Poznań University
of Economics
and Business

Kyrylo Mordan

Implementacja metody empirical likelihood dla informatywnego braku odpowiedzi w pakiecie statystycznym R

Implementation of the empirical likelihood method for non-ignorable non-response in the R statistical package

Master's thesis

Thesis Supervisor: dr Maciej Beręsewicz, prof. UEP

Field of study: Informatics and Econometrics

Poznań 2022

Contents

Abstract

The goal of the thesis was to apply a method for correcting non-ignorable non-response based on Empirical Likelihood, implemented from scratch in R, as a part of a package and tested on a simulated data to show its effectiveness. Chapter one goes though the definitions and a little introduction into the problem of this particular type of missing data. Chapter two, in detail, presents Empirical Likelihood based method and its implementation in up-and-coming package to deal with a problem of non-ignorable non-response, called nmar. It shows what the package is all about, its core features and potential use cases. Chapter three describes results that we were able to produce with our R implementation of Empirical Likelihood based method on simulated data. Our efforts did produce a package, capable to correct non-ignorable non-response problem, with a use of Empirical Likelihood method, as results of a simulation show. The simulation study, that we conducted, was able to uncover obvious limitations of the method and give us a glimpse of its properties.

Introduction

In today's world, decision making process becomes more and more reliant on inferences drawn from the data. That data may come in different shapes and forms, collected in different manner from numerous sources, but what unites it all is a problem of missing data.

Missing data is a complex issue that can have many potential caused, be of various severity and can be dealt with in different ways to prevent potential degradation in the quality of the inference. There is one particular type of missing data that will be addressed here extensively, called non-ignorable non-response. As the name suggests it cannot be simply ignored, because by doing that, estimates would be prone to biases.

In this thesis, we present a method of correcting non-ignorable non-response developed by Jing Qin et al. (2002) and based on empirical likelihood method, studied by Owen (1988), Owen (1990), and Owen (2001). We also present a statistical package called *nmar*[1] that we created with a funding from National Science Centre[2]. The package, that we developed, implements the method from scratch in R, with a range of other features, geared towards both easy of use and options to choose from, when needed. We compare its performance to a baseline method, which ignores missing data to show effectiveness of the method, we implemented.

Our main goal is to show the `nmar` package that we have developed, with usage examples. Our secondary goal is to test the empirical likelihood based method with a use of a simulation study. The hypothesis is that the method will be able to outperform the baseline approach, which is to ignore the missing data. The assumption is that, if it is able to outperform the baseline method, like this kind of a method should, it would make it a correct implementation of the method we present in the chapter 2 of the thesis.

In the first chapter, there is a brief introduction into the problem of non-ignorable non-response and ways of dealing with it. In the following chapter, the method for correcting non-ignorable non-response, based on Empirical likelihood is show in detail. Third chapter of the thesis presents the `nmar` package, its main features and usage examples, since not everything in

[1] The name of the package is taken from the name of missing data mechanism called *not missing at random*.

[2] This work was supported by the National Science Center grant Towards census-like statistics for foreign-born populations – quality, data integration and estimation (NCN OPUS 22 2020/39/B/HS4/00941)

the implementation process was clear and straight forward, and some design choices had to be made.

The final chapter, tests both the method and its implementation with a simple simulation study, to make sure that we get good enough results with our implementation. Different dispersion metrics are used to get a grasp of what can be expected from the method performance-wise, and as a secondary goal, to show that our implementation returns results that make sens.

Chapter 1

The problem of non-ignorable non-response

1.1 Data sources and their problems in context of decision making

1.1.1 Sources of data

Data collected for statistical purposes is called primary data, while other relevant data, that was collected for some other reasons, such as administrative or business, is called secondary data. This makes primary data more relevant and its value therefore higher to the secondary data if the goal is the statistical inference. This can be explained by the greater adherence to the goals of the research of the primary data, when data collection process could be modified with them in mind. Also the lack of any prior statistical treatments, that can undermined its quality for the research that it was not intended for, make primary data more desirable (Sobczyk 2007).

When processing the data, distinction between two types of statistical analysis can be made: descriptive and inferential (Sobczyk 2007). Descriptive statistics that include basic measures such as mean, median or frequencies, can be used to summarize the data for better comprehension of the raw data. For the decision making process, any inference drawn from the descriptive statistics would be limited to the particular data set and could not be extrapolated beyond it. Inferential statistics, on the other hand, does allow for extrapolation and gives an ability to draw more general conclusions from the data about objects that it does not explicitly describe. For that to be possible the data has to be a representative sample from the population, for which such inferences could be made. There also should be an estimator with a certain set of properties, able to calculate descriptive statistic for the select population based on the representative sample from it.

The kind of data, for which inferential statistics could be calculated is called statistical data. Data for which only descriptive statistics could be calculated is a non-statistical data.

Non-statistical data is basically any data, which original purpose was not to draw conclusions from it over greater population with a use of probability sampling theory. It also includes data that does not quite fall into the categories of primary or secondary data, and is know as tertiary data (Buelens et al. 2012). Where primary and secondary data are terms that usually exist in the roam of data collected for statistics and administration, tertiary data is some kind of log data, a by-product of some process, usually generated rather then collected. Sizes of the tertiary data usually also greater, which is why it is commonly referred to as ’big data’.

So called ’big data’ has by itself, many definitions. Some would call ’big data’ anything that would not load into MS Excel (a little over million rows). There are also more formal definitions like 3V that stand for volume, velocity and variety or 5V that expands it with value and veracity.

"Big data" is not the only kind of non-statistical data source. Administrative records and registers are more traditional sources of non-statistical data, that can be a source of auxiliary variables for model-based approaches, even though their original purpose could be different from statistics (Beręsewicz 2016).

There are also numerous other classifications for the data sources out there, like the one proposed by Citro (2014), where "big data" is represented by two categories: *Commercial transaction records* and *Interactions of individuals with the World Wide Web*. This classification distinguishes the more traditional "big data" source that comes from the financial sector with the Web 2.0 consequent large volumes of data, generated by user interactions.

1.1.2 Modes of inference and estimators

It is worth noting, that inference for the non-statistical data beyond objects that it explicitly describe, could be made, with greater difficulties. The inference is possible in four different modes, according to Brakel and J. Bethlehem (2008):

- design-based,
- model-assisted,
- model-based,
- algorithm-based.

Design-based estimation is possible only for statistical data, since it relies on the use of sampling weights d_i, that are the inverse of probability of inclusion (π_i). The weights of the object in the sample data should sum up to known or estimated population totals. Design-based estimator of some hypothetical parameter of interest θ use weights and possess certain qualities like:

- Sufficiency – property of the estimator to give as much information about θ from the sample as possible, and there is no estimator of θ from the same sample that would be able to provide any additional information about it
- Efficiency – property of the estimator to have as little variance as possible
- Consistency – property of the estimator to be closer to true parameter θ with increase in number of observations
- Asymptotic unbiasedness – property of the estimator to a have smaller bias with more observations, so that when number of observations goes to infinity, bias of the estimator goes to zero

Model-assisted estimation like design-based estimation is possible only in the presence of statistical data, but unlike design-based estimation, it can also take advantage of the non-statistical data sources like register data. It is basically an improved version of the design-based inference, where survey weights not only constructed from the design weights, but also from the auxiliary information.

Model-based estimation is a predictive approach to inference. The idea is to fit the model on the available data, by estimating its parameters, in a way that unknown values could be predicted by it. The data that feeds the model does not necessary need to representative of the population in a way statistical data should be. The auxiliary information should be correlated enough with the target variable, for the estimates to be any good. This paves way for the uses of non-statistical data sources. The trick is to train the model on the data, which variability would cover the whole population, even though a sample itself does not need to representative of it, and be able to explain its variability in a sufficient manner.

Algorithmic estimation like model-based estimation is a predictive approach to inference. It makes use of non-parametric method as well as method that are generally referred to as being 'black box'. Algorithmic methods cannot be expressed by the mathematical models in any straight-forward fashion, which makes them different from the ones used for model-based estimation. Algorithmic approaches can also be more 'data hungry', because of their hyper-parameters that need to be tuned. Algorithmic estimation is even further departure from the traditional methods of statistical inference, since the need for data quantity and disregard for the sample representativeness, makes 'big data' non-statistical data more suitable for the task.

1.2 Missing data

1.2.1 Problem of missing data

There is rarely a primary source of data that does not suffer from some kind of a missing data. It plagues not only data that comes from statistical data sources like censuses and various types of surveys, but also all kinds of non-statistical data that comes in the form of "big data", among others. But where with some "big data" sources, quantity trumps quality, it creates additional challenges for the sampled data, since there is always a risk to overlook some vital piece of information, when generalizing results over the whole population. Whatever the source of the data is, even for "big data" sources it does pose a challenge in a form of a potentially non-ignorable non-response problem, that has to be addressed one way or the other.

It is also worth mentioning that occurrence of missing data in the dataset does not always mean the complete lack of information. In some cases, missing data can be very informative, if represented in a certain way. In a rectangular data, some variables may not have a logical reason to occur at the same time, but would have to be filled with values either way, whether it is coded as NA, some negative number or any other missing data code, example of which can be seen in McMullen (2001).

1.2.2 Types of missing data

Missing data may come in different shapes and forms. Each data source has its own specification, that reflects its origins, collection method, types of variables and so on. Survey data, for example, has a complex list of possible errors that could result in a missing data, as can be seen in figure 1.1.

Sampling errors like, for example, a sampling frame, that does not cover the whole target population, result in a missing data. Missing data in that case would be the part of the population not covered by the sampling frame. Similar thing may occur in the realm of non-sampling errors, where parts of the data could be lost or simply not taken into account during processing, which would make it as if they were missing.

If these kinds of missing data would have separate codes from the regular non-response, the data could be transformed, for the purposes of estimation to mitigate at least slightly the problem of missing data. For example, date variables, that represent an occurrence of the event, that haven't happened yet, would have missing values, that are not classified as non-response. When estimating mean number of days from the event to the current date, zeros could be assumed, where that kind of missing value was. The estimates would not be technically biased,

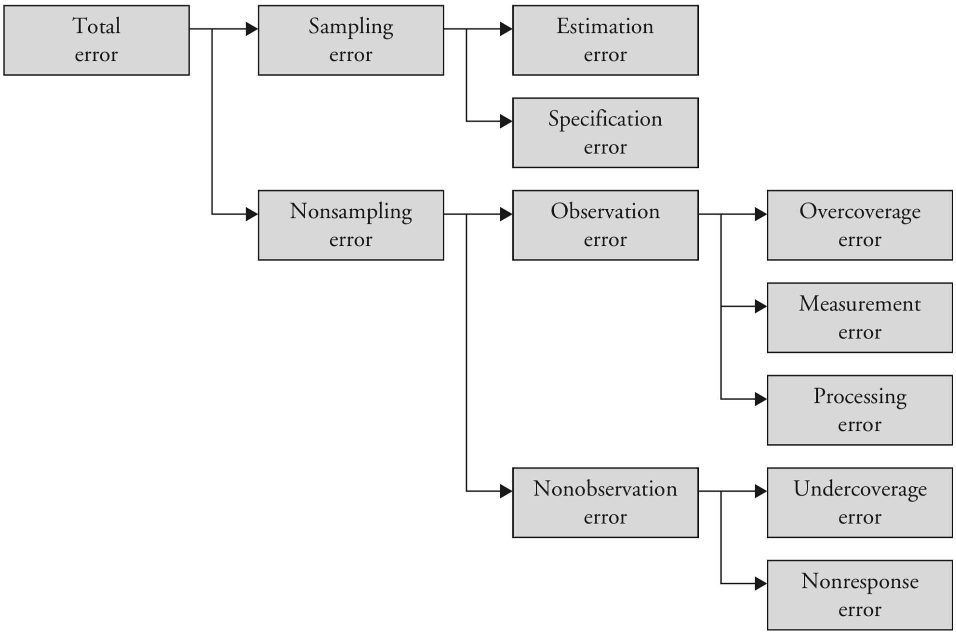

Figure 1.1. A classification of survey errors

Source : Based on Biffignandi and J. G. Bethlehem (2011).

since even if we knew the date of the future event that is missing, since it is happening after the current date, the number of days between them would still be zero. Figure 1.2 illustrates this example.

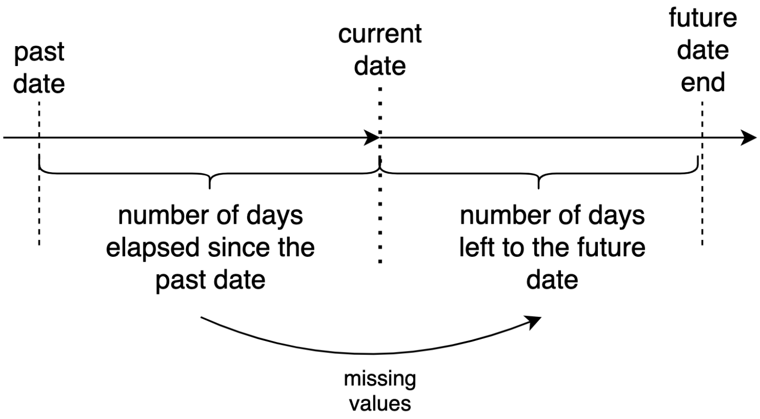
past date
current date
future date end
number of days elapsed since the past date
number of days left to the future date
missing values

Figure 1.2. Example of missing data that is not a problem

Source : Own elaboration

In broader terms, errors mentioned above could be classified to the avoidable type, since changes to the design of the survey for example, or a way of representing our data would resolve the missing values problem, at least to the extent, that it is possible. Non-response error however, could not be so easily avoided. Non-response error is caused by the lack of response that is not simply a result of resolving a logical contradictions to represent data in a rectangular format, since the answer exists, we just were not given access to it. There is obviously a component of the non-response, that could be addressed to some extent in a survey design that would improve response rates. But by no means, such method could guarantee its complete and outer elimination, without a use of drastic measures, that can rarely be enforced, if at all in the more common data sources.

1.3 Non-response missing data type

In general, non-response could be divided into the *unit non-response* and *item non-response* as noted by Yan and Curtin (2010) and those, that came before them. Unit could be understood as an observation, a respondent, a sampled individual or simply a row of data in the rectangular data. By item we understand a particular data point for the unit, like a value of a particular variable or a column in a tidy data set.

Unit non-response represent complete lack of information provided by the selected unit, while item non-response refers to a situation, when some only some information was not provided by the sampled unit. In the case of item non-response, missing values would occur in one of the patterns, shown in the figure 1.3.

In short, these patterns could be understood in the following manner:

- (A) Univariate pattern – when missing values are concentrated in one of the variables.
- (B) Multivariate – when missing values are not exclusive to one variable, but instead are a part of multiple variable with a same level of missingness. Could be a result of joining two set, where one had unit non-response problem.
- (C) Monotone pattern – when missing values occur in multiple variables and are related in a way that missing values for one variable, always means missing values for another one. Real world example cause for this pattern, could be an early drop out from a survey.
- (D) General pattern – when missing data appears throughout the dataset in a way that there is no clear pattern.
- (E) Underidentified pattern – similar to the general pattern, apart from having no units, for which no missing values would occur for all of the variable at the same time. The data in such form can be considered to not be suitable for estimation.
- (F) Latent Variable Pattern – when one of the variables is missing for the entire sample.

- (G) Planned missingness – when missing data is intentional for a large proportion of the respondents.
- (H) File matching – a similar pattern to the underidentified in a way that estimation becomes challenging when it occurs. It can be identified when there are two sets variables that are never observed together.

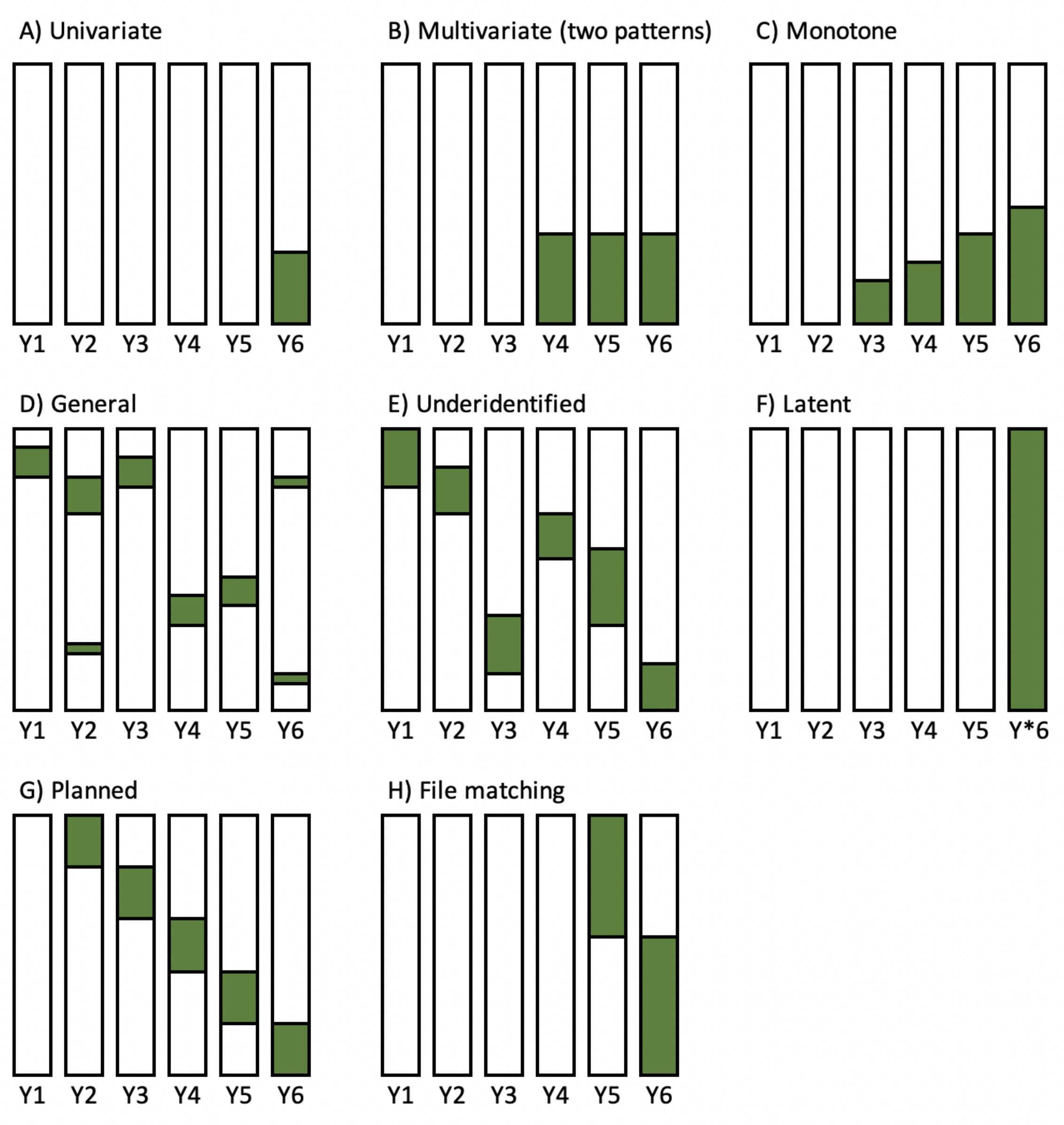

Figure 1.3. Missing data patterns

Source : Based on Enders (2022) and Little and Rubin (1987).

These patterns can divided into two groups: connected and unconnected as mentioned by Buuren (2018). Connected patterns can be identified by the occurrence of at least one unit, for which all values are available. They are needed, so that estimation of unknown parameters in a model, where variables in a pattern are explanatory, would work. Univariate, multivariate, monotone and general are such patterns. Other patterns mentioned earlier are not, at least without reducing a number of variables until the pattern becomes connected. This is easy with planned pattern, since its variables are not meant to be used as explanatory at the same time for parameter estimation, but not so much for the underidentified pattern. This is a case, because when the pattern for explanatory variables is unconnected, the response becomes depended on the variables, which themselves are subject to non-response, which makes data with underidentified patter insufficient for estimation.

1.3.1 Problems caused by non-response

Our interest is in the situation, when assumption about random nature of the missing data cannot be made. This means that ignoring the missing data for a variable of interest during estimation of its mean or any other statistic of that sort would have negative consequences.

According to Lundström and Särndal (1999), Manski (2016), Peytchev (2013), Szymkowiak (2019), and Toepoel and Schonlau (2017), non-response results in:

- increase in variance due to reduction of effective sample or population size.
- corruption of distribution shapes of the affected variable, which not only affects final results but also estimation of a feature, dependant on the affected variable.
- appearance of systematic difference in estimated value and the true one, a.k.a. bias.

Non-response bias is a deviation of the estimated value from the true value. Non-response itself causes a bias, if for some parts of the population, non-response happens more often then for the others. When estimation is attempted, ignoring or insufficiently addressing the problem of missing data may result in bias, since some of parts of the population become underrepresented

.

We understand non-ignorable non-response as a situation when a probability of response is conditioned on both auxiliaries and the target variable and could not be explained by the auxiliaries only, as shown in the equation 1.1.

$$P(R=1|\mathbf{X},Y) \neq P(R=1|\mathbf{X}) \tag{1.1}$$

For the unbiased estimator $\hat{Y}_1$, it could be shown that it is equal to the population statistic $\overline{Y}$, like in 1.2. But if the estimator is biased, like the estimator $\hat{Y}_2$ in 1.3, then bias 1.4 could be defined as a difference between the real value and its estimator.

$$E(\hat{Y}_1) = \overline{Y}, \tag{1.2}$$

$$E(\hat{Y}_2) \neq \overline{Y}, \tag{1.3}$$

$$\text{Bias}(\hat{Y}_2) = E(\hat{Y}_2) - \overline{Y}. \tag{1.4}$$

1.3.2 Response mechanisms

Presence of non-response does not mean a presents of non-response bias, if non-response itself is random. Depending on a relationship between response, target variable subject to non-response and auxiliary variables, the problem may be addressed rather easily, with some effort or painfully and unreliably. Those relationships can be summarised by the response mechanism.

There are three response mechanism applicable in the case of missing data, that are very common in the literature on the topic of non-response, originally introduced by Little and Rubin (1987) in their current form. These are

- Missing Completely at Random (MCAR) – response is dependent neither on the target variable nor on the explanatory variables and the response itself is not considered to be selective.
- Missing at Random (MAR) – response is not dependant on the target variable but is dependant on the auxiliaries, which are explanatory variables for the target. This makes response selective.
- Not Missing at Random (NMAR or MNAR) – response or rather lack thereof is dependant on the auxiliaries that do not suffer from the non-response themselves or their condition could be addressed though weighting, like in the case of MAR, but unlike MAR is also dependant on the target variable, subject to non-response.

The assumptions about the response mechanism are important, to select an appropriate method of dealing with the problem, so that bias could be avoided. MCAR is basicly implying to ignore the missing values, since not relationship between lack response and the data at hand was found. In practice though, lack of observed relationship and random missingness is not the same, which does make the assumption unrealistic, purely on the basis of the data at hand. MAR makes more sense, since it tries explain the response with additional information, which may or may not be sufficient. But available information may not be sufficient to model response and

NMAR would have to be assumed. Under NMAR response is basicaly dependant on information that we do not have, which are missing values from the target variable. The model depended on an assumption which cannot be verified in a straight forward way, but rather though simulations.

Unfortunately, determining which assumption can be made is very problematic. As discussed by the Buuren (2018), test to determine where a response mechanism is MCAR or MAR exist, but their practical value is unclear and they are not widely used.

1.4 Possible causes of non-response

Non-response is a complex topic that can have many causes, depending on a data source, collection method and the population. In appear to be a growing concern and may be worsening with time as Vandenplas et al. (2018) argues on the basis of European social survey. Reasons like socioeconomic changes in society, innability to participate in a survey and privacy concerns are attributed to the decline.

There are studies that are trying to determine the reasons for the non-response in health surveys, like a study of non-response causes in a Finish nation-wide health survey by Korkeila et al. (2001) or a study of non-response bias in a Dutch national survey on adolescent health by Cheung et al. (2017). They do show difference in sociodemographic and behaviour factor between respondents and non-response, the questionnaire itself asking for a sensitive information and voluntary nature of such surveys having an impact on a non-response bias that results from those differences.

The decline in response rates, could be seen as a common pool resource problem as Leeper (2019) proposed, where the repondends or rather their ability and motivation to give response is not abundant which makes it prone to overextraction by the reaserchers that are interested in the similar topic and target the same population.

1.5 Approaches to dealing with non-response

Depending on the cause of the non-response, an appropriate method can be selected with a goal of elimination of non-response bias. These methods typicaly fall into one of the following categories, described by Szymkowiak (2019):

- **Preventive techniques** are techniques of intervening at the stage of designing a survey or during the data collection process to make a participation more appealing or even selecting a sampling frame with non-response avoidance in mind.

- **Reductive techniques** are similar to the preventive techniques, but they are more on the "damage control" side of things. The focus is to incentify units that did not respond or are assumed to have a low probability of response with money, reminders, repeat phone or email contact and even replacing them with a unit with similar characteristics.
- **Corrective techniques** are the methods of dealing with non-response post data collection stage. Those are statistical methods of adjustment, which goal is to minimise a potential bias resulting from incomplete data, using available data and making various assumptions.

There are many more methods that are geared toward dealing with non-response, but not all of them claim the ability of dealing with non-ignorable non-response. This is closely related to the response mechanism assumtions, which were presented ealier.

The simplest thing do to is to get rid the missing values and make estimation under the assumption of MCAR. This kind of an approach may be applied, when missing data is a very small percentage of the recorded units. Generally, the strategy cannot be relied on to produce unbiased results especially, when trying to calculate estimates for within groups in the population.

More complicated approaches are either imputing missing values or using additional information about the units during estimation process in a hope of minimising information loss, that non-response introduces. They can be also divided into four categories, by the nature of their core principal: Weighting techniques, Imputation, Model-based approaches and some of their combinations. Little and Rubin (1987)

Weighting techniques make use of the fact that known probability of selection is an inverse of the design weights. The Horvitz-Thompson estimator of the mean:

$$\hat{Y}_{\text{HT}} = \frac{1}{N}\sum_{i=1}^{N} \pi_i^{-1} y_i, \tag{1.5}$$

where N is a size of the population, π_i is the probability of selection for the unit i and y_i is target variable, for which estimator is to be calculated, lies at the core of the weigting technique. The idea is to reweight the design weights in a way, that non-response can be compensated for and bias would be avoided.

The imputation can be simple, like deductive imputation or mean imputation. This class of methods usually is more focused on avoiding sample reduction more then correcting potential bias. Although in select cases, when the quality of the data is suboptimal and the so additional information, in a form of a common sense or even expert knowledge is applicable, the method can be very powerful.

Model based approaches assume some kind of the model, that should explain relationship between non-response, target variable and additional information in a way, that helps to recover unbiased population characteristics. The underlying assumptions that model caries, if correctly specified, makes the approach most suitable for the case of non-ignorable non-response. The reason for this is an ability to not only correct bias but also avoid creating it, when there is none to begin with, and the MCAR response mechanism could have been assumed.

Imputation can also be model based, where a linear regression or some kind of a model is used in a machine learning fashion, to impute missing values with model predictions. Imputation can be attempted on multiple values at the same time like with Multiple Imputation by Chained Equations method van Buuren and Groothuis-Oudshoorn (2011), which unlike simple imputation is claimed to be able to deal with MAR.

Corrective techniques that use additional information to compensate for the item or even unit non-response. Those methods use models parametric and non parametric, weights, population means and whatever other information can be included about the units that have missing values, during estimation process. The empirical likelihood based method, discussed in this chapter is one of those methods, that possess elements of both model-based and weighting approaches.

1.6 Summary

There are various data sources out there. For the purposes of the researcher, some types of data are more valuable then others. There are two types of statistical analysis that can be done on any given data source, in the context of fueling a data-driven decision making process. Descriptive statistics does not allow to make conclusions about the population outside of the sample, while with inferential statistics, extrapolation is possible. Statistical data sources are more suited for inferential statistics, although it is possible to make inferences with non-statistical data with model-based and algorithmic modes of inference. Missing data messes up the ability to make inferences, but statistical data is more susceptible to the problem. There are many reasons for the occurrence of the missing data and types of the missing data. Our interest lies in the the problem of non-ignorable non-response. This type of missing data can lead to biased estimates if not addressed. There three types of approaches when dealing with non-response: preventive techniques, reductive techniques and correcting techniques. Preventive and reductive techniques focus and the data collection process, while corrective technique are employing various statistical approaches to mitigate the problem of missing data.

Chapter 2

Empirical likelihood method for non-ignorable non-response

2.1 Basic settings and notation

Table 2.1 presents two situations when we have or do not have information about auxiliary variables X or sampling weights d for non-respondents. In the first case (denoted *Case I*) sampling weight d and auxiliary X is available for all units in the sample. This situation occurs when information about sampled units is taken from administrative data. The *Case II* assumes that we do not have information about non-respondents because we either did not have administrative data for these units (e.g. we sample flats not persons) or non-respondents refuse to participate and no information about their characteristics is recorded.

Table 2.1. Settings – missing data general case 1

Units	X	Y	d
Case I			
Respondents	yes	yes	yes
Non-respondents	yes	no	yes
Population means	yes	no	–
Case II			
Respondents	yes	yes	yes
Non-respondents	no	no	no
Population means	yes	yes	–

Source: own elaboration.

In the first case, we use whole sample size n, while in the second we can only use a subset of respondents of size m. This means that the efficiency of estimation based only on m is lower than when the whole sample size is used. Moreover, when we have background information about non-respondents then we may simultaneously correct for non-response and use X to impute Y thus resulting with more efficient estimation (lower variance).

Table 2.2 provides basic notation for the study.

Table 2.2. Basic notation for the study

Symbol	Description
N	population size
n	sample size
m	number of respondents
Y	target variable
y	target variable for respondents
y_i	target variable for the i^{th} unit
X	matrix of auxiliary variables
x_i	vector of auxiliary variables for the i^{th} unit
μ_X	vector of known population averages
R_i	response indicator taking $\{0,1\}$ values
θ	vector of assumed response model parameters
$w(y,X,\theta)$	assumed response model (i.e. $P(R=1\|Y,\mathbf{X},\theta)$)
W	response rate $P(R=1)$
$F(y_i,x_i)$	join distribution of (Y,X)
$dF(y_i,x_i)$	change in the join distribution
$L(\theta,dF)$	likelihood function
$l(\theta,W)$	log likelihood function
$\lambda_1\&\lambda_2$	Lagrange multipliers
ψ^*	vector of real parameters
ψ	vector of parameter estimates
$f(\psi)$	first derivatives of the log likelihood function
$J(\psi)$	jacobian matrix, second derivatives of the log likelihood function
$\overline{Y}$	population mean of the target variable
$\hat{\overline{Y}}$	estimated mean of the target variable

Source: own elaboration.

2.2 Empirical likelihood based approach

The empirical likelihood method described in this section is a semi-parametric approach. This means that it does not make assumptions about underlying distribution of the target variable, but does so for the missing data model. The method is based on empirical likelihood (EL) developed by Owen (1988), Owen (1990), and Owen (2001).

EL was adapted for non-ignorable non-response by Jing Qin et al. (2002) and is a further development of their work on general modeling approach, which allows that partial missing observations of the target variable also depend on the target variable itself. It should be able to consistently estimate the expected value of a non-ignorable missing random variable using auxiliary covariate variables. It is done by the use of a model which hybrid likelihood is composed of an empirical likelihood and a parametric one.

We start with the likelihood function given by equation (2.2)

$$L(\theta, dF) = \prod_{i=1}^{m} w(y_i, x_i, \theta) dF(y_i, x_i) \times \prod_{i=m+1}^{n} \int\int (1 - w(y_i, x_i, \theta)) dF(y_i, x_i),$$

where the first part of the equation is calculated for the respondents and the second part for the non-respondents, where $F(y_i, x_i)$ is a join distribution of (Y, X), $w(y_i, x_i, \theta)$ is probability of response given target Y and auxilairy variables X and we assume that $w(y_i, x_i, \theta)$ is a parametric response model (e.g. logistic regression).

Equation (2.2) can be rewritten as

$$\left\{ \prod_{i=1}^{m} \frac{w(y_i, x_i, \theta) dF(y_i, x_i)}{W} \right\} W^m (1 - W)^{n-m}, \tag{2.1}$$

where $W = P(R = 1) = \int\int w(y_i, x_i, \theta) dF(y, x)$ is the unconditional response rate. The first term in (2.1) is the likelihood conditioning on $R = 1$, and the term $W^m (1 - W)^{n-m}$ is the binomial likelihood of R.

If population means μ_x are available then we can maximise the semi-parametric likelihood (2.1) subject to constraints

$$p_i \geq 0, \quad \sum_{i=1}^{m} p_i = 1, \quad \sum_{i=1}^{m} w(y_i, x_i, \theta) - W = 0,$$

and

$$\sum_{i=1}^{n} p_i (x_i - \mu_x) = 0,$$

where p_i is the jump of the F at (y_i, x_i). By introducing Lagrange multipliers and profiling for all the values of p_i, we obtain

$$p_i = dF(y_i, x_i) = \frac{1}{m[1 + \lambda_1 (x_i - \mu_x) + \lambda_2 (w(y_i, x_i, \theta) - W)]}, \tag{2.2}$$

where λ_1 is a vector of Lagrange multipliers connected with auxiliary information constraint (i.e. difference between x_i and its known population means μ_x) and λ_1 is a scalar connected with unconditional response rate W. Thus, the constrained log-likelihood function is given by (2.3).

$$l(\theta,W)=\sum_{i=1}^{m}\ln w(y_i,x_i,\theta)+(n-m)\ln(1-W)-m\ln m-\sum_{i=1}^{m}\ln\Big(1+\lambda_1^T(x_i-\mu_x)+\lambda_2(w(y_i,x_i,\theta)-W)\Big) \tag{2.3}$$

Neither W, nor parameters θ of assumed parametric response model $w(y_i,x_i,\theta)$ are known, which in addition to λ_1, λ_2 makes them a vector of parameters that needs be estimated with maximum likelihood.

Given the parameter vector $\psi^* = (\lambda_1^T, \lambda_2, \theta^T, W)$, and setting partial derivatives of (2.3) with respect to ψ^* to zero, results in the following system of non-linear equations

$$f^*(\psi^*)=\begin{pmatrix}\mathrm{l}^{\lambda_1}\\ \mathrm{l}^{\lambda_2}\\ \mathrm{l}^{\theta}\\ \mathrm{l}^{W}\end{pmatrix}=0,$$

where

$$\begin{aligned}
&\mathrm{l}^{\lambda_1} := -\textstyle\sum_{i=1}^{m}\frac{x_i-\mu_x}{z_i},\\
&\mathrm{l}^{\lambda_2} := -\textstyle\sum_{i=1}^{m}\frac{x_i-\mu_x}{z_i},\\
&\mathrm{l}^{\theta} := \textstyle\sum_{i=1}^{m}\left[\frac{\partial\ln w(y_i,x_i,\theta)}{\partial\theta}-\frac{\lambda_2\partial w(y_i,x_i,\theta)/\partial\theta}{z_i}\right],\\
&\mathrm{l}^{W} := \frac{m/n-1}{1-W}-\lambda_2,
\end{aligned}$$

with

$$z_i=[1+\lambda_1^T(x_i-\mu_x)+\lambda_2(w(y_i,x_i,\theta)-W)],\quad (i=1,...,m),$$

and where $\mathrm{l}^W = 0$ is equivalent to $\lambda_2=\frac{n/m-1}{1-W}$. Therefore it is not necessary to explicitly set λ_2 to be estimated, since this parameter is clearly defined by W. This reduces the parameter vector to $\psi=(\lambda_1^T,\theta^T,W)$ and the system of equations to be solved is

$$f(\psi)=\begin{pmatrix}\mathrm{l}^{\lambda_1}\\ \mathrm{l}^{\lambda_2}\\ \mathrm{l}^{\theta}\end{pmatrix}=0. \tag{2.4}$$

Second derivatives form a Jacobian matrix of f:

$$J(\psi) = \begin{pmatrix} 1^{\lambda_1\lambda_1} & 1^{\lambda_1\theta} & 1^{\lambda_1 W} \\ 1^{\lambda_2\lambda_1} & 1^{\lambda_2\theta} & 1^{\lambda_2 W} \\ 1^{\theta\lambda_1} & 1^{\theta\theta} & 1^{\theta W} \end{pmatrix}, \tag{2.5}$$

with

$$\begin{aligned}
&1^{\lambda_1\lambda_1} := \frac{\partial 1^{\lambda_1}}{\partial \lambda_1^T} = \textstyle\sum_{i=1}^n \frac{(x_i-\mu_x)(x_i-\mu_x)^T}{z_i^2},\\
&1^{\lambda_2\lambda_1} := \frac{\partial 1^{\lambda_2}}{\partial \lambda_1} = \textstyle\sum_{i=1}^n \frac{(x_i-\mu_x)^T (w(y_i,x_i,\theta))}{z_i^2},\\
&1^{\lambda_1\theta} := \frac{\partial 1^{\lambda_1}}{\partial \theta^T} = \textstyle\sum_{i=1}^n \frac{\lambda_2 (x_i-\mu_x)\cdot \partial w(y_i,x_i,\theta)/\ \partial\theta^T}{z_i^2} = \left(1^{\theta\lambda_1}\right)^T,\\
&1^{\lambda_1 W} := \frac{\partial 1^{\lambda_1}}{\partial W} = -\textstyle\sum_{i=1}^n \frac{\lambda_2 (x_i-\mu_x)}{z_i^2},\\
&1^{\lambda_2\theta} := \frac{\partial 1^{\lambda_2}}{\partial \theta^T} = -\textstyle\sum_{i=1}^n \frac{\frac{\partial w(y_i,x_i,\theta)}{\partial\theta}}{z_i} + \textstyle\sum_{i=1}^n \frac{\lambda_2 \frac{\partial w(y_i,x_i,\theta)}{\partial\theta}(w(y_i,x_i,\theta)-W)}{z_i^2},\\
&1^{\lambda_2 W} := \frac{\partial 1^{\lambda_2}}{\partial W} = \textstyle\sum_{i=1}^n \frac{1}{z_i} - \textstyle\sum_{i=1}^n \frac{\lambda_2 (w(y_i,x_i,\theta)-W)}{z_i^2},\\
&1^{\theta\theta} := \frac{\partial 1^{\theta}}{\partial \theta^T} = \textstyle\sum_{i=1}^n \frac{\partial^2 \ln w(y_i,x_i,\theta)}{\partial\theta\partial\theta^T} - \textstyle\sum_{i=1}^n \frac{\lambda_2 \frac{\partial^2 w(y_i,x_i,\theta)}{\partial\theta\partial\theta^T}}{z_i} + \textstyle\sum_{i=1}^n \frac{\left[\lambda_2 \frac{\partial w(y_i,x_i,\theta)}{\partial\theta}\right]\left[\lambda_2 \frac{\partial w(y_i,x_i,\theta)}{\partial\theta^T}\right]}{z_i^2},\\
&1^{\theta W} := \frac{\partial 1^{\theta}}{\partial W} = -\textstyle\sum_{i=1}^n \frac{\lambda_2^2 \partial w(y_i,x_i,\theta)/\ \partial\theta^T}{z_i^2}
\end{aligned}$$

With those derivatives, Newton-Raphson algorithm can be used to get maximum likelihood estimates of ψ. These estimates could be plugged into the equation 2.2, so that we could estimate the unbiased mean of the target variable, as shown in the equation 2.6, since in its very essense, the method reweights non missing values in such way, that for the aggregate estimator, bias caused by non-ignorable non-response would be corrected.

$$\hat{\bar{Y}} = \sum_{i=1}^{m} \hat{p}_i y_i \tag{2.6}$$

Variance for the estimator can be calculates with a use of the bootstrap, by drawing a set number of samples from our sample and calculating regular variance on it, since the analytical form of the variance estimator was not provided.

2.2.1 Link functions and its derivatives

The response model $w(y, X, \theta)$ if logistic regression was assumed, would look like the equation 2.7, equation 2.8 for probit and 2.9 for cloglog.

$$w_1(y,x,\theta) = \frac{\exp(\eta)}{1+\exp(\eta)} \tag{2.7}$$

$$w_2(y,x,\theta) = \Phi(\eta) = \frac{1}{2\pi}\int_{-\infty}^{\eta} \exp\left(-\frac{\eta^2}{2}\right), \tag{2.8}$$

$$w_3(y,x,\theta) = 1 - \exp(-\exp(\eta)) \tag{2.9}$$

where

$$\eta = \theta_0 + \theta_1 x_{1,i} + ... + \theta_{k-1} x_{k-1,i} + \theta_k y_i$$

Their first and second derivatives, and first and second derivatives of logarithms are presented in equations 2.10, 2.11 and 2.12

- logit link

$$\begin{aligned}
&\tfrac{\partial w_1(y_i,x_i,\theta)}{\partial\theta} = h_i\left(1 - w_1(y_i,x_i,\theta)\right) w_1(y_i,x_i,\theta),\\
&\tfrac{\partial^2 w_1(y_i,x_i,\theta)}{\partial\theta\partial\theta} = h_i h_i^T \left(1 - 2w_1(y_i,x_i,\theta)\right)\left(1 - w_1(y_i,x_i,\theta)\right) w_1(y_i,x_i,\theta),\\
&\tfrac{\partial \ln w_1(y_i,x_i,\theta)}{\partial\theta} = h_i\left(1 - w_1(y_i,x_i,\theta)\right),\\
&\tfrac{\partial^2 \ln w_1(y_i,x_i,\theta)}{\partial\theta\partial\theta} = -h_i h_i^T \left(1 - w_1(y_i,x_i,\theta)\right) w_1(y_i,x_i,\theta),
\end{aligned} \tag{2.10}$$

- probit link

$$\begin{aligned}
&\tfrac{\partial w_2(y_i,x_i,\theta)}{\partial\theta} = -h_i \exp\left(-\tfrac{(\eta)^2}{2}\right)\tfrac{\eta}{\sqrt{2\pi}},\\
&\tfrac{\partial^2 w_2(y_i,x_i,\theta)}{\partial\theta\partial\theta} = -h_i h_i^T \left(1 - (\eta)^2\right)\exp\left(-\tfrac{(\eta)^2}{2}\right)\tfrac{1}{\sqrt{2\pi}},\\
&\tfrac{\partial \ln w_2(y_i,x_i,\theta)}{\partial\theta} = -h_i\left(\eta\right),\\
&\tfrac{\partial^2 \ln w_2(y_i,x_i,\theta)}{\partial\theta\partial\theta} = -h_i h_i^T,
\end{aligned} \tag{2.11}$$

- cloglog link

$$\begin{aligned}
&\tfrac{\partial w_3(y_i,x_i,\theta)}{\partial\theta} = h_i \exp\left(-\exp\left(\eta\right)\right)\exp\left(\eta\right),\\
&\tfrac{\partial^2 w_3(y_i,x_i,\theta)}{\partial\theta\partial\theta} = h_i h_i^T \left(1 - \exp\left(\eta\right)\right)\exp\left(-\exp\left(\eta x\right)\right)\exp\left(\eta x\right),\\
&\tfrac{\partial \ln w_3(y_i,x_i,\theta)}{\partial\theta} = h_i \exp\left(-\exp\left(\eta\right)\right)\tfrac{\exp(\eta x)}{1-\exp(-\exp(\eta))},\\
&\tfrac{\partial^2 \ln w_3(y_i,x_i,\theta)}{\partial\theta\partial\theta} = h_i h_i^T \left(1 - \left(1 + \tfrac{\exp(-\exp(\eta))}{1-\exp(-\exp(\eta))}\right)\exp\left(\eta\right)\right)\left(\exp\left(-\exp\left(\eta\right)\right)\right)\tfrac{\exp(\eta)}{1-\exp(-\exp(\eta))},
\end{aligned} \tag{2.12}$$

where

$$h_i = \begin{pmatrix} 1 \\ \mathbf{x}_i \\ y_i \end{pmatrix}. \tag{2.13}$$

2.3 Estimation of parameters using Newton-Raphson method

2.3.1 Starting parameters

As Bücker (2011) also noted, the choice of the starting parameter plays a decisive role for the Newton method. This should be as close as possible to the true value, to guarantee the convergence of the algorithm to the zero point. Sensible starting values can be found for the parameters W and λ_1 on the basis of their properties. Since $W = P(R = 1)$ is the proportion of missing realizations of Y, $W^{(0)} = m/n$ is a good estimate and thus a good starting value for W. Since $\lambda_1 \xrightarrow{P} 0$, $\lambda^{(0)} = 0$ is a reasonable starting value for the Lagrangian parameter. For θ it is more difficult to find good estimates that are close to the true parameter. If a relationship between Y and X is known, this can be used. If $y_i \approx m(x_i, \eta)$ $(i = 1, ..., N)$, the missing observations of the random variable Y can be estimated with the help of this model and then the parameter vector θ can be estimated with the help of the new observations. These then supply $\theta^{(0)}$ as the starting value. In this case, the quality of the starting value for θ essentially depends on the quality of the model m. This can, for example, be a regression model estimated on the basis of the observed data. The stronger the distortion due to the missing data, the less suitable the start values will be.

2.3.2 The algorithm

In this subsection, the steps for the Newton-Raphson algorithm, shown earlier by Bücker (2011) and Jiahua Chen et al. (2008), are presented. This algorithm can be used for the Empirical Likelihood based method, described in the previous subsection.

To numerically determine the zero of f or the maximum of the log-likelihood, the $k-$th step of the Newton method goes through

$$\psi^{(k+1)} = \psi^{(k)} - J(\psi^{(k)})^{-1} f(\psi^{(k)})$$

An important point when estimating by means of empirical likelihood is that the secondary condition $p_i > 0$ must always be fulfilled. This should be checked in every step, otherwise the algorithm will usually not converge. If the secondary condition is violated, the step length of the algorithm is to be reduced. Finally, the following pseudo-code gives a possibility to estimate the parameters with the help of the Newton-Raphson algorithm:

1. Choose as starting values $\lambda_1^{(0)} = 0$, $W^{(0)} = m/n$ and $\theta^{(0)}$ (as described above).

Set $\gamma = 1$, $\psi^{(0)} = (\lambda^{(0)T}, \theta^{(0)T},\ W^{(0)})$. Also choose $k = 0$ as the iteration step and the tolerance level $\varepsilon = 10^{-8}$.

2. Determine

$$\nabla = -J(\psi^{(k)})^{-1} f(\psi^{(k)}).$$

If $\|\nabla\| < \varepsilon$ end the iteration and give $\lambda_1^{(k)}, W^{(k)}, \theta^{(k)}$ and $\lambda_2^{(k)}$ as an estimator. Otherwise continue with step 3.

3. Calculate $\delta = \gamma\nabla$. If for

$$z_i(\psi^{(k)}) = \left[1 + \lambda_1^{(k)T}(x_i - \mu_x) + \frac{m/n - 1}{1 - W^{(k)}}(w(y_i, x_i, \theta^{(k)}) - W^{(k)})\right]$$

the inequality

$$z_i(\psi^{(k)} + \delta) \leq 0$$

holds or for

$$l(\psi^{(k)}) = \sum_{i=1}^{m} \ln w(y_i, x_i, \theta^{(k)}) + (n - m)\ln\Big(1 - W^{(k)}\Big) - \sum_{i=1}^{m} \ln z_i(\psi^{(k)})$$

the inequality

$$l(\psi^{(k)} + \delta) \leq l(\psi^{(k)})$$

holds, set $\gamma = \gamma/2$ and repeat the step. Otherwise go with the next step.

4. Set

$$\psi^{(k+1)} = \psi^{(k)} + \delta$$

and

$$\gamma = (k+1)^{-1/2}$$

Increase k by 1 and go back to step 2.

2.4 Summary

In this section, the method based on Empirical likelihood for dealing with non-ignorable non-response was presented. We discussed how the method works and showed equations, that were crucial for implementing the method. Apart from that, the estimation process, in the form of Newton-Raphson algorithm, was shown step by step, with discussion about the choice of starting parameters. We have also presented 3 different response models, with all the necessary calculations that were double checked by us, and implemented in the NMAR package, presented in the following chapter.

Chapter 3

Implementation of the empirical likelihood based method in `nmar` package

3.1 The `nmar` package

`nmar` is a statistical package that implements methods for dealing with non-ignorable non-response problem in R. Its name stands for "Not missing at random", which is a response mechanism, presented by Little and Rubin (1987). In its current iteration, it implements empirical likelihood (EL) based approach, that was discussed earlier. The package aims to provide a set of tools to deal with non-ignorable non-response, mainly though the use of corrections methods, like the one presented in chapter 2, hence the name.

`nmar` package at its current form has a set of features, geared towards estimation under non-ignorable non-response assumption for implemented EL based method.

First and foremost, these include the ability to generate starting parameters for maximum likelihood estimation of semi-parametric model 2.3. There are two implemented approaches to get the starting parameters:

- with linear regression,
- with generalized calibration.

Linear regression is a simple and naive method of getting starting parameters for the the response model, and can only be used for the cases when auxiliary information is available for non-respondents. It models missing target with auxiliaries and then uses its estimates to train model 2.7.

Generalized calibration, in comparison, does not require auxiliaries for the missing target, so it can also be used for the case shown in table 2.1 *Case II*. But its limitation is that it requires the number of dependent variables for both target and response model to be the same, which makes it less flexible for other two cases. Function `genCalib()` from package `sampling` developed by (Tillé and Matei 2021) is used for generalized calibration in `nmar` package.

There are two working implementations for the same EL based method. One of them is implemented with a use of `rootSolve` package (Soetaert 2009), and another with Newton-Raphson algorithm, implemented from scratch based on steps mentioned by Bücker (2011). The difference between them is that `rootSolve`-based solution takes in first derivatives 2.4 and is a bit more stable, while the Newton-Raphson-based one makes use of Jacobian matrix (2.5), allows for easy tolerance level adjustment, which not only makes calculation, on average, faster but also makes it more flexible.

Logistic regression is not the only response mechanism that can be assumed in the nmar package. The user can also choose among probit and cloglog functions. The derivatives for those were calculated partly by hand, partly with a use of Clausen and Sokol (2020) package called `Deriv`. Derivatives of the link functions, calculated and implemented in the nmar package, can be seen in the appendix to this thesis.

The package can deal with missing data cases described in table 2.1. Apart from that, when no weights are provided, the package would still be able to deal with both cases presented in table 2.1, but certain assumptions are being made. For the missing data case from table 2.1 *Case I*, number of rows in the provided data is assumed to be the size of the sample, equal weights are being given to the respondents, where for each responded the weight is n/m. For the missing data case from table 2.1 *Case II*, that is not really possible, since only the data for respondents is available, so the package would only be able to make estimates if population size is provided as a separate parameter, where after that same principal applies, as for the 2.1. Due to certain design choices, the package does not actually take any weights into account during estimation of the parameters, but applies them after the fact in a selected estimator. This somewhat changes appearance of the estimator, where is the weights are basically used to retrieve population size and scale down the results. The way to incorporate weights in the estimation process is currently during development, so they would not appear in simulation study.

After parameters of the model 2.2 are estimated, estimators of mean can be calculated as well. Apart from estimator 2.6, there are two more reference estimators of mean implemented.

$$\hat{\bar{Y}} = \frac{\sum_{i=1}^{m} \frac{y_i}{w(y_i,x_i,\theta)}}{\sum_{i=1}^{m} \frac{1}{w(y_i,x_i,\theta)}} \frac{n}{N} \tag{3.1}$$

$$\hat{\bar{Y}} = \frac{m\sum_{i=1}^{m}\frac{y_i}{w(y_i,x_i,\theta)}}{N^2} \tag{3.2}$$

Though trial and error, estimators 3.1 and 3.2 were adapted from what Jing Qin et al. (2002) used in their article for the same method. The $\frac{n}{N}$ term in the back of every estimator is optional and would only be used if naive method of incorporating weights is used. Those estimators are a bit less sophisticated then estimator 2.2, because unlike 2.2, they fully rely on information from response model, where 2.2 also takes into account population mean for auxiliaries (when available).

3.2 Main functions

There are two external function for potential users to interact to, `nmar()` and `estimate()`. Those functions were built with a use of not only base R functionality, but also by taking advantage of functionality, provided by other packages, that are worth mentioning here.

Among them `formula.tool`, developed by Brown (2018), was used to derive information from formulas provided by the user, which helped with simplifying interactions with external functions for potential users. There are also packages like `doParallel` Corporation and Weston (2022) and `foreach` Microsoft and Weston (2022) that helped tremendously with improving runtime of the boostrap method, used for calculating variance of the estimates, but also made to give potential users greater control over how parallelization is used for the calculations (more examples in Usage section).

3.2.1 the `nmar` function

The function implements methods like Empirical Likelihood to correct non-ignorable non-response in provided set of data. Models need to be provided, where response and target relationships with auxiliary variables are specified. If starting parameters are not available, model for estimation of starting parameters with linear models can be provided into control.target, otherwise target model will be used. Empirical likelihood has two viable implementations. Rootsolve implementation is used by default and is generally more stable. Implementation that uses Newton-Raphson algorithm tends to be faster but can also be less reliable.

`Usage`

```
nmar(
    data,
    response,
    target,
    response.family = c("logit", "probit", "cloglog"),
    target.family = NULL,
    pop.totals = NULL,
    pop.means = NULL,
    type = c("EL", "GMM", "LM.START", "GC.START"),
  control.nmar = ctrl.nmar(start.params = NULL, start.method = "LM.START",
        estimation.method = "rootsolve",
        pop.size = NULL, eps = 1e-08, weights.use = "PCAD"),
    control.target = ctrl.target(start.model = NULL),
    control.response = ctrl.response(),
    subset.args = NULL,
    weights = NULL,
    warning.messages = T
    )
```

Arguments

data dataframe with missing data

response formula notation for selecting response variable and the variables it assumed to depend on

target formula notation for establishing relationship between variables, to be used in glm

response.family one of the following link functions: "logit", "probit" or "cloglog", that is assumed in response model of EL method

target.family .

pop.totals .

pop.means population means or estimator of population mean for each auxiliary variable used in the target model

type what estimation technique should be applied (e.g. EL, GMM or LM.START)

control.nmar additional arguments for `nmar` funtions provided with `ctrl.nmar()`, where `start.params` – starting parameters for the EL method, if not provided would be calculated by default. `estimation.method` – selection between differant implementation for the same method, like "rootsolve" and "newton-raphson", `eps` – tolerance level used only for the "newton-raphson" implementation, `weights.use` – the way, weights are going to be used for later estimation (only post calculation at the moment)

`control.target` additional arguments for defining relationship between target and auxiliary variables provided with `ctrl.target()`.

`control.response` additional arguments for defining relationship between response, target and auxiliary variables provided with `ctrl.response()`.

`subset.args` place to pass arguments to `subset` function from base R in quotation marks (data argument does not need to be passed)

`weights` vector of design weights or name of the column from data, that contains them

`warning.messages` turn off warnings for possibly faster performance

`nmar` function returns an object that corresponds to a selected method, for Empirical likelohood "empLik" object will be returned, that can be later fed into `estimate` to get corrected estimators.

Other methods like `print`, `summary` and `print.summary` are also available for the 'empLik' object.

An object of class "empLik" is a list that contains:

call funtion call

target target formula provided in `nmar` funtion

response response formula provided in `nmar` function

model model formula provided in `nmar` function for estimating starting parameters

params fitted parameters that include: Lagrange multipliers l1 and l2, thetas and response rate W

iter number of iterations it took Newton-Raphson algorithm

pi_conds constraints check

link response family

data data inputed into `nmar` function

estInp list of parameters for internal functions that include things like preprocessed data and selected link functions

meanAux population or estimated column means of auxiliary variables

m number of rows in provided dataset

n number of responders in provided dataset

3.2.2 estimate function

This is function that, given a `nmar` object and estimator name or names, calculates them for the target variable subject to non-ignorable non-response. The functions returns an object for which print and summary method are defined, that itself is a list of selected estimator values, their standard deviations and variations.

```
## S3 method for class 'empLik'
estimate(
  nmarObject,
  func = c("mean", "mean.1", "mean.2", "total"),
  variance = c("analytic", "bootstrap"),
 control = list(nBS = 500, estimate.only = F, eps = 1e-05, start.seed = "none",
    parallel = F, cluster = NULL, mc.cores = NULL)
)
```

`nmarObject` empLik object, which is an output of the `nmar` function

`func` function or vector of functions to be calculated out of ("mean","mean.1","mean.2","total")

`variance` method of calculating variance (currently only bootstrap is available)

`control` control arguments, where `nBS` – number of bootstrap samples (500 by default), `estimate.only` – binary variable, where False results into no variation estimation, `eps` – tolerance level regulator (the higher, the faster), `start.seed` – seed from which sampling will start, `parallel` – binary variable, if False, parallelization will not be used `cluster` – user-defined cluster for parallelization, `mc.cores` – number of processor cores used for parallelization

3.3 Usage examples

You can install the development version of nmar like so:

```
install_github("ncn-foreigners/nmar")
```

Listing 3.1. package instalation

There are some usage examples presented in this section, which can also be found in the static webpage for the package, among other things. The webpage was build with a use of pkgdown package, developed by Wickham et al. (2022).

3.3.1 Calculating starting parameters

3.3.1.1 Starting parameters with linear regression

First lets generate a simple dataset that consists of two independent variables Z,X and one dependent Y with the following relationship between them.

$$Y = 0.5X + \frac{ZX^{1/2}}{5},$$

where $Z \sim N(0,1)$, $X \sim \chi^2(6)$.

```
set.seed(352)
N <- 10000
Z <- rnorm(N)
X <- rchisq(N,df=6)/2
Y <- 0.5*X + (Z*(X^(1/2)))/5
sim.data <- data.frame(Y,X,Z)
```

Listing 3.2. Starting parameters example 1.1

Missing values in Y are generated with a following logistic response mechanism.

$$w(y,x,\theta) = \frac{exp(\theta_1 y + \theta_0)}{1 + exp(\theta_1 y + \theta_0)}, \theta_1 = 0.8, \theta_0 = -0.5$$

```
w <- function(theta,y){
    ex <- exp(theta[1] + theta[2]*y)
    return(ex/(1+ex))
}
resp <- w(c(0.8,-0.5),Y)
R <- rbinom(N,1,resp)
data.resp <- sim.data[R == TRUE,]
mx <- as.matrix(colMeans(sim.data[,c("X","Z")]))
data.missing <- data.frame(Y = ifelse(R==0,NA,Y),X = X,Z=Z)
```

Listing 3.3. Starting parameters example 1.2

Now we can load nmar package and calculate starting parameters for the EM method.

The output coefficients model the relationship between response and target variable and can be used as a starting parameters for empirical likelihood method.

```
start <- nmar::nmar(target = Y~X+Z,
                    response = R~Y,
                    data = data.missing,
                    type = "LM.START")

start
```

Listing 3.4. Starting parameters example 1.3

```
##
## Call:  glm(formula = response.form, family = binomial, data = data.predicted)
##
## Coefficients:
## (Intercept)            Y
##      0.8677      -0.5352
##
## Degrees of Freedom: 9999 Total (i.e. Null);  9998 Residual
## Null Deviance:      13850
## Residual Deviance: 13320     AIC: 13320
```

Listing 3.5. Starting parameters example 1.3 – results

3.3.1.2 Starting parameters with generalized calibration

This method of generating starting parameters for the Empirical Likelihood is less flexible then the one showed in another example. It requires the number of dependent variables for both target and response model to be the same. The advantage to using it over the other one is that it allows to estimate starting parameters, when full set of auxiliaries is not available, but instead only the global mean for then or an estimator of that mean.

First lets generate a simple dataset that consists of two independent variables Z,X and one dependent Y with the following relationship between them.

$$Y = 0.5X + \frac{ZX^{1/2}}{5}$$

$Z \sim N(0,1)$ and $X \sim \chi^2(6)$.

```
set.seed(352)
N <- 10000
Z <- rnorm(N)
X <- rchisq(N,df=6)/2
Y <- 0.5*X + (Z*(X^(1/2)))/5
sim.data <- data.frame(Y,X,Z)
```

Listing 3.6. Starting parameters example 2.1

Missing values in Y are generated with a following logistic response mechanism.

$$w(y,x,\theta) = \frac{exp(\theta_2 y + \theta_1 x + \theta_0)}{1 + exp(\theta_2 y + \theta_1 x + \theta_0)}, \theta_2 = 0.2, \theta_1 = 0.8, \theta_0 = -0.5$$

```
w <- function(theta,y){
    ex <- exp(theta[1] + theta[2]*y)
    return(ex/(1+ex))
}
resp <- w(c(0.2,0.8,-0.5),Y)
R <- rbinom(N,1,resp)
data.resp <- sim.data[R == TRUE,]
mx <- as.matrix(colMeans(sim.data[,c("X","Z")]))
data.missing <- data.frame(Y = ifelse(R==0,NA,Y),X = X,Z=Z)
```

Listing 3.7. Starting parameters example 2.2

Now we can load `nmar` package and calculate starting parameters for the EM method with generalized calibration method.

The output coefficients model the relationship between response and target variable and can be used as a starting parameters for empirical likelihood method.

```
start <- nmar::nmar(target = Y~X+Z,
                    response = R~Y+X,
                    data = data.missing,
                    pop.means = mx,
                    type = "GC.START")
start
```

Listing 3.8. Starting parameters example 2.3

```
## (Intercept)           Y           X
## -0.38593902  0.20565325  0.03484367
```

Listing 3.9. Starting parameters example 2.3 – results

3.3.2 Empirical likelihood estimation

3.3.2.1 Estimation with nmar

In this example, starting parameters and data set from previous example will be used.

```
start.par
```

```
## (Intercept)           Y
##  -0.9713623   0.1963578
```

The input data should have missing data for the target variable and look something like this.

```
head(data.missing)
```

```
##            Y         X          Z
## 1         NA 4.4880033  1.0560717
## 2         NA 1.4013262 -0.6952264
## 3 2.1108904 3.7100855  0.6641397
## 4        NA 4.0658324  0.1743772
## 5 0.4867665 0.7917198  0.5108337
## 6 0.5240237 0.9683684  0.2024244
```

True mean of auxiliary variables or the closest approximation is also needed.

```
mx
```

```
##              [,1]
## X  2.9985930319
## Z -0.0004574139
```

There are two variants for empirical likelihood estimation in nmar package. One of them uses only first derivatives and makes calculations with `rootSolve`. Another uses both first and second derivatives and uses Newton-Raphson algorithm.

Rootsolve based method is a little bit more stable, and can be utilized with providing EL for a type in `nmar` function.

```
RS <- nmar(data = data.missing,
           response = R~Y,
           target = Y~X+Z,
           response.family="logit",
           pop.means = mx,
           type="EL")

RS
```

Listing 3.10. Estimation with `nmar` example 1.1

```
## Warning: Starting parameters were not provided, lm approximation will be used
## instead.
## Warning: Model for lm approximation was not provided, target will be used instead
    .

##
## Call:
## nmar(data = data.missing,
##      response = R ~ Y,
##      target = Y ~ X + Z,
##      response.family = "logit",
##      pop.means = mx,
##      type = "EL")
##
## Coefficients:
##                  l1                    l2                 theta
    W
## 0.002585, -0.011144                 2.959      -0.9719, 0.1968
    0.3382
```

Listing 3.11. Estimation with `nmar` example 1.1 – results

Newton-Rapson based method can be faster with lower presicion that can be manipulated with a change of eps, and can be utilized with providing EL for a type in `nmar` function.

```
NR <- nmar(data = data.missing,
           response = R~Y,
           target = Y~X+Z,
           response.family="logit",
           pop.means = mx,
           type="EL",
           control.nmar = ctrl.nmar(
                    estimation.method = "newton-raphson",
                    eps = 1e-11)

summary(NR)
```

Listing 3.12. Estimation with `nmar` example 1.2

```
## Warning: Starting parameters were not provided, lm approximation will be used
## instead.
## Warning: Model for lm approximation was not provided, target will be used instead
    .

##
## Call:
## nmar(data = data.missing,
##      response = R ~ Y,
##      target = Y ~ X + Z,
##      response.family = "logit",
##      pop.means = mx,
##      type = "EL",
##      control.nmar = ctrl.nmar(estimation.method = "newton-raphson",
##                               eps = 1e-11))
##
## Link: logit
##
## Coefficients:
##                  l1                     l2                 theta
    W
## 0.002584, -0.011143                2.959       -0.9727, 0.1967
    0.338
##
##
## Iterations: 5
```

Listing 3.13. Estimation with `nmar` example 1.2 – results

When summary function is used on the empLik object, maximization conditions and link are also provided. The most desirable outcome is when all binary conditions are TRUE except for the last one. When different eps are selected ch2, ch3, ch4 can deviate a little bit from desired 1, 0, 0, but ch1b should always be TRUE.

The output object can be used to calculate different characteristics of the target variable with estimate function.

```
estimate(RS)
```

Listing 3.14. Estimation with `nmar` example 1.3

```
##          Estimate         Sd         Var
## Mean      1.50138 0.0254697 0.000648707
## Mean.1    1.50134 0.0255052 0.000650514
## Mean.2    1.50137  0.031683  0.00100381
## Total     7506.83   14544.2   211532502
```

Listing 3.15. Estimation with `nmar` example 1.3 – results

The function has 3 different mean estimators built in.

3.3.2.2 Estimation with nmar when weights are available

For this example, new simulated data set will be generated. To make it easier to follow, for those that read previous examples, the data is generated in an identical manner with an addition of weights at the very end of the process.

```
library(nmar)
set.seed(3543)
N <- 5000
Z <- rnorm(N)
X <- rchisq(N,df=6)/2
Y <- 0.5*X + (Z*(X^(1/2)))/5
sim.data <- data.frame(Y,X,Z)
```

Listing 3.16. Estimation with nmar example 2.1

Response mechanism is logistic, same as before.

```
w <- function(theta,y){
    ex <- exp(theta[1] + theta[2]*y)
    return(ex/(1+ex))
}

resp <- w(c(-1,.2),Y)
R <- rbinom(N,1,resp)
data.resp <- sim.data[R == TRUE,]
mx <- as.matrix(colMeans(sim.data[,c("X","Z")]))
data.missing <- data.frame(Y = ifelse(R==0,NA,Y),X = X,Z=Z)
weights <- ifelse(is.na(data.missing$Y),NA,sample(seq(1,4,0.2),N,replace = T))
```

Listing 3.17. Estimation with nmar example 2.2

In the final step, weights are sampled with replacement from sequence of numbers between 1 and 4 with step of 0.2.There are no weights for the rows, where our target variable Y has missing values, generated in this example, since only such weights can be utilized by the package. In this particular example, unused weights are hollowed out with *NAs*, but the package can also accept weights vector that is without *NAs* as long as the order of those weights corresponds to the order of of rows with available target.

In comparison to the previous examples, here the ability to use a formula for the target, if relationship is know, is demonstrated.

```
start <- nmar(target = Y~Z*(I(X^(1/2))),
              response = R~Y,
              data = data.missing,
              type = "LM.START")
(start.par <- start$coefficients)
```

Listing 3.18. Estimation with nmar example 2.3

```
## (Intercept)           Y
##  -0.9662068   0.1928061
```

Listing 3.19. Estimation with nmar example 2.3 – results

The input data should also have missing data and look something like this.

```
head(data.missing)
```

```
##           Y         X          Z
## 1        NA 4.4880033  1.0560717
## 2        NA 1.4013262 -0.6952264
## 3 2.1108904 3.7100855  0.6641397
## 4        NA 4.0658324  0.1743772
## 5 0.4867665 0.7917198  0.5108337
## 6 0.5240237 0.9683684  0.2024244
```

True mean of auxiliary variables or the closest approximation is also needed, and since the weights are going to be used only post calculations, the mean values do need to calculated with accordance with them.

```
mx
```

```
##             [,1]
## X   2.9985930319
## Z -0.0004574139
```

This time only Newton-Raphson variant will be used for estimations with weights.

```
NR <- nmar(data = data.missing,
          response = R~Y,
          target = Y~X+Z,
          response.family="logit",
          pop.means = mx,
          type="EL",
          control.nmar = ctrl.nmar(estimation.method = "newton-raphson",
                                   eps = 1e-11),
          weights = weights

summary(NR)
```

Listing 3.20. Estimation with `nmar` example 2.4

```
## Warning: Starting parameters were not provided, lm approximation will be used
   instead.
## Warning: Model for lm approximation was not provided, target will be used instead
   .
##
## Call:
## nmar(data = data.missing,
##      response = R ~ Y,
##      target = Y ~ X + Z,
##      response.family = "logit",
##      pop.means = mx,
##      type = "EL",
##      control.nmar = ctrl.nmar(estimation.method = "newton-raphson",
##                               eps = 1e-11),
##      weights = weights)
##
## Link: logit
##
## Coefficients:
##  l1                  l2               theta        W
## 0.002584, -0.011143  2.959       -0.9727, 0.1967    0.338
##
```

```
## Iterations: 5
```

Listing 3.21. Estimation with `nmar` example 2.4 – results

The estimate function is described in detail in another example. There is no need to do anything extra at this point, even though weights are going to be used.

```
estimate(NR)
```

Listing 3.22. Estimation with `nmar` example 2.5

```
##         Estimate        Sd        Var
## Mean     1.25831 0.0322541 0.00104032
## Mean.1   1.25817  0.032256 0.00104045
## Mean.2   1.25894 0.0322637 0.00104094
## Total    6294.68    12353.3  152603561
```

Listing 3.23. Estimation with `nmar` example 2.5 – results

3.3.2.3 Estimation with `nmar` with partial auxiliaries

Generated data from the previous examples, as well `nmar_object`, will be used here.

```
library(nmar)
set.seed(3543)
N <- 5000
Z <- rnorm(N)
X <- rchisq(N,df=6)/2
Y <- 0.5*X + (Z*(X^(1/2)))/5
sim.data <- data.frame(Y,X,Z)
w <- function(theta,y){
    ex <- exp(theta[1] + theta[2]*y)
    return(ex/(1+ex))
}
resp <- w(c(-1,.2),Y)
R <- rbinom(N,1,resp)
data.resp <- sim.data[R == TRUE,]
mx <- as.matrix(colMeans(sim.data[,c("X","Z")]))
data.missing <- data.frame(Y = ifelse(R==0,NA,Y),X = X,Z=Z)
data.missing_1 <- data.missing[complete.cases(data.missing),]
set.seed(674)
weights <- sample(1:5,1691,replace = T)
```

Listing 3.24. Estimation with `nmar` example 3.1

But this time, all of the auxiliaries will not be used. For that to work, starting parameters should be estimated with generalized calibration method. There are two possible cases:

- No weights, population size available

```
EL.object_1 <- nmar(data = data.missing_1,
                    response = R ~ Y+X,
                    target = Y ~ X+Z,
                    response.family="logit",
                    pop.means = mx,
                    control.nmar = ctrl.nmar(
                                start.method = "GC.START",
                                pop.size = N),
                    type="EL",
                    weights = NULL)
estimate(nmarObject = EL.object_1)
```

Listing 3.25. Estimation with nmar example 3.2

```
## Warning: Starting parameters were not provided, lm approximation will be used
## instead.
## Warning: Model for lm approximation was not provided, target will be used instead
    .

##        Estimate       Sd       Var
## Mean     1.50173 0.122238 0.0149422
## Mean.1   1.50181 0.120516  0.014524
## Mean.2   1.50181  0.12311 0.0151561
## Total    7509.04  16500.9 272279358
```

Listing 3.26. Estimation with nmar example 3.2 – results

- Use of weights

```
EL.object_2 <- nmar(data = data.missing,
                    response = R~Y+X,
                    target = Y~X+Z,
                    response.family="logit",
                    pop.means = mx,
                    control.nmar = ctrl.nmar(
                                   start.method = "GC.START"),
                    type="EL",
                    weights = weights)
estimate(nmarObject = EL.object_2)
```

Listing 3.27. Estimation with nmar example 3.3

```
## Warning: Starting parameters were not provided, lm approximation will be used
## instead.
## Warning: Model for lm approximation was not provided, target will be used instead
    .

##        Estimate        Sd         Var
## Mean     1.4888 0.0283548 0.000803996
## Mean.1  1.48887 0.0286992 0.000823642
## Mean.2  1.48887  0.030654 0.000939669
## Total   7444.36   14462.9   209176800
```

Listing 3.28. Estimation with nmar example 3.3 – results

3.3.3 The estimate function usage examples

Generated data from the previous examples, as well nmar_object, will be used here.

```
library(nmar)
set.seed(3543)
N <- 5000
Z <- rnorm(N)
X <- rchisq(N,df=6)/2
Y <- 0.5*X + (Z*(X^(1/2)))/5
sim.data <- data.frame(Y,X,Z)
w <- function(theta,y){
    ex <- exp(theta[1] + theta[2]*y)
    return(ex/(1+ex))
}
resp <- w(c(-1,.2),Y)
R <- rbinom(N,1,resp)
data.resp <- sim.data[R == TRUE,]
mx <- as.matrix(colMeans(sim.data[,c("X","Z")]))
data.missing <- data.frame(Y = ifelse(R==0,NA,Y),X=X,Z=Z)
start <- nmar(target = Y~X+Z,
              response = R~Y,
              data = data.missing,
              type = "LM.START")
start.par <- start$coefficients
EL.object <- nmar(data = data.missing,
                  response = R~Y,
                  target = Y~X+Z,
                  response.family="logit",
                  pop.means = mx,
                  type="EL")
estimate(EL.object)
```

Listing 3.29. Estimate function example 1.1

To make use of the `estimate` function for EL, one needs only an object of a class `emplik`. The output is a data.frame with each selected estimators, in each row and their respective estimates with standard deviation and variation calculated with bootstrap method.

```
##        Estimate        Sd         Var
## Mean     1.50138 0.0254697 0.000648707
## Mean.1   1.50134 0.0255052 0.000650514
## Mean.2   1.50137  0.031683  0.00100381
## Total    7506.83   14544.2   211532502
```

Listing 3.30. Estimate function example 1.1 – results

By default, the function displays and calculates all available estimators and bootstrap variance for them. User can select estimator to reduce computation time a little.

```
estimate(EL.object, func = c("mean","total")
```

Listing 3.31. Estimate function example 1.2

```
##       Estimate        Sd         Var
## Mean    1.50138 0.0259581 0.000673824
## Total   7506.83   14535.2   211272080
```

Listing 3.32. Estimate function example 1.2 – results

The function, by default, does not use parallelization for the bootstrap variance calculation on all the cores on the processor minus 1. The starting seed for the boostrap variation calculations can be provided.

```
estimate(EL.object, control = list(start.seed = 123))
```

Listing 3.33. Estimate function example 1.3

```
##        Estimate        Sd         Var
## Mean     1.50138 0.0241097 0.000581276
## Mean.1   1.50134 0.0241455 0.000583003
## Mean.2   1.50137 0.0309896 0.000960358
## Total    7506.83   14559.6   211981588
```

Listing 3.34. Estimate function example 1.3 – results

Things that can changed to decrease or increase computing time of boostrap variance, include:

- Changing the number of cores used for parallelization, for example to 4.

```
estimate(EL.object, control = list(mc.cores = 4))
```

Listing 3.35. Estimate function example 1.4

```
##         Estimate         Sd         Var
## Mean     1.50138 0.0238197 0.000567377
## Mean.1   1.50134 0.0238536 0.000568994
## Mean.2   1.50137 0.0294035 0.000864568
## Total    7506.83   14583.8   212687313
```

Listing 3.36. Estimate function example 1.4 – results

- Using user-defined cluster

```
library(parallel)
library(doParallel)
library(foreach)
mc.cores <- parallel::detectCores() -- 1
cluster_for_EL_estimate_boostrap <-
                        parallel::makePSOCKcluster(mc.cores)
doParallel::registerDoParallel(
                      cluster_for_EL_estimate_boostrap)
estimate(NR,control = list(parallel = T,
            cluster = cluster_for_EL_estimate_boostrap)
parallel::stopCluster(cluster_for_EL_estimate_boostrap)
```

Listing 3.37. Estimate function example 1.5

```
##         Estimate         Sd         Var
## Mean     1.50138 0.0248367 0.000616864
## Mean.1   1.50134 0.0248751  0.00061877
## Mean.2   1.50137 0.0301758 0.000910579
## Total    7506.83   14562.1   212055509
```

Listing 3.38. Estimate function example 1.5 – results

- Not using parallelization (default)

```
estimate(EL.object, control = list(parallel = F))
```

Listing 3.39. Estimate function example 1.6

```
##         Estimate         Sd         Var
## Mean     1.50138 0.0248559 0.000617814
## Mean.1   1.50134 0.0248959 0.000619804
## Mean.2   1.50137 0.0299816 0.000898898
## Total    7506.83   14557.7   211925344
```

Listing 3.40. Estimate function example 1.6 – results

- Increasing tolerance level (eps) to speed up the calculations (with cost of some accuracy).

```
estimate(EL.object, control = list(eps = 1e-4))
```

Listing 3.41. Estimate function example 1.7

```
##         Estimate        Sd         Var
## Mean     1.50138 0.0254332 0.000646847
## Mean.1   1.50134 0.0254743 0.000648939
## Mean.2   1.50137 0.0307677 0.000946648
## Total    7506.83   14539.1   211386327
```

Listing 3.42. Estimate function example 1.7 – results

- Using less boostrap samples for calculating variation (500 is default)

```
estimate(EL.object, control = list(nBS = 100))
```

Listing 3.43. Estimate function example 1.8

```
##         Estimate        Sd         Var
## Mean     1.50138 0.0267406 0.000715062
## Mean.1   1.50134 0.0267747 0.000716884
## Mean.2   1.50137   0.03164  0.00100109
## Total    7506.83   14614.4   213581660
```

Listing 3.44. Estimate function example 1.8 – results

- Not counting standard deviation and variation

```
estimate(EL.object, control = list(estimate.only = T))
```

Listing 3.45. Estimate function example 1.9

```
##         Estimate   Sd  Var
## Mean     1.50138 NULL NULL
## Mean.1   1.50134 NULL NULL
## Mean.2   1.50137 NULL NULL
## Total    7506.83 NULL NULL
```

Listing 3.46. Estimate function example 1.9 – results

3.4 Summary

In this section we presented our up-and-coming statistical package, called `nmar`. Most prominent features of the packages were discussed, and some design choices uncovered. We have also gone though the usage cases, where the process of estimation from calculating the starting parameters to calculating estimates was described in great detail, in various situations. The following chapter will test our implementation with a simulation study, to make sure that it indeed will be able to outperform a baseline method.

Chapter 4

Simulation study

4.1 Data generation procedure

The simulation study was conducted to test both the package and the method itself in a controlled environment. Our datasets were generated out of ten different models for the target variable. Logistic regression was used for response model with 6 sets of parameters to generate 3 levels of response for each model. For each of the ten models, ten datasets with $N = 10000$ were sampled, but thirty were tested, since introducing missingness into the data created 3 sets for each of the sampled ten ($C = 10$).

Models 4.1 should represent a gradual decrease in linear relationship between the target variable Y and auxiliary X and was used before by Chen and Qin (1993), Jing Qin et al. (2002), and Robinson (1987).

$$\begin{aligned}
Y_1 &= X + \tfrac{ZX^{1/2}}{5}, \\
Y_2 &= X + 0.5X^2 + \tfrac{ZX^{1/2}}{5}, \\
Y_3 &= 1.5 + X + \tfrac{ZX^{1/2}}{5}, \\
Y_4 &= 3 + X - 0.05X^2 + \tfrac{ZX^{1/2}}{5}, \\
Y_5 &= 5 + \tfrac{ZX^{1/2}}{5},
\end{aligned} \tag{4.1}$$

where $Z \in N(0,1)$ and $X \in \chi^2(6)/2$

Models 4.2, on the other hand, suppose to represent gradual decrease in relationship between target variable Y and auxiliary variables X and Z that are available, as opposed to the hidden ones like H1 and H2, that will not be used for estimation.

$$
\begin{aligned}
Y_6 &= 1.5 + 0.5X + Z, \\
Y_7 &= (1.5 - 0.5H1) + 0.6(1.5 + 0.5X + Z), \\
Y_8 &= 0.6(H2 - H1) + 0.4 * (1.5 + 0.5X + Z), \\
Y_9 &= 0.7(H2 - H1) + 0.3(1.5 + 0.5X + Z), \\
Y_{10} &= 0.8(H2 - H1) + 0.2(1.5 + 0.5X + Z),
\end{aligned} \tag{4.2}
$$

where $H1 \in B(1,0.7)$ and $H2 \in \chi^2(8)/2.2$.

Models 4.1 and 4.2 are not enough to generate data that can be used to test our correction method. There should also be some missingness introduced for the target variable, where assumption 1.1 would hold. To do that the following sets of parameters 4.3 for the response mechanism were used. There are two sets of parameters for the same response rate, since for models 4.2 apart from X, Z is also taken into account.

$$
\begin{aligned}
&P(R = 1|\theta_{001} = 1, \theta_{101} = -0.33) = \tfrac{exp(\theta_{101}Y + \theta_{001})}{1 + exp(\theta_{101}Y + \theta_{001})} \sim 50\%, \\
&P(R = 1|\theta_{011} = 1, \theta_{111} = -0.5, \theta_{211} = 0.17) = \tfrac{exp(\theta_{211}X + \theta_{111}Y + \theta_{011})}{1 + exp(\theta_{211}X + \theta_{111}Y + \theta_{011})} \sim 50\%, \\
&P(R = 1|\theta_{002} = 2, \theta_{102} = -0.33) = \tfrac{exp(\theta_{102}Y + \theta_{002})}{1 + exp(\theta_{102}Y + \theta_{002})} \sim 72\%, \\
&P(R = 1|\theta_{012} = 2, \theta_{112} = -0.5, \theta_{212} = 0.17) = \tfrac{exp(\theta_{212}X + \theta_{112}Y + \theta_{012})}{1 + exp(\theta_{212}X + \theta_{112}Y + \theta_{012})} \sim 72\%, \\
&P(R = 1|\theta_{003} = 3, \theta_{103} = -0.33) = \tfrac{exp(\theta_{103}Y + \theta_{003})}{1 + exp(\theta_{103}Y + \theta_{003})} \sim 87\%, \\
&P(R = 1|\theta_{013} = 3, \theta_{113} = -0.5, \theta_{213} = 0.17) = \tfrac{exp(\theta_{213}X + \theta_{113}Y + \theta_{013})}{1 + exp(\theta_{213}X + \theta_{113}Y + \theta_{013})} \sim 87\%,
\end{aligned} \tag{4.3}
$$

Figures 4.1, 4.2 show densities for the generated target variable for each model and simulated level of response. For both sets of models, for lower response rate, distribution seem to be shifted leftward, which would suggest presence of a potential bias. The line in the center of each distribution is its median. The median can also be observed in the table 4.1 in greater detail.

More bias with increase in non-response can be seen in the models 4.1 with supposedly stronger linear relationship, although bias in the data, generated for model 2 is visually more prominent then model 1, which suppose to have a stronger linear relationship.

There is no such pattern for the models 4.2. Medians from the table 4.1, clearly show that for models 6-10, with exception of 7, appearance of hidden variable did not cause more bias.

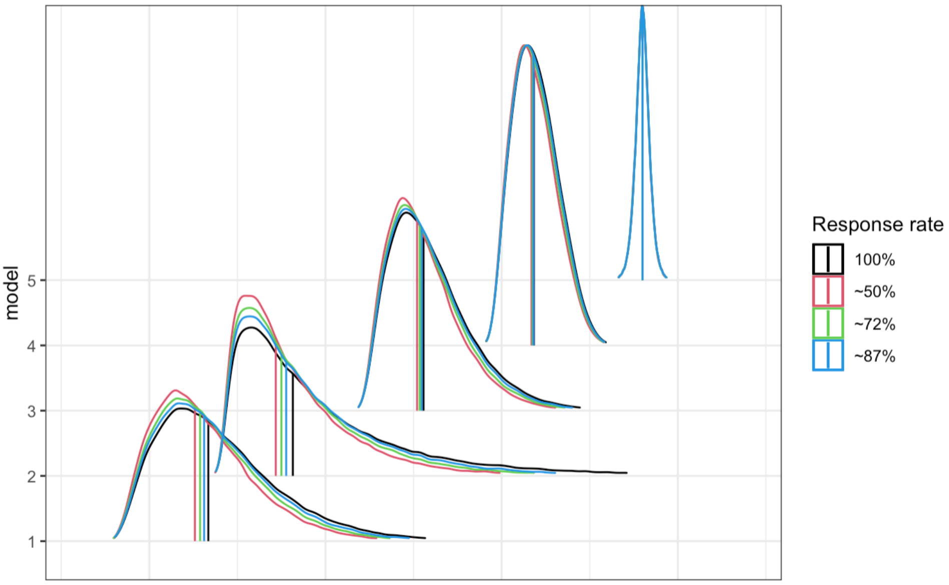

Figure 4.1. Comparison of densities for models 1-5 with variable response rates

Source: own elaboration.

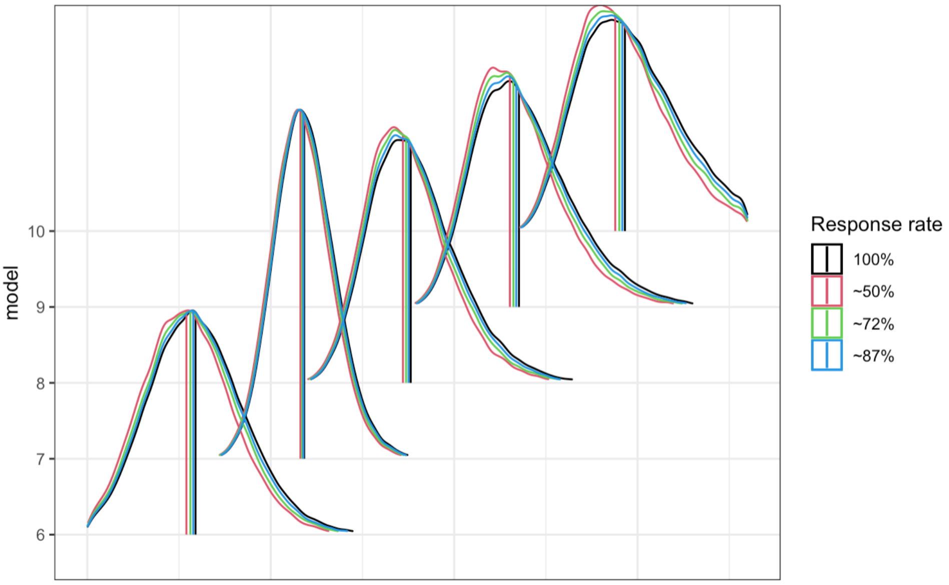

Figure 4.2. Comparison of densities for models 6-10 with variable response rates

Source: own elaboration.

4.2 Assessment of simulation study

To better illustrate the results of multiple simulations, the following metrics were used. All of them are basically dispersion for C samples that was done for each model.

Table 4.1. Comparison of median for models 1-10 with variable response rates

model	resp_100	resp_87	resp_72	resp_50	diff_resp_100_50
1	11.67	11.56	11.44	11.29	0.38
2	14.09	13.89	13.75	13.59	0.50
3	17.78	17.73	17.67	17.60	0.18
4	20.92	20.90	20.88	20.85	0.07
5	24.00	24.00	24.00	23.99	0.01
6	2.94	2.87	2.80	2.69	0.25
7	5.91	5.89	5.86	5.81	0.10
8	8.82	8.76	8.69	8.60	0.21
9	11.77	11.69	11.62	11.52	0.25
10	14.73	14.64	14.54	14.42	0.30

Source: own elaboration.

$$\text{Average Absolute Bias (AAB)} = \frac{\sum_{c=1}^{C} |\hat{\overline{Y}}_c - \overline{Y}_c|}{C} \tag{4.4}$$

$$\text{Standard Deviation (SD)} = \sqrt{\frac{\sum_{c=1}^{C} (\hat{\overline{Y}}_c - \overline{Y}_c)^2}{C}} \tag{4.5}$$

where $\hat{\overline{Y}}_c$ is an estimated population mean for the sample c and $\overline{Y}_c$ is the true population mean for the sample c.

Equation 4.4 is an average absolute bias (ABB) and reducing it is our goal, but it might not be the best measure to illustrate the performance of the method on the simulated data. To check some properties of the empirical likelihood based method, shown in the chapter 2, standard deviation of the estimators, calculated in three different ways was used. Equation 4.5 shows a standard deviation of the estimate from the true value for each sample.

4.3 Results

Before we analyse results of applying our package to correct simulated non-ignorable non-response in generated data, lets have a look at the bias caused by the naive method. For each of the models, it can be clearly seen in the figure 4.3 that the less response there was, the more biased results of the estimation were. It can also be seen, that for some models the impact of the non-response was not as severe as for the others.

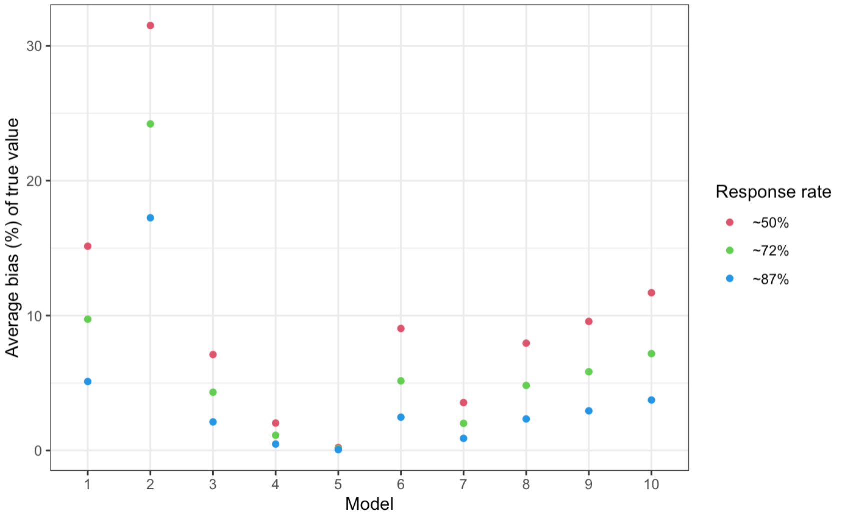

Figure 4.3. Relative bias of the naive method (simple mean) by response rates and different models

Source: own elaboration.

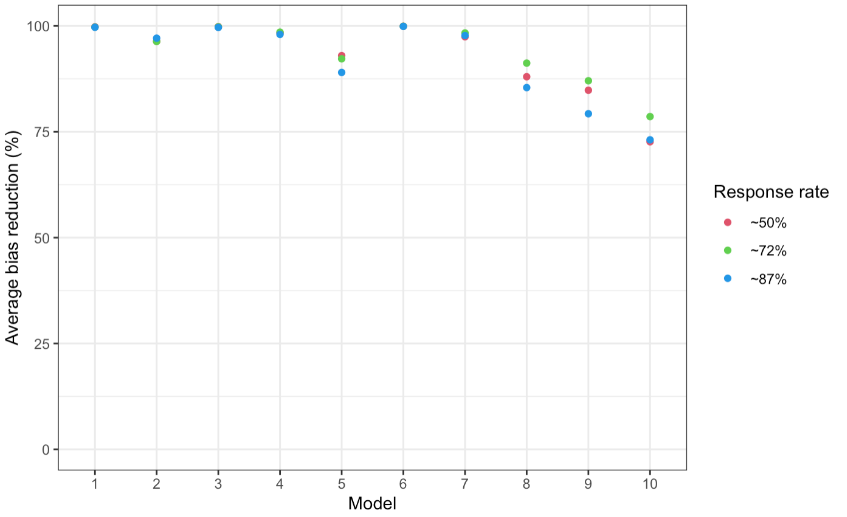

Figure 4.4. Relative improvements in reducing bias by EL over naive method for variable response rates and different models

Source: own elaboration.

After using empirical likelihood based method implemented in `nmar` package on our generated data, relative improvements in the bias reduction are noticeably smaller, when relationship between target variable and observed explanatory variables is weaker.

The plots above showed us average results, since for each of those dots ten samples were used. The following ones are meant to show how stable were those results in our simulation. Scatter plots for the models 1-5 show a very similar dynamic to the one we observed in the generated data 4.1. Models with stronger linear relationship had higher dispersion of results, except for the model 2 which had more dispersion the model 1. If the response rate had any effect, it is barely noticeable here.

Table 4.2. Bias relative to the bias at 50% response rate

model	naive.bias_50	diff_72_87	diff_50_72
1	0.45	0.31	0.36
2	0.95	0.20	0.23
3	0.21	0.31	0.39
4	0.06	0.33	0.44
5	0.01	0.30	0.43
6	0.27	0.30	0.43
7	0.10	0.32	0.45
8	0.24	0.30	0.41
9	0.28	0.28	0.43
10	0.34	0.27	0.52

Source: own elaboration.

Table 4.2 shows that even though for model 1-10 average bias reduction % was not constant, relatively to the bias at 50%, EL had a more or less constant edge over the naive method between 50 % & 72 % response rate and 72 % & 87 % response rate. Table include naive bias relative to the real value at 50 % response and bias reduction that EL has managed to perform divided by the naive bias at 50 % response rate. Table 4.3 shows calculation, visualized in the figure 4.3 and 4.4.

Table 4.3. Table of results for models 1-10 and variable response rates

nrm	model	Est	True	Naive	naive.bias	naive.bias.p	imp.p
1	1	3.00	3.00	2.55	0.45	0.15	1.00
2	1	3.00	3.00	2.71	0.29	0.10	1.00
3	1	3.00	3.00	2.85	0.15	0.05	1.00
1	2	2.97	3.00	2.06	0.95	0.32	0.96
2	2	2.97	3.00	2.27	0.73	0.24	0.96
3	2	2.99	3.00	2.48	0.52	0.17	0.97
1	3	3.00	3.00	2.79	0.21	0.07	1.00
2	3	3.00	3.00	2.87	0.13	0.04	1.00
3	3	3.00	3.00	2.94	0.06	0.02	1.00
1	4	2.97	2.97	2.91	0.06	0.02	0.98
2	4	2.97	2.97	2.94	0.03	0.01	0.99
3	4	2.97	2.97	2.96	0.01	0.00	0.98
1	5	3.00	3.00	2.99	0.01	0.00	0.93
2	5	3.00	3.00	3.00	0.00	0.00	0.92
3	5	3.00	3.00	3.00	0.00	0.00	0.89
1	6	3.00	3.00	2.73	0.27	0.09	1.00
2	6	3.00	3.00	2.85	0.15	0.05	1.00
3	6	3.00	3.00	2.93	0.07	0.02	1.00
1	7	2.95	2.95	2.85	0.10	0.04	0.97
2	7	2.95	2.95	2.89	0.06	0.02	0.98
3	7	2.95	2.95	2.93	0.03	0.01	0.98
1	8	2.95	2.96	2.72	0.24	0.08	0.88
2	8	2.95	2.96	2.82	0.14	0.05	0.91
3	8	2.95	2.96	2.89	0.07	0.02	0.85
1	9	2.95	2.95	2.67	0.28	0.10	0.85
2	9	2.94	2.95	2.78	0.17	0.06	0.87
3	9	2.94	2.95	2.86	0.09	0.03	0.79
1	10	2.96	2.94	2.60	0.34	0.12	0.73
2	10	2.92	2.94	2.73	0.21	0.07	0.79
3	10	2.92	2.94	2.83	0.11	0.04	0.73

Source: own elaboration.

Plots 4.5 show dispersion in the estimated done by EL with a use of a scatter plot, where to encompass all the relevant response rates, from estimates for 50 % response rate 0.2 was subtracted and for 87 % added, but the actual estimates are around value of 3 for all the relevant response rates.

For models 6-10 the figure 4.5 tell a different story. The model 7 does not break a pattern here and it can be said the less model is dependent on the significant unobserved variables, the less is a dispersion and lower response rates do have higher dispersion.

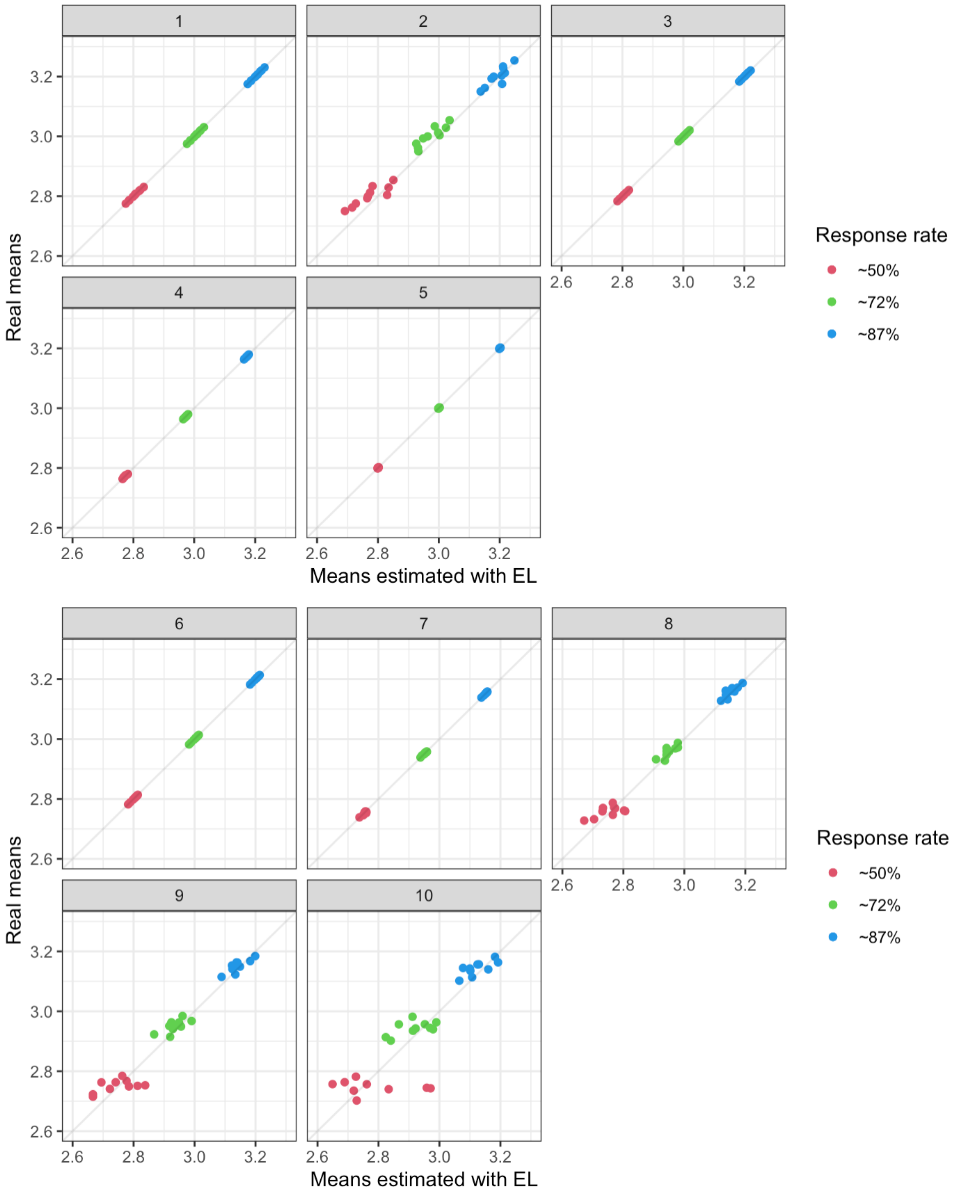

Figure 4.5. Comparison of dispersion in EL results for models 1-5 (top) and 6-10 (bottom) and variable response rates

Source: own elaboration.

Conclutions

Missing data is problem and a very particular subset of this called non-ignorable non-response was covered briefly, but hopefully comprehensible enough for the reader to understand and appreciate the complexity of the topic.

Second chapter the thesis as covered empirical likelihood based method that we implemented in our up-and-coming package, called `nmar`. More about the package itself, some of its functionality and general use cases were presented in the chapter 3.

Chapter 4 was meant to show how the method performs in different simulated scenarios. Where some strange thing were uncovered in the results, but nothing that can be explained.

The lower impact of the non-response for the naive method can be explained by the lower variability the target had in the those models, and also lower relative bias all together. The bias increase is very similar relatively to the lower response rate, but for obvious reasons is variable in absolute terms, even when the metric is relative, it is not relative to the lower response rate value.

Generated data, where gradual decrease in linear relationship and dependence on observable variables was simulated did not always have such gradual decrease in bias and its reduction as one might expect.

The explanations for that is simple. The linearity of the model is a lot less important in our simulation, then how much the target variable depends on the explanatory variables that we do observe versus some constant or other variable that we do not observe. To some extent this is also the case with the empirical likelihood based method itself, as can be seen in figure 4.4 for the model 5.

The response rate does effect the results only as variability of the model is explained by the unobserved variables, which is clearly seen in the top figure **??** than in the bottom figure **??**. This could be explained by the growing uncertainty that the lower response rate introduces, when variability of the target variable cannot be sufficiently explained the auxiliaries at hand.

The empirical likelihood based as a method for correcting non-ignorable non-response bias was shown here in a context of a very simple simulation, so the results should not be taken as conclusive by any means. More research is need, and we hope that our package will be of use to make it faster and less tedious then it might have been otherwise.

Bibliography

Beręsewicz, M. (2016). Internet data sources for real estate market statistics, 9–28. online.

Biffignandi, S., & Bethlehem, J. G. (2011). Handbook of Web Surveys, 59–140.

Brakel, J., & Bethlehem, J. (2008). Model-based estimation for official statistics.

Brown, C. (2018). *formula.tools: Programmatic Utilities for Manipulating Formulas, Expressions, Calls, Assignments and Other R Objects* [R package version 1.7.1]. https://CRAN.R-project.org/package=formula.tools

Bücker, M. (2011). Statistische Modelle mit nicht-ignorierbar fehlender Zielgröße und Anwendung in der Reject Inference.

Buelens, B., Boonstra, H., Brakel, J., & Daas, P. (2012). Shifting paradigms in official statistics: From design-based to model-based to algorithmic inference. *Statistics Netherlands Discussion Paper*.

Buuren, S. (2018). *Flexible Imputation of Missing Data, Second Edition*. https://doi.org/10.1201/9780429492259

Chen, J., & Qin, J. (1993). Empirical likelihood estimation for finite populations and the effective usage of auxiliary information. *Biometrika*, *80*(1), 107–116. https://doi.org/10.1093/biomet/80.1.107

Chen, J. [Jiahua], Variyath, A. M., & Abraham, B. (2008). Adjusted Empirical Likelihood and Its Properties. *Journal of Computational and Graphical Statistics*, *17*(2), 426–443. Retrieved October 4, 2022, from http://www.jstor.org/stable/27594315

Cheung, K., ten Klooster, P., Smit, C., de Vries, H., & Pieterse, M. (2017). The impact of non-response bias due to sampling in public health studies: A comparison of voluntary versus mandatory recruitment in a Dutch national survey on adolescent health. *BMC Public Health*, *17*. https://doi.org/10.1186/s12889-017-4189-8

Citro, C. (2014). From multiple modes for surveys to multiple data sources for estimates. *Survey Methodology*, *40*, 147–152.

Clausen, A., & Sokol, S. (2020). *Deriv: R-based Symbolic Differentiation* [Deriv package version 4.1]. https://CRAN.R-project.org/package=Deriv

Corporation, M., & Weston, S. (2022). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package* [R package version 1.0.17]. https://CRAN.R-project.org/package=doParallel

Enders, C. K. (2022). *Applied missing data analysis*. Guilford Publications.

Korkeila, K., Suominen, S., Ahvenainen, J., Ojanlatva, A., Rautava, P., Helenius, H., & Koskenvuo, M. (2001). Non-response and related factors in a nation-wide health survey. *European Journal of Epidemiology*, *17*(11), 991–999. https://doi.org/10.1023/A:1020016922473

Leeper, T. J. (2019). Where Have the Respondents Gone? Perhaps We Ate Them All. *Public Opinion Quarterly*, *83*(S1), 280–288. https://doi.org/10.1093/poq/nfz010

Little, R., & Rubin, D. (1987). *Statistical analysis with missing data*. Wiley.

Lundström, S., & Särndal, C.-E. (1999). Calibration as a standard method for treatment of nonresponse. *Journal of official statistics*, *15*(2), 305.

Manski, C. F. (2016). Credible interval estimates for official statistics with survey nonresponse [Innovations in Measurement in Economics and Econometrics]. *Journal of Econometrics*, *191*(2), 293–301. https://doi.org/https://doi.org/10.1016/j.jeconom.2015.12.002

McMullen, Q. (2001). How to Represent Missing Data: Special Missing Values vs. 999999999. *Westat, Rockville MD*.

Microsoft & Weston, S. (2022). *foreach: Provides Foreach Looping Construct* [R package version 1.5.2]. https://CRAN.R-project.org/package=foreach

Owen, A. (1988). Empirical likelihood ratio confidence intervals for a single functional. *Biometrika*, *75*(2), 237–249. https://doi.org/10.1093/biomet/75.2.237

Owen, A. (1990). Empirical Likelihood Ratio Confidence Regions. *The Annals of Statistics*, *18*(1), 90–120. https://doi.org/10.1214/aos/1176347494

Owen, A. (2001). Empirical Likelihood. https://doi.org/10.1201/9781420036152

Peytchev, A. (2013). Consequences of survey nonresponse. *The ANNALS of the American Academy of Political and Social Science*, *645*(1), 88–111.

Qin, J. [Jing], Leung, D., & Shao, J. (2002). Estimation with Survey Data under Nonignorable Nonresponse or Informative Sampling. *Journal of the American Statistical Association*, *97*(457), 193–200. http://www.jstor.org/stable/3085774

Robinson, J. (1987). Conditioning Ratio Estimates Under Simple Random Sampling. *Journal of the American Statistical Association*, *82*(399), 826–831. Retrieved October 9, 2022, from http://www.jstor.org/stable/2288792

Sobczyk, M. (2007). Statystyka / (Wyd. 5 uzup.), 20–21.

Soetaert, K. (2009). *rootSolve: Nonlinear root finding, equilibrium and steady-state analysis of ordinary differential equations* [R package 1.6].

Szymkowiak, M. (2019). Podejście kalibracyjne w badaniach społeczno-ekonomicznych, 135–141.

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

Toepoel, V., & Schonlau, M. (2017). Dealing with nonresponse: Strategies to increase participation and methods for postsurvey adjustments. *Mathematical Population Studies*, *24*, 1–5. https://doi.org/10.1080/08898480.2017.1299988

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, *45*(3), 1–67. https://doi.org/10.18637/jss.v045.i03

Vandenplas, C., Beullens, K., Loosveldt, G., & Stoop, I. (2018). Response Rates in the European Social Survey: Increasing, Decreasing, or a Matter of Fieldwork Efforts? https://doi.org/10.13094/SMIF-2018-00003

Wickham, H., Hesselberth, J., & Salmon, M. (2022). *pkgdown: Make Static HTML Documentation for a Package* [R package version 2.0.4]. https://CRAN.R-project.org/package=pkgdown

Yan, T., & Curtin, R. (2010). The Relation Between Unit Nonresponse and Item Nonresponse: A Response Continuum Perspective. *International Journal of Public Opinion Research*, *22*(4), 535–551. https://doi.org/10.1093/ijpor/edq037

List of Tables

List of Figures

List of R codes