model that encourages sparsity but retains large coefficients when necessary.

The penalization approach of MCP is particularly useful when the goal is to reduce the bias introduced by regularization methods like LASSO while still performing variable selection. MCP allows for greater flexibility by reducing the penalty as the coefficient magnitude grows, which can lead to more accurate estimation of large coefficients.

Definition

The MCP penalty is expressed as:

$$p_\lambda(x;\gamma) = \begin{cases} \lambda|x| - \frac{x^2}{2\gamma}, & \text{if } |x| \leq \gamma\lambda \\ \frac{1}{2}\gamma\lambda^2, & \text{if } |x| > \gamma\lambda \end{cases}$$

and its derivative as

$$q_\lambda(x;\gamma) = \begin{cases} \left(\lambda - \frac{|x|}{\gamma}\right)\operatorname{sign}(x), & \text{if } |x| \leq \gamma\lambda, \\ 0, & \text{if } |x| > \gamma\lambda \end{cases}$$

The MCP approach balances between the extreme sparsity encouraged by LASSO and the reduced shrinkage effect of Ridge regression. This makes MCP a flexible tool for scenarios where the true model has a mix of small and large coefficients, and precise estimation of larger coefficients is critical.

5.4 Minorization–maximization algorithm

In this part of thesis, we discuss how to solve equation (5.1). Since this equation does not have explicit form of the solution, we use iterative algorithms to solve it. We show the general approach that can implemented to any of penalizations approach described in the previous section. The solution consists of two steps - minorization–maximization algorithm and Newton–Raphson iterative method.

The Minorization-Maximization (MM) algorithm is an iterative method used to solve optimization problems by simplifying them into more manageable subproblems. Each iteration consists of two steps: minorization and maximization or minimization, depending on the goal.

5.4.1 Minorization Step

In the minorization step, a surrogate function $g(x|x^{(t)})$ is constructed, which satisfies the following conditions:

$$g(x|x^{(t)}) \leq f(x) \quad \text{for all } x,$$
$$g(x^{(t)}|x^{(t)}) = f(x^{(t)}),$$

where $f(x)$ is the target function to be maximized (or minimized), and $x^{(t)}$ is the current estimate of the variable of interest. The surrogate function g is easier to optimize and serves as a lower bound to f at all points except at $x^{(t)}$, where both functions are equal.

5.4.2 Maximization Step

In the maximization step, the surrogate function is optimized to provide a new estimate:

$$x^{(t+1)} = \arg\max_x g(x|x^{(t)}).$$

For minimization problems, this step involves finding the minimum of the surrogate function:

$$x^{(t+1)} = \arg\min_x g(x|x^{(t)}).$$

5.4.3 Iteration

The steps are repeated with the new estimate $x^{(t+1)}$ used to construct a new surrogate function in the subsequent minorization step. This process is iterated until convergence, which is typically when changes in the objective function or in the parameter estimates fall below a predetermined threshold, indicating that further iterations are unlikely to yield significant improvement.

5.4.4 Convergence Properties

Each iteration of the MM algorithm guarantees that the objective function will not decrease:

$$f(x^{(t+1)}) \geq f(x^{(t)})$$

for a maximization problem, ensuring stability and convergence towards a local maximum. For minimization, the inequality is reversed. This property makes the MM algorithm particularly appealing for problems where other optimization methods may struggle due to the complexity of the function or the presence of multiple local extrema.

Applying this approach to out problem we get

$$U^{\mathrm{p}}(\tilde{\alpha}) = U(\tilde{\alpha}) - \begin{pmatrix} q_{\lambda_{\tilde{\theta}}}(|\tilde{\theta}|)\,\mathrm{sgn}(\tilde{\theta})\frac{|\tilde{\theta}|}{\epsilon+|\tilde{\theta}|} \\ q_{\lambda_{\tilde{\beta}}}(|\tilde{\beta}|)\,\mathrm{sgn}(\tilde{\beta})\frac{|\tilde{\beta}|}{\epsilon+|\tilde{\beta}|} \end{pmatrix} = 0, \tag{5.2}$$

equation to solve, where ϵ is predefined small number (in application we use $1e^{-4}$).

The second part is an iterative search of solution. As described earlier, the NR method is based on the first degree derivatives of the objective function (gradient). in the case of $\boldsymbol{\theta}$ parameters, these will be derivatives of the functions described in Table 3.2 under GEE column and in the case of $\boldsymbol{\beta}$ parameters these are the gradient from OLS method applied to non-probability sample (S_A). In general it leads us to the following set of gradient functions for the first part of equation (5.1)

$$\begin{aligned} \nabla(\alpha) &= \frac{\partial U(\alpha)}{\partial \alpha^{\mathrm{T}}} = \mathrm{diag}\left\{\frac{\partial U_1(\theta)}{\partial \theta^{\mathrm{T}}}, \frac{\partial U_2(\beta)}{\partial \beta^{\mathrm{T}}}\right\}, \\ \frac{\partial U_1(\theta)}{\partial \theta^{\mathrm{T}}} &= -\frac{1}{N}\sum_{i=1}^{N} R_i^A \frac{1-\pi_A\left(\boldsymbol{x}_i^{\mathrm{T}}\theta\right)}{\pi_A\left(\boldsymbol{x}_i^{\mathrm{T}}\theta\right)}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}, \\ \frac{\partial U_2(\beta)}{\partial \beta^{\mathrm{T}}} &= -\frac{1}{N}\sum_{i=1}^{N} R_i^A m^{(1)}\left(\boldsymbol{x}_i^{\mathrm{T}}\beta\right)^2 \boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}, \end{aligned} \tag{5.3}$$

and similarly for the second part of 5.1

$$\Lambda(\alpha) = \begin{pmatrix} q_{\lambda_1}(|\alpha|) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & q_{\lambda_{2p}}(|\alpha 2p|) \end{pmatrix} \tag{5.4}$$

Let α start at an initial value $\tilde{\alpha}^{[0]}$. With the other co-ordinates fixed, the k th Newton–Raphson update for θ_j is

$$\tilde{\alpha}_j^{[k]} = \tilde{\alpha}_j^{[k-1]} + \left\{\nabla_{jj}\left(\tilde{\alpha}^{[k-1]}\right) + N\Lambda_{jj}\left(\tilde{\alpha}^{[k-1]}\right)\right\}^{-1}\left\{U_j\left(\tilde{\alpha}^{[k-1]}\right) - N\Lambda_{jj}\left(\tilde{\alpha}^{[k-1]}\right)\tilde{\alpha}_j^{[k-1]}\right\},$$

where $\nabla_{jj}(\alpha)$ and $\Lambda_{jj}(\alpha)$ are the j th diagonal elements in $\nabla(\alpha)$ and $\Lambda(\alpha)$ respectively. The procedure cycles through all the $2p$ elements of α and is repeated until convergence.

It is recommended to use K-fold cross validation for selectiing tuning parameters $(\lambda_\theta, \lambda_\beta)$ which minimize following loss functions for set of parameters $\boldsymbol{\alpha}$.

$$\text{Loss}\left(\lambda_\theta\right) = \sum_{j=1}^{p}\left(\sum_{i=1}^{N}\left[\frac{R_i^A}{\pi_i^A\left\{\boldsymbol{x}_i^{\mathrm{T}}\hat{\theta}\left(\lambda_\theta\right)\right\}} - \frac{R_i^B}{\pi_i^B}\right]\boldsymbol{x}_{i,j}\right)^2$$

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

where $\hat{\theta}\left(\lambda_\theta\right)$ and $\hat{\beta}\left(\lambda_\beta\right)$ are penalized estimators with tuning parameters $\lambda_\theta, \lambda_\beta$ for selection and outcome model respectively. For estimation we consider only the union of covariates $\boldsymbol{X}_C$, where $C = \hat{M}_\theta + \hat{M}_\beta$ and $\hat{M}_\theta = \left\{j : \hat{\theta}_j \neq 0\right\}$ and $\hat{M}_\beta = \left\{j : \hat{\beta}_j \neq 0\right\}$. In short, we estimate only on truly important variables for selection and outcome models.

Chapter 6

R nonprobsvy package

All of the methods described in this paper have been implemented in the R package `nonprobsvy`. In addition, it should be noted that there are several packages that allow the correction of selection bias in nonprobability samples, such as Marra and Radice (2023), Luis Castro Martín and del Mar Rueda (2020) or even Tillé and Matei (2021). 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. 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 et al. (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).

6.1 Functions and Features

The nonprobsvy package offers several functions to facilitate the integration of non-probability and probability samples. Below are the functions with brief descriptions:

Function	Description
cloglog_model_nonprobsvy	Estimates the model using a complementary log-log link function for weight adjustments.
confint.nonprobsvy	Computes confidence intervals for selection model coefficients.
controlInf	Constructs a list of control parameters for statistical inference.
controlOut	Sets control parameters for the outcome model.
controlSel	Establishes control parameters for the selection model.
genSimData	Generates simulated data according to the methodology described by Chen, Li, and Wu (2020).
logit_model_nonprobsvy	Estimates the model using a logit link function for weight adjustments.
pop.size	Estimates the size of the population based on the provided object from the nonprobsvy class.
probit_model_nonprobsvy	Estimates the model using a probit link function for weight adjustments.
summary.nonprobsvy	Generates summary statistics for models of the nonprobsvy class.
vcov.nonprobsvy	Provides an estimated covariance matrix for model coefficients.

Some of these functions such as logit_model_nonprobsvy, cloglog_model_nonprobsvy, probit_model_nonprobsvy, controlInf, controlOut, controlSel are internal and do not serve the end user. The rest operate on the object of the nonprob class defined by the nonprob function described in the following section.

6.2 Main Function: nonprob

The core function of the package is nonprob, which fits models for inference based on non-probability surveys. It integrates various methods for estimating population means, sums and means of covariates, and uses the functionality of the survey package when a probability sample is available. In addition, we can distinguish between three control functions: controlInf,

`controlSel` and `controlOut`, in which a number of optional arguments are available related to the choice of algorithm for a given estimation method, the method of calculation, the desire to select variables, bias minimization and much more. Let us look at the syntax of the function and the type of arguments that can be defined in it.

6.2.1 Usage

The main function has a number of arguments, some of which have a default value, some of which must be defined, and some of which are completely optional.

```
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,
  ...
)
```

Next, we will describe the control functions already mentioned in the package. The first of them concerns arguments generally related to statistical inference. Here, for example, we can decide what type of variance to calculate, how to replicate weights in the bootstrap for the MI estimator with the GLM method, whether we want to use a method minimizing bias, change the number of iterations in bootstrap algorithms, and more.

```
controlInf(
  vars_selection = FALSE,
  var_method = c("analytic", "bootstrap"),
  rep_type = c("auto", "JK1", "JKn", "BRR", "bootstrap", "
   subbootstrap", "mrbbootstrap",
    "Fay"),
  bias_inf = c("union", "div"),
  num_boot = 500,
  bias_correction = FALSE,
  alpha = 0.05,
  cores = 1,
  keep_boot,
  pmm_exact_se = FALSE,
  pi_ij
)
```

Separately, we define control arguments for IPW models, in particular the number of iterations for fitting a given method, optimization methods, weighting methods (MLE or GEE), the h function for the GEE method, the type of penalizing function in the case of variable selection, and related parameters (e.g., lambda), as well as several others.

```
controlSel(
  method = "glm.fit",
  epsilon = 1e-04,
  maxit = 500,
  trace = FALSE,
  optimizer = c("maxLik", "optim"),
  maxLik_method = "NR",
  optim_method = "BFGS",
  dependence = FALSE,
  key = NULL,
  est_method_sel = c("mle", "gee"),
  h = c(1, 2),
```

```
  penalty = c("SCAD", "lasso", "MCP"),
  a_SCAD = 3.7,
  a_MCP = 3,
  lambda = -1,
  lambda_min = 0.001,
  nlambda = 50,
  nfolds = 10,
  print_level = 0,
  start_type = c("glm", "naive", "zero")
)
```

The last control function is related to parameters for mass imputation methods. Some of these overlap with arguments for IPW, but additionally, we can choose the predictive mean matching method, the way of searching for nearest neighbors, the selection of k for the PMM method, which minimizes variance, and more.

```
controlOut(
  epsilon = 1e-04,
  maxit = 100,
  trace = FALSE,
  k = 1,
  penalty = c("SCAD", "lasso", "MCP"),
  a_SCAD = 3.7,
  a_MCP = 3,
  lambda_min = 0.001,
  nlambda = 100,
  nfolds = 10,
  treetype = "kd",
  searchtype = "standard",
  predictive_match = 1:2,
  pmm_weights = c("none", "prop_dist"),
  pmm_k_choice = c("none", "min_var"),
  pmm_reg_engine = c("glm", "loess")
)
```

6.2.2 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
data	Data frame with data from the non-probability sample.
selection	Formula for the selection (propensity) equation.
outcome	Formula for the outcome equation.
target	Formula with target variables.
svydesign	Optional svydesign object containing probability sample and design weights.
pop_totals	Optional named vector with population totals of the covariates.
pop_means	Optional named vector with population means of the covariates.
pop_size	Optional double with population size.
method_selection	Character string specifying the method for propensity score estimation (e.g., "logit").
method_outcome	Character string specifying the method for response variable estimation (e.g., "glm").
family_outcome	Character string describing the error distribution and link function to be used in the model (e.g., "gaussian").
subset	Optional vector specifying a subset of observations to be used in the fitting process.
strata	Optional vector specifying strata.
weights	Optional vector of prior weights to be used in the fitting process.
na_action	Function indicating what should happen when the data contain NAs.
control_selection	List indicating parameters to use in fitting selection model for propensity scores.
control_outcome	List indicating parameters to use in fitting model for outcome variable.
control_inference	List indicating parameters to use in inference based on probability and non-probability samples.
start_selection	Optional vector with starting values for the parameters of the selection equation.
start_outcome	Optional vector with starting values for the parameters of the outcome equation.
verbose	Logical value indicating if verbose output should be printed.

`x`	Logical value indicating whether to return the model matrix of covariates as part of the output.
`y`	Logical value indicating whether to return the vector of outcome variable as part of the output.
`se`	Logical value indicating whether to calculate and return the standard error of the estimated mean.
`...`	Additional optional arguments.

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

```
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

```
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

```
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.

```
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 $h(\boldsymbol{x}_i, \boldsymbol{\theta}) = \boldsymbol{x}_i$).

```
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 neighbors in the NN algorithm.

```
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.

```
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

```
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
```

The `summary` command displays more descriptive information about the analysis performed. In a single call, we get the most important information, such as the estimated mean, its standard deviation, sample and population sizes, parameter values and their properties, such as statistical significance tests. At the end of the call, we can also analyse the distribution of the trimmed weights (inverse of the propensity scores), the residuals from the model and the AIC or BIC values.

The package can be installed both from CRAN and from the github site (development

version), where a number of tutorials and documentation can be found. It is worth mentioning that during the first few months, the package has already been installed more than 1300 times (as of 18/09/2024).

Chapter 7

Simulation study

All the theoretical considerations require some justification also in the application of the described methods themselves. Therefore, we will carry out a series of simulations to analyse and compare the precision or efficiency of the estimators. We will focus on comparing three different approaches to mean estimation, namely: mass imputation, inverse probability weighting and doubly robust estimation. We will also compare different methods for obtaining the estimators, such as maximum likelihood estimation, k-nearest neighbour methods and predictive mean matching for mass imputation.

This chapter can be divided into 6 main parts:

1. Comparison of mass imputation, inverse probability weighted and doubly robust estimators.
2. Comparison of different mass imputation methods such as generalized linear models, k-nearest neighbors and predictive mean matching.
3. Comparison of different propensity score methods such as maximum likelihood estimation and calibration equations, including different linking functions.
4. Comparison of different doubly robust methods, including a combination of mass imputation and inverse probability methods and a bias minimization method.
5. Comparison of analytical and bootstrap methods of variance calculating.

In general, we will first present the simulation scheme, then the results and finally their analysis and conclusions.

7.1 Basic setup

For the six different simulations, we propose three variants for creating the population in which we will study the properties of the estimators. We will first present their details and origin, and then turn to the results themselves.

7.1.1 Simulation of the Inverse Probability Weighting Estimators

The setup for the simulation is copied from Yang and Kim (2020). For propensity score estimators we generate a finite population $F_N = \{(x_i, y_i) : i = 1, \ldots, N\}$ where $N = 10,000$. Here, y_i represents the outcome variable, which can be either continuous or binary, and x_i is a p-dimensional vector of covariates with $p = 50$. The first component of each x_i is set to 1, and the remaining components are generated independently from a standard normal distribution.

Two types of samples are drawn from this population:

- A non-probability sample S_A of size $n_A \approx 2000$, selected according to the selection indicator $R_i^A \sim \text{Bernoulli}(\pi_i^A)$.
- A probability sample S_B of average size $n_B = 500$, selected under Poisson sampling with selection probabilities proportional to $\pi_i^A \propto 0.25 + |X_{1i}| + 0.03|y_i|$.

For the non-probability sampling probability π_i^A, we consider two models:

1. Linear model (PSM I): $\text{logit}(\pi_i^A) = \alpha_0^T x_i$, where $\theta_0 = (-2, 1, 1, 1, 1, 0, \ldots, 0)^T$.
2. Non-linear model (PSM II): $\text{logit}(\pi_i^A) = 3.5 + \theta_0^T \log(x_{2i}) - \sin(x_{3i} + x_{4i}) - x_{5i} - x_{6i}$, where $\theta_0 = (0, 0, 0, 3, 3, 3, 3, 0, \ldots, 0)^T$.

We generate the outcome variable y_i using the following models:

1. Continuous outcome (OM I – y_{11}): $y_i = \beta_0^T x_i + \epsilon_i$, $\epsilon_i \sim N(0, 1)$, where $\beta_0 = (1, 0, 0, 1, 1, 1, 1, 0, \ldots, 0)^T$.
2. Non-linear continuous outcome (OM II – y_{12}): $y_i = 1 + \exp\{3 \sin\Big(\beta_0^T x_i\Big)\} + x_{5i} + x_{6i} + \epsilon_i$, $\epsilon_i \sim N(0, 1)$.
3. Binary outcome (OM III – y_{21}): $y \sim \text{Bernoulli}(\pi_y(x))$ with $\text{logit}(\pi_y(x)) = \beta_0^T x$.
4. Non-linear binary outcome (OM IV – y_{22}): $y \sim \text{Bernoulli}(\pi_y(x))$ with $\text{logit}(\pi_y(x)) = 2 - \log\Big((\beta_0^T x)^2\Big) + 2x_{5i} + 2x_{6i}$.

We conducted $r = 500$ replications and in assessing the quality of estimators, we considered the following metrics, which will also be used in subsequent simulations to evaluate the remaining estimators: $\text{Bias} = \bar{\hat{\mu}} - \mu$, $\text{SE} = \sqrt{\frac{\sum_{r=1}^{R}\left(\hat{\mu}^{(r)} - \bar{\hat{\mu}}\right)^2}{R-1}}$ and $\text{RMSE} = \sqrt{\text{Bias}^2 + \text{SE}^2}$, where $\bar{\hat{\mu}} = \sum_{r=1}^{R} \hat{\mu}^{(r)}/R$ and $\hat{\mu}^{(r)}$ is an estimate of the mean in the r-th replication. We will primarily focus on comparing the methods of maximum likelihood estimation (MLE) and calibration methods (GEE1 and GEE2). Additionally, we will evaluate the results depending on the applied link function (logit - table 7.1, probit - table 7.2, and cloglog - table 7.3).

For the logit link function, the results show that under PSM1, the MLE estimator exhibits significant bias across all variables, with the highest RMSE observed for the variable y_{12}. GEE1 performs better, showing lower bias and RMSE, particularly for variables y_{11} and y_{12}. GEE2, while similar to MLE in bias and RMSE, performs slightly better for y_{12}. Under PSM2, all estimators exhibit relatively low bias and RMSE, indicating improved performance. In this case, GEE1 consistently shows the lowest bias and RMSE across all variables.

Table 7.1: Simulated Bias, SE, RMSE (multiplied by 100) of main estimators - logit

Estimator	y11			y12			y21			y22		
	Bias	SE	RMSE	Bias	SE	RMSE	Bias	SE	RMSE	Bias	SE	RMSE
PSM 1												
IPW MLE	-0.769	0.928	1.21	-0.934	3.00	3.14	-0.122	0.232	0.262	0.0473	0.212	0.218
IPW GEE1	0.00703	0.113	0.113	0.0525	0.402	0.406	0.00169	0.0300	0.0300	0.00429	0.0265	0.0269
IPW GEE2	-0.769	0.928	1.21	-0.958	3.02	3.17	-0.121	0.233	0.262	0.0467	0.212	0.217
PSM 2												
IPW MLE	0.163	0.260	0.307	-1.24	0.234	1.26	-0.00621	0.0322	0.0328	-0.170	0.0391	0.175
IPW GEE1	-0.000920	0.0995	0.0995	-1.21	0.114	1.22	-0.0265	0.0130	0.0295	-0.0807	0.0139	0.0819
IPW GEE2	0.163	0.260	0.307	-1.24	0.234	1.26	-0.00621	0.0322	0.0328	-0.170	0.0391	0.175

For the probit link function, the results indicate that under PSM1, the MLE estimator again displays high bias and RMSE, particularly for y_{12}. GEE1 outperforms MLE, showing much lower bias and RMSE across all variables. GEE2's performance, while similar to MLE, is slightly worse in terms of RMSE. Under PSM2, the performance of all estimators improves, with GEE1 continuing to show the best performance, particularly for variables y_{12} and y_{22}.

Table 7.2: Simulated Bias, SE, RMSE (multiplied by 100) of main estimators - probit

Estimator	y11			y12			y21			y22		
	Bias	SE	RMSE	Bias	SE	RMSE	Bias	SE	RMSE	Bias	SE	RMSE
PSM 1												
IPW MLE	-1.13	1.83	2.15	-1.23	5.90	6.03	-0.161	0.428	0.457	0.0538	0.392	0.396
IPW GEE1	0.00169	0.116	0.116	0.0273	0.431	0.432	0.000555	0.0319	0.0319	0.00287	0.0285	0.0287
IPW GEE2	-1.25	1.68	2.09	-1.51	5.28	5.49	-0.175	0.404	0.440	0.0549	0.361	0.365
PSM 2												
IPW MLE	0.123	0.225	0.256	-1.23	0.203	1.25	-0.0108	0.0284	0.0304	-0.153	0.0356	0.157
IPW GEE1	-0.00581	0.100	0.100	-1.21	0.115	1.22	-0.0269	0.0132	0.0299	-0.0808	0.0139	0.0820
IPW GEE2	0.192	0.298	0.355	-1.23	0.273	1.26	-0.000942	0.0378	0.0378	-0.189	0.0509	0.196

For the cloglog link function, under PSM1, the MLE estimator continues to exhibit high bias and RMSE, especially for y_{12}. Both GEE1 and GEE2 perform better, with lower bias and RMSE, and GEE1 showing a slight edge in most cases. In PSM2, the performance of all estimators is generally better, with GEE1 consistently outperforming MLE and GEE2, especially for variables y_{12} and y_{22}.

Table 7.3: Simulated Bias, SE, RMSE (multiplied by 100) of main estimators - cloglog

Estimator	y11			y12			y21			y22		
	Bias	SE	RMSE	Bias	SE	RMSE	Bias	SE	RMSE	Bias	SE	RMSE
						PSM 1						
IPW MLE	0.453	0.125	0.470	0.343	0.251	0.425	0.0665	0.0215	0.0699	-0.0238	0.0251	0.0346
IPW GEE1	0.000992	0.112	0.112	0.0786	0.392	0.400	0.00130	0.0295	0.0295	0.00449	0.0262	0.0266
IPW GEE2	-0.497	0.751	0.901	-0.616	2.32	2.40	-0.0942	0.196	0.217	0.0332	0.166	0.170
						PSM 2						
IPW MLE	0.321	0.500	0.594	-1.20	0.450	1.28	0.0158	0.0649	0.0668	-0.276	0.0752	0.286
IPW GEE1	-0.00572	0.100	0.100	-1.21	0.114	1.22	-0.0270	0.0132	0.0301	-0.0809	0.0139	0.0821
IPW GEE2	0.142	0.247	0.285	-1.23	0.221	1.25	-0.00937	0.0312	0.0326	-0.163	0.0366	0.167

When comparing the three link functions (logit, probit, and cloglog), it is evident that no single method universally outperforms the others in all scenarios. However, the cloglog link function shows slightly better performance in terms of lower bias and RMSE for some variables, particularly under the more complex sampling model PSM2. This might suggest that cloglog is more robust in handling non-linear relationships in the data, especially when combined with the GEE1 estimator, which consistently performs best across all functions. The logit function, while generally effective, exhibits higher RMSE and bias in more complex scenarios, particularly when used with the MLE estimator. Probit tends to fall between logit and cloglog in terms of performance, showing moderate bias and RMSE across most scenarios.

Overall, the MLE estimator tends to have the highest bias and RMSE values across all models and link functions, indicating less reliable estimates compared to the other methods. GEE1 generally shows the best performance across all link functions and variables, with consistently lower bias and RMSE compared to MLE and GEE2. GEE2 performs similarly to MLE in many cases, but with slightly better bias and RMSE values. However, it does not match the performance of GEE1.

In conclusion, GEE1 consistently provides the most accurate estimates with the lowest bias and RMSE across all link functions (logit, probit, and cloglog). The MLE estimator generally performs the worst, particularly in more complex models (PSM1), where bias and RMSE are highest. GEE2, while an improvement over MLE, still does not match the performance of GEE1. There is no clear winner among the link functions as the performance varies depending on the variable and estimator in question. However, across all link functions, GEE1 emerges as the most reliable estimator, particularly when dealing with non-linear models or more complex sampling designs.

7.1.2 Simulation of Variance methods comparison and effectiveness of doubly robust estimators

For another package of simulation we use setup from Chen, Li, and Wu (2020). We consider a finite population of size $N = 20,000$. Each unit i in the population has a response variable y_i and a set of auxiliary variables x following the specified regression model:

$$y_i = 2 + x_{1i} + x_{2i} + x_{3i} + x_{4i} + \sigma\epsilon_i, \quad i = 1, 2, \ldots, N,$$

where the auxiliary variables are defined as:

$x_{1i} = z_{1i}$,
$x_{2i} = z_{2i} + 0.3x_{1i}$,
$x_{3i} = z_{3i} + 0.2(x_{1i} + x_{2i})$,
$x_{4i} = z_{4i} + 0.1(x_{1i} + x_{2i} + x_{3i})$,

and the distributions of the z variables are:

$z_{1i} \sim \text{Bernoulli}(0.5)$,
$z_{2i} \sim \text{Uniform}(0, 2)$,
$z_{3i} \sim \text{Exponential}(1)$,
$z_{4i} \sim \chi^2(4)$.

The error terms ϵ_i are iid as $N(0, 1)$. The standard deviation σ is adjusted to control the correlation coefficient ρ between y and the linear predictor $x\beta$ at values 0.3, 0.5, and 0.8. Accordingly, we have labeled the 3 explanatory variables - y30, y60 and y80.

Two distinct sampling strategies are employed:

- **Nonprobability Sample** S_A: Selected by Poisson sampling with inclusion probabilities π_i^A determined by the logistic regression model:

$$\log\left(\frac{\pi_i^A}{1 - \pi_{A_i}}\right) = \theta_0 + 0.1x_{1i} + 0.2x_{2i} + 0.1x_{3i} + 0.2x_{4i},$$

  where θ_0 is set such that $\sum_{i=1}^{N} \pi_{A_i} = n_A$, matching the target sample size n_A.

- **Probability Sample** S_B: Taken by randomized systematic PPS (Probability Proportional to Size) sampling with inclusion probabilities π_i^B proportional to $z_i = c + x_{3i}$. The constant c is chosen to ensure the variation of survey weights meets the condition $\max z_i / \min z_i = 50$.

For the response variable $y30$, which corresponds to a low correlation ($\rho = 0.3$) between the response and the predictor, the DR MLE and DR GEE1 estimators demonstrate the smallest bias and RMSE, indicating a strong performance in this scenario. The DR MLE exhibits a bias of 0.0129, standard error of 0.417, and RMSE of 0.418, closely followed by DR GEE1, which shows slightly lower bias and RMSE values. DR PMM and DR NN, while still relatively accurate, show higher bias and RMSE, suggesting that predictive mean matching and nearest neighbor methods might introduce additional variability in cases of low correlation.

As the correlation increases to a moderate level ($\rho = 0.5$) with the response variable $y60$, the performance of DR MLE and DR GEE1 remains strong, with both methods showing minimal bias and nearly identical RMSE values (0.218 and 0.217, respectively). DR PMM and DR NN continue to exhibit higher bias and RMSE, indicating that these methods may be less robust when the correlation between the predictor and response increases. In particular, DR NN shows a higher bias (0.0190) and RMSE (0.302), suggesting that the nearest neighbor imputation might be more sensitive to changes in correlation.

For the highly correlated scenario ($\rho = 0.8$) represented by the response variable $y80$, DR MLE and DR GEE1 again outperform the other methods, with DR GEE1 showing virtually no bias (-0.0000375) and the lowest RMSE (0.174). DR PMM and DR NN display higher bias and RMSE, with DR PMM showing the highest bias (-0.0504) and RMSE (0.704) among all methods. These results suggest that, in the presence of a strong correlation, the performance of predictive mean matching and nearest neighbor imputation deteriorates compared to regression-based methods.

Table 7.4: Simulated Bias, SE, RMSE (multiplied by 100) of Various DR Estimators

Estimator	y30			y60			y80		
	Bias	SE	RMSE	Bias	SE	RMSE	Bias	SE	RMSE
DR MLE	0.0129	0.417	0.418	0.00310	0.217	0.218	0.000118	0.174	0.174
DR GEE1	0.0123	0.416	0.417	0.00282	0.217	0.217	-0.0000375	0.174	0.174
DR PMM	0.0364	0.702	0.703	0.0128	0.312	0.313	-0.0504	0.702	0.704
DR NN	0.0470	0.651	0.653	0.0190	0.301	0.302	-0.0398	0.651	0.653

Overall, DR MLE and DR GEE1 consistently demonstrate the best performance across all scenarios, with low bias and RMSE, making them the most reliable estimators in this study. The results indicate that regression-based multiple imputation methods paired with standard IPW or GEE1 for IPW offer robust and accurate estimates, even as the correlation between the response and predictor increases. In contrast, DR PMM and DR NN, which use predictive mean matching and nearest neighbor imputation, respectively, tend to introduce more variability, particularly in scenarios with higher correlation. Therefore, for doubly robust estimation in finite populations, DR MLE and DR GEE1 are recommended, especially when dealing with moderate to high correlations between predictors and the response variable.

Table 7.5: Mean Values of Analytical and Bootstrap Variances

Estimator	y30		y60		y80	
	Analytical	Bootstrap	Analytical	Bootstrap	Analytical	Bootstrap
DR MLE	0.416	0.222	0.432	0.171	0.216	0.174
IPW MLE	0.493	0.298	0.483	0.302	0.332	0.263
DR GEE	0.412	0.221	0.429	0.171	0.215	0.173
IPW GEE1	0.457	0.220	0.428	0.250	0.284	0.173
DR GEE2	0.416	0.222	0.433	0.171	0.216	0.174
IPW GEE2	0.493	0.298	0.484	0.302	0.332	0.263
MI GLM	0.418	0.218	0.421	0.172	0.217	0.172
MI NN	0.509	0.336	0.758	0.183	0.247	0.223
MI PMM	0.509	0.302	0.653	0.183	0.247	0.210

Table 7.5 compares the mean values of analytical and bootstrap variances across different estimators for the variables $y30$, $y60$, and $y80$. Overall, bootstrap variances tend to be lower than analytical variances, suggesting that bootstrap methods produce more conservative estimates of variability, which can lead to narrower confidence intervals and potentially higher coverage rates.

For the DR MLE and IPW MLE estimators, the analytical variances consistently exceed the bootstrap variances. For instance, the analytical variance for $y80$ with IPW MLE is 0.332, while the bootstrap variance is 0.263. This pattern indicates that bootstrap methods might offer more accurate reflections of true variability, which could explain their better coverage performance.

The GEE-based estimators (DR GEE, IPW GEE1, and DR GEE2) show minimal differences between analytical and bootstrap variances, indicating robustness in both variance estimation methods. This consistency suggests that either method could be reliably used with GEE estimators.

MI-based estimators exhibit more pronounced differences, particularly with MI NN, where the analytical variance for $y30$ is 0.509 compared to 0.336 for bootstrap. This significant reduction in variance with bootstrap methods may enhance coverage reliability but also highlights potential underestimation risks with analytical variance.

Table 7.6: Coverage Rates for Different Variance Estimators

Estimator	y30		y60		y80	
	Analytical	Bootstrap	Analytical	Bootstrap	Analytical	Bootstrap
DR MLE	0.964	0.966	0.956	0.960	0.948	0.950
DR GEE1	0.960	0.968	0.954	0.956	0.946	0.946
DR GEE2	0.964	0.970	0.956	0.958	0.948	0.948
IPW MLE	0.974	0.966	0.974	0.958	0.982	0.950
IPW GEE1	0.974	0.966	0.994	0.952	0.998	0.944
IPW GEE2	0.974	0.968	0.974	0.964	0.982	0.954
MI GLM	0.972	0.974	0.958	0.960	0.950	0.952
MI NN	0.876	0.986	0.898	0.982	0.916	0.972
MI PMM	0.878	0.970	0.868	0.954	0.914	0.962

Table 7.6 shows results for coverage rates of set of variance estimators. Starting with the DR MLE estimator, we observe that the coverage rates are generally consistent across both analytical and bootstrap methods, with slight deviations. For instance, in the case of $y30$, the coverage rate is 0.964 for the analytical variance and slightly higher at 0.966 for the bootstrap variance, suggesting that both methods provide reliable coverage, though bootstrap may offer a marginal improvement in this context.

The DR GEE1 and DR GEE2 estimators show similar trends, with coverage rates that are closely aligned between the analytical and bootstrap approaches. However, there is a slight dip in the coverage rate for $y80$ using both methods, particularly for DR GEE1 (0.946), which indicates a potential limitation in maintaining coverage at higher levels of correlation. Despite this, the bootstrap approach consistently matches or slightly exceeds the analytical method, suggesting a slight advantage of bootstrap in these scenarios.

In comparison, the IPW MLE and IPW GEE1 estimators demonstrate notably high coverage rates, particularly for the analytical method, where IPW GEE1 achieves nearly perfect coverage (0.998) for $y80$. The bootstrap estimates, while slightly lower, still provide robust coverage, confirming the effectiveness of the IPW method in achieving reliable confidence intervals, especially in higher correlation scenarios.

Looking at the MI-based estimators, MI GLM and MI NN generally perform well, with bootstrap variance estimation providing slightly better coverage than the analytical method across most variables. For instance, MI NN achieves a high coverage rate of 0.986 with bootstrap for $y30$, significantly outperforming its analytical counterpart (0.876), indicating that bootstrap variance estimation might be particularly beneficial for MI NN in terms of improving coverage reliability.

In contrast, the MI PMM estimator exhibits the lowest coverage rates among the methods analyzed, particularly when using analytical variance estimation. The coverage for $y30$ is notably low at 0.878, but it improves significantly with bootstrap variance estimation to 0.970, suggesting that MI PMM may suffer from under-coverage when relying on analytical variance

estimates. This underlines the importance of using bootstrap variance estimation for MI PMM to achieve more reliable confidence intervals.

7.1.3 Comparison of all estimators and mass imputation methods

For the last setup we generate a finite population of size $N = 10^5$ according to Kim et al. (2021). Each individual in the population is associated with three random variables x_1, x_2, and x_3, which are generated independently from a normal distribution $N(2, 1)$. The error term ϵ follows $N(0, 1)$.

The population features three response variables defined by the following models:

1. $y_{1i} = 1 + 2x_{1i} + \epsilon_i$,
2. $y_{2i} = -1 + x_{1i} + x_{2i} + x_{3i} + \epsilon_i$,
3. $y_{3i} = -10 + x_1^2 + x_2^2 + x_3^2 + \epsilon_i$,

where the indices correspond to different regression models used to generate data.

The samples are drawn using simple random sampling without replacement (SRSWOR) with two distinct strategies:

- **Non-probability Sample** S_A: This sample is drawn from the population divided into two strata defined by $x_i \leq 2$ and $x_i > 2$. The sizes of the strata samples are set as $n_{strata1} = 0.7n_A$ and $n_{strata2} = 0.3n_A$, with n_A taking values 500 and 1,000.
- **Probability Sample** S_B: This sample also employs SRSWOR with a fixed size $n_B = 500$.

Two results are reported in this section. The first one for mass imputation estimators analysis and all of them inclusing IPW, MI and DR methods.

In 7.7 table we report MI simulation with Monte Carlo bias (Bias), standard error (SE), relative mean square error (RMSE) and empirical coverage rate (CR) of 95% confidence intervals using the analytical variance estimator based on $R = 500$ simulations for each Y variables.

Table 7.7: Simulated Bias, SE, RMSE (multiplied by 100) and CI of 5 estimators

Estimator	Y_1				Y_2				Y_3			
	Bias	SE	RMSE	CR	Bias	SE	RMSE	CR	Bias	SE	RMSE	CR
					$n_A = 500$							
Naive	-64.08	4.86	64.26	–	-31.48	5.98	32.04	–	-127.15	20.92	128.86	–
					correctly specified							
GLM	0.30	9.90	9.90	94.20	0.18	8.64	8.64	94.60	0.11	33.36	33.36	94.00
NN1	0.21	11.07	11.07	94.60	-1.32	9.74	9.83	94.60	-19.86	32.09	37.74	90.40
NN5	0.18	10.24	10.24	94.40	-2.48	8.30	8.66	94.80	-38.09	30.76	48.96	74.20
PMM1A	0.21	11.07	11.07	94.80	0.09	10.09	10.09	95.80	-0.80	33.99	34.00	94.40
PMM1B	0.28	9.89	9.90	95.00	0.16	8.64	8.64	94.20	-0.64	33.33	33.34	94.00
PMM5A	0.18	10.24	10.24	94.60	-0.11	8.92	8.92	95.00	-2.06	33.31	33.37	93.80
PMM5B	0.19	9.86	9.86	94.60	0.10	8.63	8.63	94.40	-2.07	33.12	33.18	94.00
					mis-specified							
GLM	–	–	–	–	–	–	–	–	-10.83	28.72	30.69	93.20
NN1	–	–	–	–	–	–	–	–	-5.58	37.13	37.55	94.20
NN5	–	–	–	–	–	–	–	–	-14.14	31.90	34.89	91.20
PMM1A	–	–	–	–	–	–	–	–	-1.25	38.05	38.07	97.40
PMM1B	–	–	–	–	–	–	–	–	-10.41	28.66	30.49	93.80
PMM5A	–	–	–	–	–	–	–	–	-3.99	31.19	31.45	96.60
PMM5B	–	–	–	–	–	–	–	–	-10.21	28.64	30.40	93.80
					$n_A = 1,000$							
Naive	-64.48	7.09	64.87		-32.14	8.53	33.25	–	-129.02	30.86	132.66	–
					correctly specified							
GLM	0.05	10.46	10.46	95.80	-0.10	9.40	9.40	95.00	-0.18	33.68	33.68	94.40
NN1	-0.12	11.77	11.77	94.40	-2.13	10.26	10.48	94.80	-29.91	32.43	44.12	83.80
NN5	-0.41	10.69	10.70	95.60	-3.93	9.04	9.86	91.20	-54.85	30.26	62.65	55.80
PMM1A	-0.12	11.77	11.77	94.60	-0.22	10.96	10.96	94.60	-1.48	34.15	34.18	94.60
PMM1B	-0.01	10.45	10.45	95.20	-0.15	9.40	9.40	95.20	-1.61	33.49	33.53	94.40
PMM5A	-0.41	10.69	10.70	96.00	-0.45	9.64	9.65	94.80	-4.11	33.09	33.35	94.20
PMM5B	-0.25	10.40	10.40	95.40	-0.26	9.37	9.37	95.00	-4.36	33.12	33.41	94.60
					mis-specified							
GLM	–	–	–	–	–	–	–	–	-10.98	33.06	34.83	92.60
NN1	–	–	–	–	–	–	–	–	-10.36	39.93	41.25	95.60
NN5	–	–	–	–	–	–	–	–	-23.61	34.04	41.43	88.60
PMM1A	–	–	–	–	–	–	–	–	-4.32	42.73	42.95	96.60
PMM1B	–	–	–	–	–	–	–	–	-10.39	33.01	34.60	93.00
PMM5A	–	–	–	–	–	–	–	–	-7.23	35.85	36.58	94.40
PMM5B	–	–	–	–	–	–	–	–	-10.16	32.94	34.47	92.80

For the second variable Y_2, we notice a pattern similar to that of Y_1, with one key difference. As the number of nearest neighbours increases, the bias for the NN estimator also increases. In contrast, for the PMM B estimator, changing the number of nearest neighbours does not significantly affect the bias or standard error. The empirical coverage rates for all estimators remain close to the nominal 95%.

When examining the last variable Y_3 (non-linear), significant differences become apparent. With correctly specified transformations and linear regression, the bias is minimal for the GLM and PMM estimators, but not for the NN estimator. The PMM A estimator, using $\hat{y} - \hat{y}$ matching, and the GLM exhibit the lowest bias and a coverage rate near the nominal 95%.

This observation aligns with Remark 4 regarding the robustness of $\hat{y} - \hat{y}$ matching.

As expected, when the model is mis-specified (rows under *mis-specified*), i.e., using linear regression with only x_1 and x_2, all estimators exhibit some bias. However, the PMM A estimator shows the least bias. Interestingly, increasing k for the NN estimator leads to higher bias. On the other hand, the PMM B estimators are characterized by the lowest variance and consequently the lowest RMSE (along with PMM5A). All estimators maintain a coverage rate close to the nominal 95%.

Table 7.8: Simulated Bias, SE, RMSE (multiplied by 100) of main estimators

Estimator	Y_1			Y_2			Y_3			Y_3 (mis-specified)		
	Bias	SE	RMSE	Bias	SE	RMSE	Bias	SE	RMSE	Bias	SE	RMSE
Sample Size = 500												
Naive	-64.08	4.86	64.26	-31.48	5.98	32.04	-127.15	20.92	128.86	-124.00	29.30	128.00
GLM	0.30	9.90	9.90	0.18	8.64	8.64	-0.20	32.30	32.30	-11.70	33.00	35.00
IPW	-3.54	12.80	13.30	-1.42	10.50	10.60	53.50	56.50	77.80	-3.40	41.90	42.10
DR	-1.39	10.40	10.50	-0.63	9.04	9.07	-2.13	32.20	32.30	-0.60	35.50	35.50
Sample Size = 1000												
Naive	-64.48	7.09	64.87	-32.14	8.53	33.25	-129.02	30.86	132.66	-128.00	20.40	130.00
GLM	0.05	10.46	10.46	-0.10	9.40	9.40	-0.20	32.10	32.20	-13.90	28.70	31.90
IPW	-4.02	11.70	12.40	-1.70	9.35	9.50	52.20	51.00	73.00	-7.28	35.30	36.10
DR	-1.33	9.84	9.93	-0.57	8.47	8.49	-1.97	32.20	32.30	-3.02	31.00	31.20

The simulation results presented in Table 7.8 provide a clear comparison between the Naive, GLM (Generalized Linear Model), IPW (Inverse Probability Weighting), and DR (Doubly Robust) estimators across three response variables y_1, y_2, and y_3, with sample sizes of 500 and 1000. The Naive estimator, which simply calculates the mean from the non-probability sample, consistently exhibits substantial bias and high RMSE across all scenarios, indicating its inadequacy for reliable inference in this context. This is particularly evident with y_1, where the Naive estimator's bias is -64.08 and RMSE is 64.26, highlighting its significant deviation from the true population parameters.

In contrast, the GLM estimator significantly improves upon these results, achieving minimal bias and lower RMSE, especially when the model is correctly specified. For instance, with y_1, the GLM estimator reduces the bias to 0.30 and RMSE to 9.90. This pattern of performance suggests that GLM is quite effective, particularly in scenarios where the underlying model assumptions are met. However, while the GLM estimator is robust in correctly specified models, its performance, though still superior to Naive, shows some limitations when compared to more flexible approaches.

The IPW estimator, while better than the Naive approach, shows greater sensitivity to the complexity of the response models, particularly for y_3, where it struggles to maintain low bias and RMSE. For y_3 under a sample size of 500, the IPW estimator results in a bias of 53.50 and an RMSE of 77.80, indicating that IPW may not fully account for the intricacies of the data,

particularly in non-linear settings.

The DR estimator stands out as the most reliable method across all conditions, offering consistently low bias and RMSE, even in the presence of model mis-specification. For example, in the case of y_3, which is non-linear and more complex, the DR estimator outperforms the others with an RMSE of 32.30, matching the performance of GLM in correctly specified scenarios but demonstrating superior resilience to mis-specification. Moreover, even when the sample size increases to 1000, the DR estimator maintains its robustness, achieving an RMSE of 9.93 for y_1, thus confirming its effectiveness in both small and large samples.

Overall, the results suggest that while GLM is highly effective in simple, well-specified models, its utility diminishes in more complex settings where mis-specification risk is higher. On the other hand, the DR estimator proves to be the most versatile and reliable, particularly in scenarios where the model assumptions may not hold perfectly. This makes the DR approach highly recommended for statistical inference in non-probability samples, particularly when dealing with complex data structures and potential model mis-specification.

Chapter 8

Empirical study

In this chapter, 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). Example observation of the survey look as follows:

id_unit	**sector**	**class**	**nace**	**region**	**weight**
a9cc990df6a99ab215a1bc13f51d4825c7d52d18	0	D	O	14	1
c9dbaf50890165ebe810aa770de0e9df903dc35b	0	D	O	24	6
718e0bba42bcec6ed98f9690db6d26cb7b93c880	0	D	R.S	14	1
532a1879a692b9d7bbb7282ba757d028156ef341	0	D	R.S	14	1
0b6b623fa45e257284a3049d097af322841337e3	0	D	R.S	22	1
c855a825e80866c00c7513721d5fcb38929f3cd6	0	S	R.S	26	1

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.

id_unit	**sector**	**class**	**nace**	**region**	**single_shift**
7fddce081cbb1dd5da072e1683e9bfd20acab593	0	D	P	30	FALSE
3d40ca689a4dca7d774981dc7db408301bf7f192	0	D	O	14	TRUE
2b5a57b0c2f03c2b252e559bcd77d52789d33d9c	0	D	O	04	TRUE
f2b18f5ef4386e70206d64810b5d3b7e0654918b	0	D	O	24	TRUE
bf17263f8fa3a9ff12d7ff60c10c4e426aeeb36e	0	D	O	04	TRUE
10fe847b7d19284e9e310105c5acfaf8f1ccbc37	1	D	C	28	FALSE

Finally the following variables are considered in the current study

- **id_unit** – unique identifier of the unit,
- **sector** – 0=public, 1=private,
- **nace** – NACE sections (from C to S; combined sections: D and E, K and L, R and S),
- **region** – region,
- **weight (JVS only)** – final weight determined in the Labour Demand Survey,
- **single_shift (CBOP only)** – whether the entity employs people on a single shift (based on the variable shiftCode).

As described in the work, the aim of this study will be to integrate these two datasets in order to carry out a correction of the percentage of single-shift vacancies. According to data description we consider JVS sample as probability (S_B) and CBOP as non-probability (S_A) one. The percentage of entities employing one shift based on CBOP data without correction is 66.05%.

Firstly, we will try to improve this percentage, and secondly to compare the results depending on the estimator used and its variation. We will also show the 95% confidence intervals of the results obtained, thus measuring the differences in the estimated variances of the estimators.

In the study, we decided to compare all the estimators with each other, along with numerous methods for their estimation. For the inverse probability weighting estimator, we compared the GLM and GEE methods, denoted in the table as IPW MLE and IPW CAL (from calibration) respectively. We have also added variable selection using the SCAD method and bootstrap estimation. Similarly, we calculated results using mass imputation methods, where we compared

nearest neighbour (NN), predictive mean matching $\hat{y} - \hat{y}$ (PMM1) and $\hat{y} - y$ (PMM2), and generalized linear models (GLM) methods. For doubly robust methods, we focused on comparing the standard method combining MLE for IPW and GLM for MI with the bias minimization (MIN) estimator. Results can be found in table 8.3 and figure 8.1. We show estimated mean, its standard errors and corresponding confidence intervals.

	Mean	Standard Error	Lower Bound	Upper Bound
IPW (ST)	0.708	0.011	0.687	0.729
IPW (CAL)	0.704	0.012	0.681	0.727
IPW (BOOT)	0.708	0.010	0.688	0.728
IPW (SCAD)	0.685	0.010	0.666	0.705
MI (LM)	0.704	0.011	0.681	0.726
MI (GLM)	0.703	0.011	0.681	0.725
MI (NN)	0.680	0.016	0.649	0.711
MI (PMM1)	0.697	0.015	0.668	0.726
MI (PMM2)	0.859	0.017	0.827	0.892
MI (GLM, SCAD)	0.703	0.011	0.681	0.725
DR (IPW)	0.703	0.011	0.681	0.726
DR (IPW, CAL)	0.704	0.011	0.682	0.726
DR (IPW, BOOT)	0.703	0.012	0.681	0.726
DR (SCAD)	0.703	0.011	0.681	0.725
DR (SCAD, MIN)	0.704	0.011	0.682	0.726

Table 8.3: Results

As can be seen the mean values of estimated percentage range from approximately 0.680 to 0.859, with the highest mean value observed for PMM2 estimator and the lowest for NN. The standard errors range from 0.01 to 0.017 with IPW bootstrap estimator having the smallest value of 0.01 and NN having the highest one, what indicates its lower precision. This also implies the narrowest and widest confidence intervals for these estimators, respectively. In general, the propensity scores methods (likelihood, calibrated – GEE) show similar mean values around 0.704 to 0.708, with relatively low standard errors, suggesting consistency and precision. The mass imputation methods (GLM, NN, PMM) are of lower values for estimated mean and exhibit more variation, particularly for PMM2 estimator. The reason is that this method is mainly suitable for continuous variables as the matching is done through Y values with predicted values what does not fit well with binary case on our study. The doubly robust methods (joint randomization, bias minimization) present consistent mean values range from 0.703 to 0.704 with low standard error values as similar to inverse probability weighted methods. We can also

observe that variables selection using SCAD method reduces the variance of estimators.

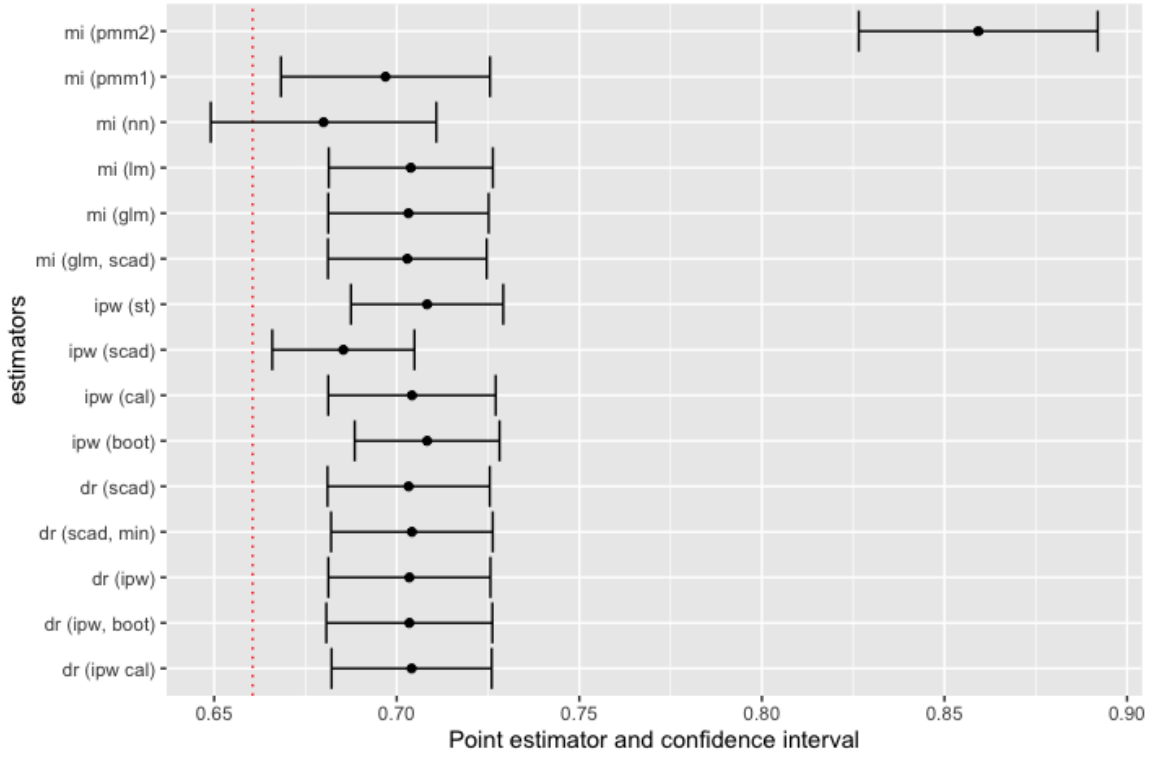

Figure 8.1: Research Results

In summary, the propensity score and doubly robust methods appear to be more consistent and precise for our study, while mass imputation estimators provide greater variability with NN and PMM2 shwoing the most deviation. General conclusion is that providing information from integration of probability and non-probability samples increased the estimated percentage of single shifts in job offers about 5% of average.

Summary

In this work we explored the integration of population data, whether from probability samples or vectors of known population means, with non–probability samples to improve statistical inference. We developed several estimators and proposed different methods for variance estimation, all supported by rigorous mathematical proofs and simulation studies.

The renewed interest in non–probability sampling stems from its cost–effectiveness and the challenges associated with traditional probability sampling, such as declining response rates and increasing operational costs. Our contribution to this evolving field is to extend established estimation techniques and to develop new methods that address these issues. Specifically, we have extended the inverse probability weighting (IPW) method by incorporating additional link functions such as probit and complementary log–log. We also derived the necessary mathematical components such as the log-likelihood function, gradient, Hessian and Jacobian for these models, thereby extending the range of applicable IPW estimators.

We also introduced advanced variable selection techniques, including the Least Absolute Shrinkage and Selection Operator (LASSO) and the Minimax Concave Penalty (MCP), for use in both probability and non-probability sample integration. This enables more accurate modelling in situations where a large number of covariates are available. In addition, we have extended the scope of estimation to cases where population–level information is only available in the form of vector totals or means, thereby extending the practical utility of these methods in scenarios with limited data.

In terms of data integration, we presented a new approach to mass imputation, specifically Predictive Mean Matching (PMM), and derived its associated variance estimation. This complements our previous work on mass imputation (MI) methods, such as generalized linear models (GLM) and nearest neighbour (NN) methods, making the comparison between different approaches more comprehensive.

Finally, the implementation of all these methods in the R package `nonprobsvy`, which is publicly available on CRAN, further enhances their accessibility to applied researchers. This package provides tools for mass imputation, IPW, doubly robust estimation, bias minimization and variable selection, making it a versatile resource for dealing with non-probability sample integration problems. The package was also mentioned in paper Ballerini et al. (2024) describing

innovative methods for survey data analysis.

Our contributions are timely given the ongoing discussions on the limitations of non-probability sampling and the lack of a unified framework. By expanding the toolbox of methods available for integrating auxiliary information from probability samples, we provide researchers with more robust options for drawing valid inferences from non-probability samples, thereby addressing a critical gap in current methodologies.

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.

Bibliography

Ballerini, V., Beraldo, D., Bocci, C., Braito, L., Milana, R., & Trans, M. (2024). *Report on mapping, harmonising and integrating novel data sources for research purposes*. University of Florence. Florence.

Bates, D., Maechler, M., et al. (2023). *Matrix: Sparse and dense matrix classes and methods* [R package version 1.6-1.1]. https://CRAN.R-project.org/package=Matrix

Beaumont, J.-F. (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.

Breheny, P., & Huang, J. (2023). *Ncvreg: Regularization paths for scad, mcp, and elastic net* [R package version 3.14-0]. 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.

Chrostowski, Ł., Beręsewicz, M., & Chlebicki, P. (2023, December). *Ncn-foreigners/nonprobsvy: Initial release* (Version 0.1.0). Zenodo. https://doi.org/10.5281/zenodo.10280114

Citro, C. F. (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]. https://CRAN.R-project.org/package=Rcpp

Elliott, M. R., & Valliant, R. (2017). Inference for Nonprobability Samples. *Statistical Science*, *32*(2). https://doi.org/10.1214/16-STS598

Epskamp, S. (2023). *Mathjaxr: Using mathjax in rd files for dynamic rendering of equations* [R package version 1.6-0]. https://CRAN.R-project.org/package=mathjaxr

Folashade Daniel Hong Ooi, R. C., & Weston, S. (2023). *Foreach: Provides foreach looping construct for r* [R package version 1.5.2]. https://CRAN.R-project.org/package=foreach

Groemping, U. (2023). *Nleqslv: Solve systems of nonlinear equations* [R package version 3.3.3]. 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]. https://CRAN.R-project.org/package=maxLik

Kim, J. K., 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. https://doi.org/10.1111/rssa.12696

Luis Castro Martín, R. F. G., & del Mar Rueda, M. (2020). *Nonprobest: Estimation in non-probability sampling.*

Lumley, T. (2004). Survey r package.

Marra, G., & Radice, R. (2023). *Gjrm: Generalised joint regression modelling.*

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/

Ripley, B., Venables, W., et al. (2023). *Mass: Support functions and datasets for venables and ripley's mass* [R package version 7.3-60]. https://CRAN.R-project.org/package=MASS

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

Steve Weston, S. W., Folashade Daniel, & Tenenbaum, D. (2022). *Doparallel: Foreach parallel adaptor for the 'parallel' package* [R package version 1.0.17]. https://CRAN.R-project.org/package=doParallel

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

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

Wu, C., & Thompson, M. E. (2020). *Sampling theory and practice*. Springer.

Yang, S., & Kim, J. K. (2020). Asymptotic theory and inference of predictive mean matching imputation using a superpopulation model framework. *Scandinavian Journal of Statistics*, *47*(3), 839–861. https://doi.org/10.1111/sjos.12429

Yang, S., Kim, J. K., & 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. https://doi.org/10.1111/rssb.12354

Yang, S., Kim, J.-K., & 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.