Eötvös Loránd University

Faculty of Social Sciences

MASTER'S THESIS

Meta-analysis of missing data handling methods with text-mining

Supervisor:

Kmetty, Zoltán

Submitted by:

Boros, Krisztián

FCS65I

Survey Statistics

May 2020

CONTENTS

1. INTRODUCTION 4

2. OVERVIEW OF MISSING DATA HANDLING METHODS 6

2.1. ASSUMPTIONS AND MECHANISMS OF MISSINGNESS 6

2.1.1. Missing Completely at Random (MCAR) 7

2.1.2. Missing at Random (MAR) 8

2.1.3. Missing Not at Random (MNAR) 8

2.2. MISSING DATA HANDLING METHODS 9

2.2.1. Deletion 9

2.2.2. Basic Imputations 10

2.2.3. Advanced methods 11

2.3. PREVIOUS RESULTS ON MISSING DATA HANDLING PRACTICES 16

3. METHODOLOGY 17

3.1. DATA COLLECTION 18

3.1.1. Data sources 18

3.1.2. Web-scraping 20

3.1.3. Data cleaning 24

3.2. TEXT-MINING 28

3.2.1. Vector space representation 33

4. RESULTS 35

4.1. DESCRIPTIVE STATISTICS 35

4.2. LOGISTIC REGRESSION MODELS 42

5. LIMITATIONS OF THE USED METHODS 46

5.1. TOTAL SURVEY ERROR 46

5.1.1. Coverage 46

5.1.2. Measurement error 47

5.1.3. Processing error and extraction problems 47

5.2. FURTHER CONSIDERATIONS 49

6. CONCLUSION 50

7. BIBLIOGRAPHY 52

8. APPENDIX 55

8.1. KEYWORDS FOR CLASSIFICATION 55
8.1.1. Data cleaning: keywords to create weight vectors 55
8.1.2. Decision tree 55

1. Introduction

Missing data appears as an immanent part of every quantitative research, considering that it is very unlikely that we gather every piece of the desired data during research. In the course of the past fifty years, researchers have developed several methods to handle and manage information loss resulting from the incompleteness of data. The most sophisticated state-of-the-art methods are the Expectation-Maximization algorithm (EM), the Full-Information Maximum Likelihood estimation (FIML), and Multiple Imputation (MI).

Even though the aforementioned procedures provide an asymptotically unbiased and efficient estimate with more or less the same restrictions as if it was conducted with more prevalent techniques (T. D. Little et al. 2016; Dong and Peng 2013; Graham, Cumsille, and Shevock 2013; Enders 2010; Schafer and Graham 2002), researchers tend to use simpler solutions, e.g. listwise-, pairwise deletion or mean imputation, primarily based on implementation simplicity.

By simplicity, we mean that applying such a solution does not require a deep understanding of the underlying mathematical principles, consequently, it becomes more likely for a popular software program to contain a version of them. Several studies were conducted on the meta-analysis of handling incomplete data, deriving data predominantly from social, psychological and educational research fields (Roth 1994; Peugh and Enders 2004; Wood, White, and Thompson 2004; Bodner 2006; Peng et al. 2006; Jeličić, Phelps, and Lerner 2009; Bell et al. 2014; Cheema 2014). However, these papers merely use a small fraction of the published scientific articles in their field of research; the review and examination of such an extensive amount of papers would be a time-consuming and cumbersome attempt.[1] But is there a more comprehensive and time-saving method to conduct a similar analysis?

The aim of this work is twofold. On one hand, we introduce a text-mining approach to collect and analyze papers while pointing out the advantages and disadvantages of this particular

[1] Some paper draws a simple random sample from the available population, some just narrows down the focus to the empirical studies.

approach. On the other hand, we try to examine the possible trends of the missing data handling methods across years and scientific fields.

In the first part of the thesis we try to give a comprehensive overview of the missing data handling methods from the basic deletion methods to the more involved imputation and maximum likelihood estimation models, and briefly demonstrate the possible applications and limitations of them. After that, we present and describe the applied text-mining and web-scraping methods, and the sampling design. In the end, the results of the analysis will be presented.

Although we are certain that this text-mining approach cannot entirely substitute the human resources used in the course of a meta-analysis, we assume that it can provide a different perspective for future research.

2. Overview of missing data handling methods

The foundations of the missing data paradigm used nowadays were established in the 1976 paper "Inference and missing data" by Donald B. Rubin. In his paper, Rubin formulated a general framework of the perception of missing data, emphasizing the role of the underlying mechanisms that cause the missingness in the observations.

If a researcher encounters a considerable number of missing data during the data processing period, then it is strongly recommended to spend some time exploring the possible process that causes the missingness. Without doing so, the estimations of the desired parameter(s) could be biased. In this chapter, we are going to give a brief overview and explanation of these mechanisms and present the current state of the paradigm using mostly the work of Dong and Peng (2013), Enders (2010), Graham et al. (2013), and R. J. A. Little and Rubin (2019).

2.1. Assumptions and mechanisms of missingness

Let's say that there exists a fully complete $\mathrm{n} \times \mathrm{p}$ dataset with no missing values and denote it with $\mathrm{Y}_{\mathrm{com}}$; n is the number of observations and p is the number of variables.[2] Such a dataset would contain every attainable information about a given phenomenon. However, in real life, it is more plausible, that a part of the information is missing or cannot be observed adequately. For this practical reason, we divide the hypothetical dataset of ours to a mutually exclusive observed ($\mathrm{Y}_{\mathrm{obs}}$) and missing ($\mathrm{Y}_{\mathrm{mis}}$) part. The former represents the information that is observed (or "real") and the latter the unobserved (or "hypothetical") information. For example, if we gather data about socioeconomic status or wealth, then the upper-tail of the income distribution could be missing due to non-response (Kośny 2019). That is the missing or unobserved part of the ideal, complete dataset. Nevertheless, we have no other option than to use the observed part of the data and make conjectures about the unobserved part.

[2] Because of consistency, throughout the thesis we use the notation of Enders (2010)

For a better understanding of the missing fraction of the data, Rubin (1976) introduced the so-called missing data indicator variable R, which takes the value of *0*, if a value of a given variable is missing, and *1* if it is not. If we have an $\mathrm{n} \times \mathrm{p}$ data-matrix, then the indicator variables are concatenated into an $\mathrm{n} \times \mathrm{p}$ Boolean matrix $\boldsymbol{R}$, where *n* is the number of observations and *p* is the number of variables. Each R variable has a theoretically unknown probability distribution which determines the occurrences[3] of the missing data and with the help of the association between the indicator variable and the observed data, we can formalize the following mechanisms.

2.1.1. Missing Completely at Random (MCAR)

The best-case scenario – but a quite unlikely one – is that the missing part of the acquired data is a random subsample of it, therefore the inference made from the incomplete dataset will be similar to the one made from the complete (the standard errors are going to be higher).

Formally expressed:

$$P(R|Y_{obs}, Y_{mis}, \phi) = P(R|\phi), \qquad \phi \in \Phi \tag{2.1}$$

Where P stands for probability distribution, R is the missing data indicator variable, $\mathrm{Y_{obs}}$ is the observed part, $\mathrm{Y_{mis}}$ is the missing part of the data, and ϕ is an arbitrary parameter that describes the association between R and the data. (2.1) expresses that the missingness is independent of the data, i.e. neither the observed values, nor the unobserved values are related to the missingness in the dataset, but there is still an unknown parameter that defines the distribution of R (Enders 2010:12).

[3] There is a more visual representation of missing data, called „missing data pattern", which basically shows where the 0-s and 1-s are located in the dataset and creates patterns of it. More information about this topic can be found in Little and Rubin (2019)

2.1.2. Missing at Random (MAR)

The name “missing at random” could be deceiving, because the missingness in a variable will be random only if we control for the variables correlate with it. This case is closer to reality, since it is very likely that the variables in a dataset are somehow related to each other. Thus, the main issue in a MAR case is to find the variables which are associated with the missingness.

More formally:

$$P(R|Y_{obs}, Y_{mis}, \phi) = P(R|Y_{obs}, \phi), \qquad \phi \in \Phi \tag{2.2}$$

Where P stands for probability distribution, R is the missing data indicator variable, $\mathrm{Y_{obs}}$ is the observed part, $\mathrm{Y_{mis}}$ is the missing part of the data, and ϕ is an arbitrary parameter that describes the association between R and the data. (2.2) says that the distribution of R is dependent on the observed part of the data and the immanent parameter of the missingness (Enders 2010:11).

2.1.3. Missing Not at Random (MNAR)

The most complicated and inconvenient case is the MNAR, where the missingness is related both partitions of the hypothetical, complete dataset. Therefore, it is a displeasing task to discover or identify the occurrence of MNAR.

$$P(R|Y_{obs}, Y_{mis}, \phi) = P(R|Y_{obs}, Y_{mis}, \phi), \qquad \phi \in \Phi \tag{2.3}$$

Where P stands for probability distribution, R is the missing data indicator variable, $\mathrm{Y_{obs}}$ is the observed part, $\mathrm{Y_{mis}}$ is the missing part of the data, and ϕ is an arbitrary parameter that describes the association between R and the data. (2.3) shows that the distribution of R remains the same because it depends on “everything” (Enders 2010:11).

2.2. Missing data handling methods

The development of missing data handling methods led to more complex and versatile techniques created to access and/or reduce the missing information. These techniques have a well-traceable evolution, but their evolving speed differs in theoretical and applied research. The main reason behind this lag could be the complexity of the modern, cutting-edge techniques (Enders 2010; R. J. A. Little and Rubin 2019; T. D. Little et al. 2016; Schafer and Graham 2002) or just the slowness of the diffusion of scientific knowledge. Nevertheless, it is important to review and show the deficiency of the easy-to-use and often uneligible methods so we can learn from them.

2.2.1. Deletion

By far, listwise- and pairwise deletions (also called Complete Case Analysis and Available Case Analysis) have been the most popular and widely-used missing data handling methods until the turn of the millennium, thanks to their user-friendly applications and simplicity (Schafer and Graham 2002). Unfortunately, these methods require to assume an MCAR mechanism, which is rare to occur in empirical research.

2.2.1.1. Listwise deletion

In the case of listwise deletion, if an observation contains even one missing value in some variable, then the whole row of the data matrix corresponding to that case will be deleted. On an intuitive level, it is quite trivial that if we delete an entire case from our dataset, then it causes information loss. Beside information loss – assuming that MCAR is not fulfilled – the parameter estimates are likely to be biased and there is a serious loss of statistical power. Graham et al. (2013) argue that the loss of statistical power is a bigger problem than the biasedness of the parameters, because if the percentage of the missing data is relatively small, then the estimations will be not as inaccurate as we would suspect (Graham and Donaldson 1993).

2.2.1.2. Pairwise deletion

Pairwise deletion seems to account for the problem of information loss by preserving cases that contain missing data. This method eliminates an observation only if it contains missing data in a

variable that is sufficient when calculating a certain parameter. The most common example is the computation of covariance and correlation matrices: we use all the available cases which have complete pairs in the required variables. For this reason, the usual problem is that the resulting correlation matrix is not positive definite. Despite the fact, that the pairwise deletion method uses a different, more "complete" subsample of the dataset, it can result in biased estimates as well as the listwise method and overcomplicate the calculation of standard errors.

2.2.2. Basic Imputations

The next, widely-used group of missing data handling methods are the imputations or substitutions of given values to the "holes" of the dataset. These single-imputation methods could be tempting because unlike deletions, they produce seemingly complete datasets and keep the values which deletion methods would omit. The following techniques are sometimes overlapping; hence the categorizations are not strict.

2.2.2.1. Mean, median and modus imputation

One of the firstly implemented solutions to fill the missing parts of a dataset is using the unconditional mean, median, or modus. The aggregated values can be calculated on different levels e.g. group mean, total median, and it is quite easy to implement this method. As the above-mentioned deletion techniques, these imputations have serious drawbacks as well: the variance calculated with the imputed scores will be too low (since they are all the same), and in the case of for example group mean imputation, the within-group variance is going to be lower and the between-group variance is going to be higher.

2.2.2.2. Regression imputation

A more effective, but still not the most efficient technique is to imputing the conditional mean to the missing places. This regression-based method[4] uses the information from other variables, but results in a too high correlation between the imputed values and overestimates the multiple

[4] In this matter, one can use a deterministic or a stochastic regression model: there is an additional noise in the latter and produces better variances.

correlation coefficient (R^2). Moreover, the values from the imputation could be hard to interpret, especially in the case of a notion measured on a Likert-scale.

2.2.2.3. Hot- and cold-deck imputation

The deck-imputation methods are mostly used in official statistics and only the source of substituted values distinguishes them from each other. "Hot" refers to the dataset which is currently used and "cold" refers to another, external dataset. The idea behind the technique is to substitute a missing value with a similar one. For example, it can be performed with the "k-nearest neighbor" algorithm, i.e. drawing randomly an observation from a subsample containing similar (and complete) observations in certain characteristics (e.g. demographics). (R. J. A. Little and Rubin 2019:76).

2.2.3. Advanced methods

In this section, we introduce the cutting-edge techniques to handle missing data and provide a more substantial description of them. As we will see, these techniques are more involved than the previous ones, although there is a fundamental connection in the underlying mathematics.

2.2.3.1. Expectation-Maximization algorithm (EM)

The Expectation-Maximization algorithm is an iterative method formulated and generalized by Dempster, Laird, and Rubin (1977). There are many possible applications of the algorithm (factor analysis, clustering)[5], but it was originally designed to give precise estimates from incomplete data. Heuristically, the algorithm consists of two steps:

1) *Expectation-step (E)*
2) *Maximization-step (M)*

In the E-step, using the available data (Y_{obs}) we estimate sufficient statistics, and in the M-step we use these statistics to give a Maximum Likelihood estimation of the desired parameters. It is a common misconception that the EM-algorithm imputes values to the missing places in the E-step. As a matter of fact, in the E-step we calculate the expectation of the sufficient statistics

[5] For further variations of the algorithm, see Little and Rubin (2019:187)

which contains all required information of the parameter, therefore from the missing part of the data ($\mathrm{Y_{mis}}$). The imputation is only happening at an abstract, informational level, but not on a technical level.

In multiple examples, the algorithm is presented by assuming that the data came from an exponential family because, with this premise, the computations will be simpler. Nevertheless, here we describe the most general form of the algorithm based on its definition in the above-mentioned paper of Dempster et al. (1977). At first, we need to introduce a function that calculates the conditional expectation of the log-likelihood function of the current parameter:

$$Q(\phi'|\phi) = E_{Y_{obs},\phi}\left[log\left(f(Y_{com}|\phi')\right)\right] \qquad 2.4$$

Where ϕ' is the parameter to be estimated given the current parameter ϕ, $\mathrm{Y_{obs}}$ is the observed partition of $\mathrm{Y_{com}}$ i.e. the complete data. Then one iteration of the algorithm goes as follows:

1) *E-step: Calculating* $\mathrm{Q}(\phi'|\phi^{(\mathrm{p})})$
2) *M-step: Finding a parameter* $\phi^{(\mathrm{p+1})}$*, which maximizes* $\mathrm{Q}(\phi'|\phi^{(\mathrm{p})})$

Where *p* denotes the p^{th} iteration of the algorithm. The benefit of the EM-algorithm that it produces unbiased estimates of the parameters under MCAR and even under MAR because during the estimation we use the conditional expectation, therefore we include the information which is in other variables. The drawbacks of the EM are that the algorithm can be quite slow[6], and the fact that during the estimation, the algorithm does not provide standard errors of the parameter estimates.

2.2.3.2. Full-Information Maximum Likelihood (FIML)

The Full-information maximum likelihood approach is a widely used technique, especially in Structural Equation Modelling. It may seem redundant to use two techniques that are using ML-

[6] There are modifications of the algorithm that are faster than the original one. For details, see for example Matsuyama (2003).

estimates of the parameters, but there are two main differences between FIML and EM: FIML is not an iterative method like EM and in the case of FIML, multivariate normality of the data has to be assumed. The assumption of multivariate normality can be challenged, but in this section (and in the thesis) we do not discuss this topic. Rather we show how this assumption yields desirable properties of the method. The FIML achieves unbiased and efficient estimations of the parameters with a small modification of the multivariate normal log-likelihood function (T. D. Little et al. 2016). The default log-likelihood function of the multivariate normal distribution can be written as follows:

$$l(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\boldsymbol{Y}) = \sum_{i=1}^{N} \left[-\frac{p}{2} ln(2\pi) - \frac{1}{2} ln|\Sigma| - \frac{1}{2} (\boldsymbol{Y}_i - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{Y}_i - \boldsymbol{\mu}) \right] \quad 2.5$$

Where $\boldsymbol{Y}$ is an $N \times p$ matrix of incomplete data (i.e. $\mathrm{Y_{obs}}$), $\boldsymbol{\mu}$ is the p-variate vector of unknown, population-level means of the variables in $\boldsymbol{Y}$, and $\boldsymbol{\Sigma}$ is the $p \times p$, unknown, population-level covariance matrix of the variables in $\boldsymbol{Y}$.

In the case of FIML, the equation is modified to:

$$l_{FIML}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\boldsymbol{Y}) = \sum_{i=1}^{N} \left[-\frac{p}{2} ln(2\pi) - \frac{1}{2} ln|\Sigma_i| - \frac{1}{2} (\boldsymbol{Y}_i - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\boldsymbol{Y}_i - \boldsymbol{\mu}_i) \right] \quad 2.6$$

The difference between equation (2.5) and (2.6) is the additional subscript of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, which means that these parameters will be calculated for each i^{th} observation. Thus, (2.5) computes the final aggregated parameters based on every accessible score. The disadvantage of FIML is that it was designed primarily for survey-type research and SEM, so the implementations of different research designs could be difficult.

2.2.3.3. Multiple Imputation (MI)

As we have shown previously, single imputation methods have several drawbacks particularly in the calculation of variances and standard errors. Multiple Imputation (MI) solves this problem

with the iteration and pooling of single imputations. In a nutshell, the estimation of a parameter with MI works in the following way:

1) Single imputation of missing values
2) Repeating 1) *M* times and fit an arbitrary model in every iteration to estimate a parameter $\hat{Q}^{(m)}$
3) Pool the parameters using Rubin's rules

The single imputation is usually a regression estimate of the missing data, but it can be hot-deck or other substitution as well. After the imputation of the missing data, we repeat this procedure M times to create a distribution from the substituted values.[7] The iteration of imputations and estimations permits the computation of between- and within imputation variance, which plays an important role in accessing the uncertainty of MI. At step 2), we end up with point estimates $\widehat{\mathrm{Q}}^{(1)}, \widehat{\mathrm{Q}}^{(2)}, \ldots, \widehat{\mathrm{Q}}^{(\mathrm{m})}, \ldots, \widehat{\mathrm{Q}}^{(\mathrm{M})}$ from the repeated imputations. The so-called Rubin's rules contain simple formulas to calculate the overall multiple imputation point estimate $\overline{\mathrm{Q}}$, which equals the arithmetic mean of the estimates:

$$\bar{Q} = \frac{1}{M} \sum_{m=1}^{M} \hat{Q}^{(m)} \qquad 2.7$$

where $\overline{\mathrm{Q}}$ is the pooled point estimate, $\widehat{\mathrm{Q}}^{(\mathrm{m})}$ is the parameter estimate from the m^{th} imputation, and M is the number of imputations. To calculate the overall variance or *total sampling variance*, we have to factorize it into between- and within-imputation variance. The pooled within-imputation variance is the arithmetic mean of the M estimated parameters:

$$\bar{U} = \frac{1}{M} \sum_{m=1}^{M} SE^2_{\hat{Q}^{(m)}} \qquad 2.8$$

[7] The value of *M* is up to the researcher, but in general, it is not greater than 100.

where U is the pooled within-imputation variance, $\mathrm{SE}^2_{\widehat{\mathrm{Q}}(\mathrm{m})}$ is the squared standard error for the parameter estimate from the m^{th} imputation, and M is the number of imputations. The between-imputation variance measures the variability of the imputed values among all M imputations:

$$B = \frac{1}{M-1}\sum_{m=1}^{M}(\hat{Q}^{(m)} - \bar{Q})^2 \qquad 2.9$$

where B is the between-imputation variance, $\widehat{\mathrm{Q}}^{(\mathrm{m})}$ is the parameter estimate from the m^{th} imputation, $\overline{\mathrm{Q}}$ is the multiple imputation point estimate, and M is the number of imputations. Thus, the total sampling variance is computed as the weighted sum of the between and within variance:

$$T = \bar{U} + (1 + M^{-1})B \approx \bar{U} + B \text{ } \textit{(if M is large)} \qquad 2.10$$

where T is the total sampling variance, $\overline{\mathrm{U}}$ is the pooled within-imputation variance, B is the between-imputation variance, and M is the number of imputations.

By and large, the modern missing data handling methods give unbiased and – in the case of MI – efficient estimates of the parameters, therefore the usage of them is highly recommended. As the rule of thumb goes, the modern approaches are at least as good as the old methods under MCAR, but they have other desirable properties which endorse the applications of such methods. Furthermore, the development and refinement of these modern techniques are happening right now in the scientific communities and there are lots of notable improvements (Tomita, Fujisawa, and Henmi 2018; Seffens, Evans, and Taylor Minority Health-Grid Network And Herman 2015; Ichikawa et al. 2019; Cham et al. 2017; Takahashi, Iwasaki, and Tsubaki 2017; van Buuren 2019; Takahashi 2017).

2.3. Previous results on missing data handling practices

Before heading towards the main part of the thesis, we review the previous research results regarding missing data handling practices. As we noted beforehand, most of the works were conducted in the social, psychological, and educational research areas, probably because of the discipline-specific origins of the missing data paradigm and the applications in the survey-type designs. In every paper, the researchers choose a journal or a set of journals and draw a simple random sample from the papers that were published at a certain time interval. A well-distinguishable and important demarcation line of the missing data handling practices in the abovementioned disciplines was the statement of the Task Force on Statistical Inference (TFSI) in which methodological standards were laid down concerning missing data handling (Wilkinson and Task Force on Statistical Inference 1999). The majority of the meta-analyses were conducted after this statement to examine the effect of it, for example by comparing papers from the before and after periods (Peugh and Enders 2004) or just describe the trends of practices (Peng et al. 2006; Jeličić, Phelps, and Lerner 2009). Despite the fact that the findings mostly pertain to social sciences (Roth 1994; Bodner 2006; Cheema 2014), further results from medical research are consistent with those mentioned above (Wood, White, and Thompson 2004; Fielding et al. 2008; Bell et al. 2014).

A common result of these meta-analyses is that of listwise- and pairwise-deletion and single imputation methods are used widespread. Peugh and Enders (2004) point out, however, that the identification of the deletion methods in the selected papers is a difficult task because they lack the documentation of such methods.[8] Therefore, we must keep in mind that the reported frequencies are not entirely reliable. After the TFSI statement, there was a serious improvement in the documentation of missing data handling methods and a slight growing trend in the adaptation of “modern” techniques.

[8] The applied techniques for finding these methods will be utilized later in this thesis.

3. Methodology

Because of the methodological nature of the thesis, this third chapter will be by far the most emphasized and detailed part of it. In this chapter, we introduce the text-mining approach we mentioned in the beginning. This section aims to guide the reader through the whole process from data gathering to analysis. We start with a short introduction of web-scraping techniques and then present the data collection. After building the *corpus*, we continue with a tedious but crucial task: data cleaning. As the data are prepared, we extract the relevant information with text-mining. In every step, we will point out the advantages and disadvantages of the methods and at the *Limitations of the used methods* section, we will give a more exhaustive discussion about the limitations and important considerations that come up through the process. At the end of the chapter, a possible analytic tool will be presented, which is the vector-space representation of words.

Most parts of the work were conducted in Python (with Jupyter Notebook IDE).[9] The reason behind this choice is that there is a variety of very-well implemented text-mining, web-scraping, natural language processing (hereafter NLP), and visualizing packages and tools in this language. Furthermore, the community around the Python language is extremely active, helpful, and up-to-date. All packages will be highlighted throughout the chapter like this: `name of the package`; the created script will be uploaded to GitHub[10] and can be accessed freely. If it is necessary for a better understanding, some code chunks will be inserted into the text.

One more thing has to be mentioned regarding the scripts: even though there are comprehensive learning materials for text-mining methods (Bengfort, Bilbro, and Ojeda 2018; Sarkar 2019; Mitchell 2018), it is inevitable to use unorthodox sources such as stackowerflow.com or stackexchange.com in some special technical cases. The reliability of these sources can be questioned, but each site we use in this thesis is strictly controlled and monitored to ensure the quality of the solutions and answers. Henceforth, the code chunks and scripts based on a

[9] There were some cases, when we used R, because of our familiarity in the classification methods and modelling in this language. IDE=Integrated development environment.

[10] https://github.com/BiliBraker/Thesis

Stackoverflow or Stackexchange source are referenced with "(SO, *number of source*)" and "(SE, *number of source*)", respectively.

3.1. Data collection

A positive aspect of a text-mining approach is the acquisition of the required data, since there is no need for long preparations or surveying procedures; the data can be easily collected from a research room or even from home.[11] On the contrary, this type of data collection lacks the favorable and important features of the solid academic methodology, more specifically the representativeness of the gathered sample or the reliable origin of the data. Primarily, these data are retrieved with web-scraping or via an Application Program Interface (henceforth API) from various sources, but naturally, they can originate from public libraries, companies, and official institutions. The goal is to obtain data that are as structured, transformable, and useable as possible.

3.1.1. Data sources

We used three main sources for collecting scientific papers: Google Scholar, arXiv, and Jstor. In this short subsection, we give a brief description of these sites.

3.1.1.1. Google Scholar

It seems plausible to use Google Scholar for searching academic works, but in our case using this site turned out to be a dead end. Despite the fact that Google Scholar is a highly effective tool in searching scientific papers, several problems emerge when it comes to the acquisition of the data from these papers. First of all, Google does not support web-scraper robots because of the misusage of data and the possible overloading of the server. Instead, the company provides an API, which can be used to gather the desired data from the website, but this service does not include Scholar searches (SE, 1). Therefore, there is no ethical, legal way to scrape data from Google Scholar and even if it was, other issues seem to appear.[12] With a little work, the metadata

[11] However, one can argue that the simplicity of data collection can vary depending on the research objective.
[12] Of course, there are attempts to by-pass these restrictions.

(titles, authors, dates, etc.) could be extracted from the search results, but the content of the papers oftentimes not, since several journals require a subscription to access the papers.

3.1.1.2. arXiv

As the organization states: "arXiv is a free distribution service and an open archive for scholarly articles in the fields of physics, mathematics, computer science, quantitative biology, quantitative finance, statistics, electrical engineering and systems science, and economics. arXiv is a collaboratively funded, a community-supported resource founded by Paul Ginsparg in 1991 and maintained and operated by Cornell University."[13]

The main benefits of using arXiv to collect scientific papers that are freely accessible, the files and metadata are easily scrapeable. Using the available API[14], we scraped 953 papers from the site. It is worth noting, however, that the papers which were uploaded to arXiv are not necessarily peer-reviewed articles, thus the quality of the works is somewhat questionable.

3.1.1.3. Jstor

Just like arXiv, Jstor also allows researchers to freely access scholarly works, but in a greater volume: "JSTOR provides access to more than 12 million academic journal articles, books, and primary sources in 75 disciplines."[15] Nonetheless, there are several written documents which only available through a university proxy or subscription; in our case, such proxy was provided by Eötvös Loránd University. Furthermore, Jstor introduced a data providing service[16] that helps researchers obtain the necessary scholarly works. One can make a customized search query of the papers with various filters, then download the metadata, n-grams, and references up to 25000 items. There is a possibility to request full texts as well (with Optical Character Recognition), but it is more time consuming, because of the agreement and data protection procedures. All in all, using this opportunity resulted in receiving 33148 papers from Jstor's archive within 3 weeks.

[13] https://arxiv.org/about/
[14] https://arxiv.org/help/api
[15] https://about.jstor.org/
[16] https://www.jstor.org/dfr/about/dataset-services?cid=dsp_j_dfr_01_2018&utm_source=jstor&utm_medium=display&utm_campaign=home_right_jstor_dfr

3.1.2. Web-scraping

We have mentioned earlier that one can obtain textual data from several sources. This oftentimes requires web-scraping techniques, since the acquirement of these data usually happens via the internet. Furthermore, during data processing, one might encounter tasks that can be completed using web-scraping techniques very effectively.

Especially in the case of applied sciences, web-scraping is a convenient and powerful tool for obtaining a large amount of valuable data from the internet. In this section, we introduce the web-scraping methods that are used in this thesis and describe their technical details. For clarity, we briefly summarize the steps:

1) Specifying the URL that will be used as a starting point
2) Making a request with **`urllib`**
3) Transform the source code with **`BeautifulSoup`**
4) Find the relevant HTML/XML tags
5) Create a data matrix from the extracted data
6) Save the data matrix

The first step determines the sample that will be used throughout the research because we have to specify the keywords of the search. In the beginning, we searched for “missing data” in Google Scholar and set the date interval to 1999-2019. Then, we inspected the HTML source to identify the relevant tags from the information of papers that will be gathered. From this search query, the title, the author, the date of publication, and the link of the paper are important. This information can be easily collected by reference to the tags that encompass them (Figure 1). When one page of the query is done, an automated web-scraper robot can be made to loop over each page and extract the HTML source from them. The first problem emerges at this point, because of Google’s data protection and anti-scraper-robot policy: Google disables the IP-address if it requests too much data too fast. The second problem stems from the fact that even if we “own” the links of the papers, we cannot download them, because the query contains journals

that require a subscription for the scientific materials. The scrape of Google Scholar ended unsuccessfully.

```
<div class="gs_ri">
    <h3 class="gs_rt" ontouchstart="gs_evt_dsp(event)">
        <span class="gs_ctc">
            <span class="gs_ct1">[BOOK]</span>
            <span class="gs_ct2">[B]</span>
        </span>
        <a id="FVWW9ItwsgQJ" href="https://books.google.com/books?hl=en&amp;lr
            =&amp;id=LJB2AwAAQBAJ&amp;oi=fnd&amp;pg=PP1&amp;dq=%22missing
            +data%22&amp;ots=RnyNVSwG7d&amp;sig=zz-jKiJ-7uk5eN5Zat6KMbshm_g"
            data-clk="hl=en&amp;sa=T&amp;ct=res&amp;cd=0&amp;d
            =3384566684124664 53&amp;ei=NrapXpAfgbOYAbK_oPAL" data-clk-atid
            ="FVWW9ItwsgQJ">
            <b>Missing data</b>
        </a>
    </h3>
    <div class="gs_a">
        <a href="/citations?user=RFRPlFoAAAAJ&amp;hl=en&amp;oi=sra">PD Allison
            </a> - 2001 - books.google.com
    </div>
    <div class="gs_rs">Using numerous examples and practical tips, this book
        offers a nontechnical explanation of
        <br>the standard methods for missing data (such as listwise or casewis
            deletion) as well as two
            <br>newer (and, better) methods, maximum likelihood and multiple
                imputation. Anyone who has …
            </div>
            <div class="gs_fl">
                <a href="javascript:void(0)" class="gs_or_sav" title="Save" ro
                    ="button">
                    <svg viewBox="-1 0 17 16" class="gs_or_svg">
                        <path d="M8 11.57l3.824 2.308-1.015-4.35 3.379-2.926-4
                            .45-.378L8 2.122 6.261 6.224l-4.449.378 3.379 2.92
                            -1.015 4.35z"></path>
                    </svg>
                </a>
                <a href="javascript:void(0)" class="gs_or_cit gs_nph" title
                    ="Cite" role="button" aria-controls="gs_cit" aria-haspopup
                    ="true">
                    <svg viewBox="-1 0 17 16" class="gs_or_svg">
                        <path d="M1.5 3.5v5h2v.375L1.75 12.5h3L6.5 8.875V3.5zM
```

Figure 1 Html example (Google Scholar; 2020.03.15.; "missing data"; 1999-2019)

In the case of arXiv, these steps have turned out to be performable, since the organization provides an API to collect the data. After retrieving the metadata from a similar source code like Google Scholar's, the information of papers was organized into a data frame with the **pandas** package (Figure 3). With their links being available, we were able to start the downloading process (Figure 2), which lasted for a few hours, but in the end, 953 PDF files were obtained. It is worth mentioning that finding the appropriate tags in the HTML code can be tedious and time-consuming because oftentimes the necessary information is nested deep into the source code. In addition, the category of each paper can be found in the metadata.

```
from urllib.request import urlretrieve

def download_file(download_url,name):
    urlretrieve(download_url, 'c:/Users/soirk/Krisztian/Egyetem/Survey
Statisztika Msc/Szakdolgozat/arxiv_pdfs/'+name+'.pdf')
    for i in links_list:
    download_file(i,i[-7:])
    print(i[-7:])
```

Figure 2 The script for downloading papers from arXiv

id	title	date	category
0812.1615v1	Missing Data using Decision Forest and Computational Intelligence	2016	stat
1610.09075v2	Missing Data Imputation for Supervised Learning	2011	stat
1102.3851v1	Missing Data Imputation and Corrected Statistics for Large-Scale Behavioral Databases	2019	cs
1904.12413v1	A convolution recurrent autoencoder for spatio-temporal missing data imputation	2014	stat
1409.0895v1	Semiparametric Inference of the Complier Average Causal Effect with Nonignorable Missing Outcomes	2019	stat
1912.12894v1	On a simultaneous parameter inference and missing data imputation for nonstationary autoregressive models	2019	stat
1903.03630v1	Imputation estimators for unnormalized models with missing data	2007	cs
0704.3635v1	Rough Sets Computations to Impute Missing Data	2018	stat
1801.03583v2	Graphical Models for Processing Missing Data	2019	math
⋮	⋮	⋮	⋮

Figure 3 Data frame of scraped metadata from arXiv

The computationally easiest part of the web-scraping was the one with Jstor because, after the documentation of the data request and other arrangements, the organization sent us the complete metadata of the papers in XML format. Consequently, steps 1) and 2) have been omitted, and just like in the previous cases with the HTML code, the search and collection of relevant tags were performed. At the end of the task, we received the metadata, full text (in TXT format), as well as the uni-, bi-, trigrams of 33148 papers. From the metadata, in addition to the previous cases, the titles of the journals were extracted to identify the category of certain papers (Figure 3), as the categories were not embedded implicitly into the metadata. In the next section, we describe the categorization methods.

Overall, during the data gathering period, we have experienced three distinct types of web-scraping. The first with Google Scholar was a difficult, and in the end, an unsuccessful attempt, because of the restrictions of Google, the different HTML sources, and the custom journal pages. In the second case with arXiv, the extraction of both the metadata and the texts were quite simple and fast, but the quality of the data is still questionable. Lastly, the Jstor data were the easiest to acquire in the terms of data extraction, and Jstor was the most reliable source of papers among the other two. Nevertheless, one should keep in mind that the data request procedure in this volume of data takes up to 2-4 weeks.

id	article_title	date	journal_title	category
30218857	Lower Prices: The Impact of Majoritarian Systems in Democracies Around the World	2008	The Journal of Politics	pol
30218889	Of Crusades and Culture Wars: â€śMessianicâ€ť Militarism and Political Conflict in the United States	2008	The Journal of Politics	pol
30218895	Do Networks Solve Collective Action Problems? Credibility, Search, and Collaboration	2008	The Journal of Politics	pol
30218904	United States Economic Aid and Repression: The Opportunity Cost Argument*	2008	The Journal of Politics	pol
30219444	Financial Regulation, Monetary Policy, and Inflation in the Industrialized World	2008	The Journal of Politics	pol
30219448	Measuring District-Level Partisanship with Implications for the Analysis of U.S. Elections*	2008	The Journal of Politics	pol
30219452	Demography, Democracy and Disputes: The Search for the Elusive Relationship Between Population Growth and International Conflict	2008	The Journal of Politics	pol
30219456	Race, Structure, and State Governments: The Politics of Higher Education Diversity	2008	The Journal of Politics	pol

30219488	Electoral Rules and the Size of the Prize: How Political Institutions Shape Presidential Party Systems	2008	The Journal of Politics	pol
⋮	⋮	⋮	⋮	⋮

Figure 4 Data frame of Jstor papers

3.1.3. Data cleaning

The cleaning procedure of the raw data is slightly different in the case of a string-type dataset because the units of the analysis are letters, words, or sentences i.e. tokens. The goal is to get a well-structured, tokenized text which can be easily and effectively analysed. To achieve that, we have to examine the current format of the texts (e.g. HTML, XML, PDF), and decide which methods are to be used. For example, there are several ways of stemming or lemmatizing[17] words to our interest, but each type of algorithm has its advantages and disadvantages. In the following section, we will describe the cleaning process of the corpus.[18]

In the first case, we had to write a converter[19], because the papers from arXiv were in PDF format. It is crucial to have an easily readable text format with text-mining,[20] since all information is stored within words and sentences. As we can see in Figure 4, even after a successful conversion, there is still a lot to do to tidy up our text: the equations from the paper became nonsense characters (Figure 5) and some words are stuck together. Another important aspect of cleaning is removing stop words[21] and punctuations because these extra characters can increase the amount of required memory and slow down the runtime of the program. Eventually, we decided not to stem or lemmatize the corpus, because it could have led to the undesirable exclusion of important keywords like “listwise” or “pairwise”. An example of the result can be seen in Figure 6.

[17] Stemming and lemmatizing are methods to reduce words to their roots or base e.g. from “imputation” to “imput”.
[18] Throughout the cleaning, we used the `nltk`, `string`, and `enchant` packages.
[19] We used the `tika` and the `textract` package for the conversion.
[20] By „readable”, we are referring to the reading process of a program.
[21] In general, „stop words” are the most frequent words in a language which contain no additional information to the researcher, e.g. „that”, „or”, „an”.

```
[…]however it does require a model for the probability an observationnnnhas complete data to provide flexibility the probabilities can be estimated us nning parametric and nonparametric methods a weighted and penalized objectivennfunction is proposed for variable selection of the linear covariates in the presencennof missing data nnconsider the sample yi xi zi ni with yi[…]
```

Figure 5 Example of an arXiv paper after conversion

```
[…]'however', 'require', 'model', 'probability', 'observation', 'complete', 'data', 'provide', 'flexibility', 'probabilities', 'estimated', 'using', 'parametric', 'nonparametric', 'methods', 'weighted', 'penalized', 'function', 'objective', 'proposed', 'variable', 'selection', 'linear', 'covariates', 'presence', 'missing', 'data', 'consider', 'sample'[…]
```

Figure 6 The cleaned and tokenized text from Figure 4

The corpora from Jstor were cleaner than the one from arXiv, since the Jstor Support Team sent us the papers already in TXT format (Figure 7); therefore, the conversion part has been omitted. Besides tokenization and removing the numbers, punctuations, and meaningless characters, we had to extract the text from the XML tags and lowercase it. One may ask while looking at Figure 8, that after the cleaning, why the stop word "at" stayed in the text? Well, we had to modify the set of stop words, since "at" occurs in the expression "missing at random".

```
<plain_text> <page sequence="1"> SOCIAl CAPItAl, ACADemIC AChIeVemeNt, AND POStgRADUAtION PlANS At AN elIte,  PRIVAte UNIVeRSIty NAthAN D. mARtIN Duke University ABStRACt: Many studies have explored how social capital influences  the academic experiences of secondary school students.[…]
```

Figure 7 Example of a Jstor paper before cleaning

```
'social', 'capital', 'academic', 'achievement', 'plans', 'at', 'elite', 'private', 'university', 'martin', 'duke', 'university', 'abstract', 'many', 'studies', 'explored', 'social', 'capital', 'influences', 'academic', 'experiences', 'secondary', 'school', 'students'[…]
```

Figure 8 The cleaned and tokenized text from Figure 6

The remaining part of the cleaning is more like a classification problem because we had to exclude those papers that explicitly deal with missing data handling methods. Our overall aim is to get a set of papers in which the researchers are only applying missing data handling methods during their examination and not making the research about them. A more elaborate discussion about this matter can be found in the *Limitations of the used methods* section. From a pragmatic point of view, we had to find a classification model, which can divide the two types of papers. After a short manual search through the papers, we saw that the titles contain a handful of information about the nature of the papers; which means that we can try to make a classification model using only the titles of the papers.

The main idea is to choose n keywords that can characterize the titles and make them separable in an n-dimensional vector space spanned by these keywords. In our case, this vector space is 6-dimensional, because we selected 6 keywords („missing", „data", „handling", „observation", „method", „incomplete"). To quantify the keywords and place the titles in this space, we assigned a „weighting" vector[22] to them which should grasp the importance of each keyword (Table 1). We define the value corresponding to the i^{th} keyword of the j^{th} title in the weighting vector as follows:

$$w_{ij} = k_{ij} * \left(\frac{1}{K_i} * 100\right) \ (i = 1 \dots I; j = 1 \dots J) \qquad 3.1$$

Where w_{ij} denotes the weight of the i^{th} keyword in the j^{th} title, I is the number of keywords, J is the number of titles, K_i is the frequency of the i^{th} keyword in all titles, k_{ij} is the frequency of the i^{th} keyword in the j^{th} title.[23]

-	id	date	category	miss	data	handl	obs	method	incomp
1	1108.2835v1	2013	stat	0	0	0	0.474	0	0
2	1404.5793v2	2014	cs	0.135	0	0	0.474	0	0
3	1906.00494v2	2011	stat	0	0.051	0	0.474	0	0
4	1910.00667v1	2019	stat	0.135	0.051	0	0.474	0	0
5	1911.02205v1	2019	cs	0	0	0	0.474	0.25	0
6	1904.07408v1	2019	cs	0	0	0	0.474	0	0

[22] It is similar to the inverse document frequency weight in word embeddings.
[23] In our case, $I = 6$ and $J = 3415$.

7	1209.2669v9	2015	stat	0	0	0	0.474	0	1.587
8	1710.04977v1	2017	stat	0	0	0	0.474	0	0
9	1711.07715v3	2017	cs	0.135	0.051	0	0.474	0	0
10	0609377v3	2006	math	0.135	0	0	0.474	0.25	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 1 Example of keyword vectors (sorted by "obs")

For the sake of example, let's look at the 4^{th} row in Table 1. We have created the vector $\boldsymbol{w}_4 = (0.135, 0.051, 0, 0.474, 0, 0)$ by concatenating $w_{1,4}, w_{2,4}, w_{3,4}, w_{4,4}, w_{5,4}, w_{6,4}$. Let's $w_{1,4}$ denote the weight value corresponding to the keyword "missing" in the 4^{th} title. The number of occurrences of this keyword in the titles is 743 and in the 4^{th} title 1. Consequently, we get the equation:

$$w_{missing,4} = 1 * \left(\frac{1}{743} * 100\right) \approx 0.135 \qquad 3.2$$

At first, to build a classification model we had to filter our dataset to include titles that contain at least one of the keywords, so the above-defined weighting vectors were assigned to a subset of the titles.[24] For simplicity, let us S denote this subset. To *train* our model, we sampled 100 titles from S and fitted a logistic regression model to the selected *training set* (hence S_{Train}). Generally, a logistic regression classifier is called a supervised learning approach in the statistical learning jargon, because we know the values of the outcome variable i.e. the classes.

In this current setup, however, we had to manually fill the outcome variable in S_{train} to give learning material to our model.[25] With this semi-supervised technique, we have fitted the logistic model and made it ready to be tested in the remaining *test* part of the dataset (hence S_{Test}). [26] And again, to be able to test the "goodness" of our classifier, we needed to manually fill the outcome variable in a subsample of S_{test}. After we have fitted our model to the subset of S_{test}, a

[24] This means that from the 33739 title in our sample, we only used 3415 to build our model.
[25] Based on the titles, we have decided whether a paper eligible or not.
[26] For clarification: $S_{train} \cup S_{test} = S \; and \; S_{train} \cap S_{test} = \emptyset$.

confusion matrix has been created of the results (Table 2). [27] As we can see from the confusion matrix, the classes are unbalanced, therefore if we calculate the accuracy, it will be most likely biased. To overcome this problem, we used the Matthews Correlation Coefficient (Matthews 1975)[28] to measure the goodness of our classification model. The MCC is 87.3%, which accounts for a good prediction ability of our model. At last, we applied the classifier to the whole dataset and excluded the papers which were about missing data handling; as a result, we excluded 973 papers from our sample.

		Actual class		Σ
		1	0	
Predicted class	1	30 TP	0 FP	30
	0	6 FN	64 TN	70
Σ		36	64	100

Table 2 Confusion matrix of the classification

3.2. Text-mining

Working with textual data may seem like an unorthodox scientific method, but we hope that the following section will show that one can extract nearly as much useful information from text as from an entirely numerical data. In the following, we will discuss some of the tools that are used generally during text-mining and describe the exact methodology that we used in our research. As mentioned before, these text-mining approaches are rather applied techniques, therefore the concerning literature is not always from peer-reviewed journals or institutionalized scientific communities.

After completing the data cleaning and the exclusion of ineligible cases from our dataset, we have ended up with a tokenized corpus ready to be analysed. The next task is to somehow identify the missing data handling methods that were used during the research. To approach this problem, we first need to explore the texts with keywords to get a picture of the distribution of the relevant

[27] TP = True Positive, TN = True negative, FP = False Positive, FN = False Negative.

[28] $MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$.

words like “imputation”, “deletion”, or “listwise”. Afterwards, we can go deeper and gather the co-occurrences of the relevant words, called *n-gram*s. The collection of n-grams is a widely used and efficient technique in text-mining, which helps to get a better understanding of the unknown corpus. In general, n-gram algorithms create a group from words that are next to each other by combining all possible co-occurrences. One can modify the value of “n”, thereby declare the cardinality of each group. For example, if we are interested in bi-grams (n=2), the algorithm creates groups from two words and counts the frequency of the co-occurrences of them (Figure 9). In this case, the rule of thumb is to use a maximum 4 or 5 as the value of “n”, because above these it is more reasonable to tokenize the texts with sentence units instead of words. In other terms, the tokenization of a text using words as units means that we create uni-grams (n=1). By increasing the value of “n”, we will know more and more about the environment of the relevant keywords and we can start to formulate decision rules to identify the meaning of the co-occurrences.

```
                ⋮
over time: 11
proportional electoral: 11
data set: 10
dependent variable: 9
exchange rate: 9
united states: 9
data sets: 8
electoral rules: 8
iv data: 8
price level: 8
between majoritarian: 7
fixed effects: 7
seats votes: 7
capita gdp: 6
lower real: 6
                ⋮
```

Figure 9 Bi-grams of a text

A similar, but more advanced representational text-mining technique is called *bag-of-words*, when we order the tokenized text into sentence vectors by counting the frequency of each word and assign a coordinate value to them (Figure 10). With a bag-of-words model, we can quantify

the importance of sentences and words. One may recall that during the previous data cleaning phase, we utilized an approach similar to the bag-of-words by assigning a weight vector to each title based on the selected keywords. This consideration leads us to the next step of text representation and analysis, which is a widely used NLP technique: word embedding. With word embedding, we will be able to create word vectors from the texts and visualize them in an *n*-dimensional vector space. Furthermore, the word embedding models can effectively grasp the semantic relations between words, because they use more complex weights to the words than simple binary or integer weights such as "term frequency-inverse document frequency" (tf-idf). Later on, we are going to use the weight vectors to visualize a part of our corpus in the *Vector space representation* section. To classify the papers in our corpus by the type of missing data handling methods, we use the raw frequencies of the keywords[29] and the bi- and tri-grams to build a decision tree.

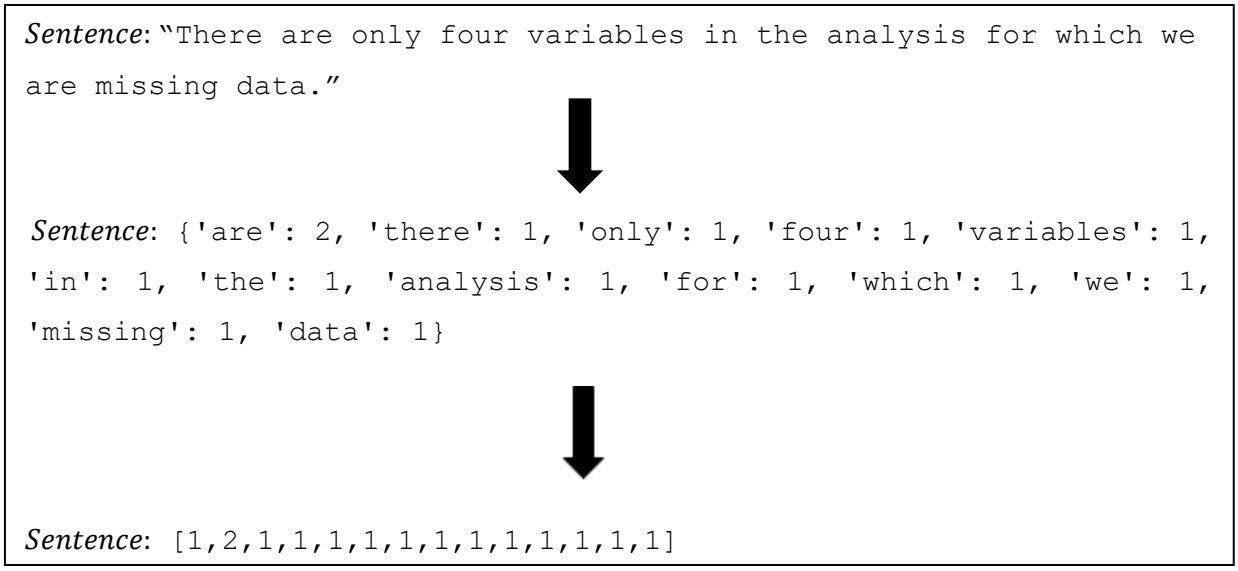


Figure 10 Bag-of-words model

We start the classification with the partition of papers into two classes: if either "imputation" or "substitution" is in the text, then it is classified into the "imputation" branch (Figure 11).[30] This

[29] All keywords can be found in the *Appendix*

[30] Not only these two words were used, but their other forms as well like "imputing" or "substituting". Throughout this section, by mentioning a keyword, we are referring to all possible forms of that word.

first division rule is quite raw because it tells us only whether a paper contains the previously mentioned keywords, and it assumes that our data[31] can be partitioned into two mutually exclusive categories. Let us continue with the "imputation" branch. In the next level, we aim to identify the imputation method by searching for "multiple", "em-algorithm", "fiml" in the ε-wide environment of the first-level keywords. After manually reviewing some context of the keywords, we have decided to maximize the value of ε in 10, which means that in each occurrence of the first-level keywords, our algorithm checks whether the second-level keywords are in their [ε-10; ε+10] environment. If so, then we reach the next leaf and our paper is classified into the "Advanced imputation" category. There is a further filter for the ones that are not classified as "Advanced imputation", because we cannot be sure, that if a paper does not use an advance method, then it will use a basic one. For that reason, we applied again the previous classifier with different keyword parameters to search for co-occurrences and then finished the "imputation" branch with 3 categories: "Advanced imputation", "Basic imputation"," Undefined/other".

Moving to the other side of the tree, we would like to know whether a paper that was not classified to the imputation branch uses some sort of deletion method. After applying the second-level keywords on this side, we have to go one step further again, since we need to narrow down to the exact deletion methods. At the end of this side, we end up with 2 categories: "Deletion method" and "Undefined/other".

[31] The data is already cleaned from ineligible cases.

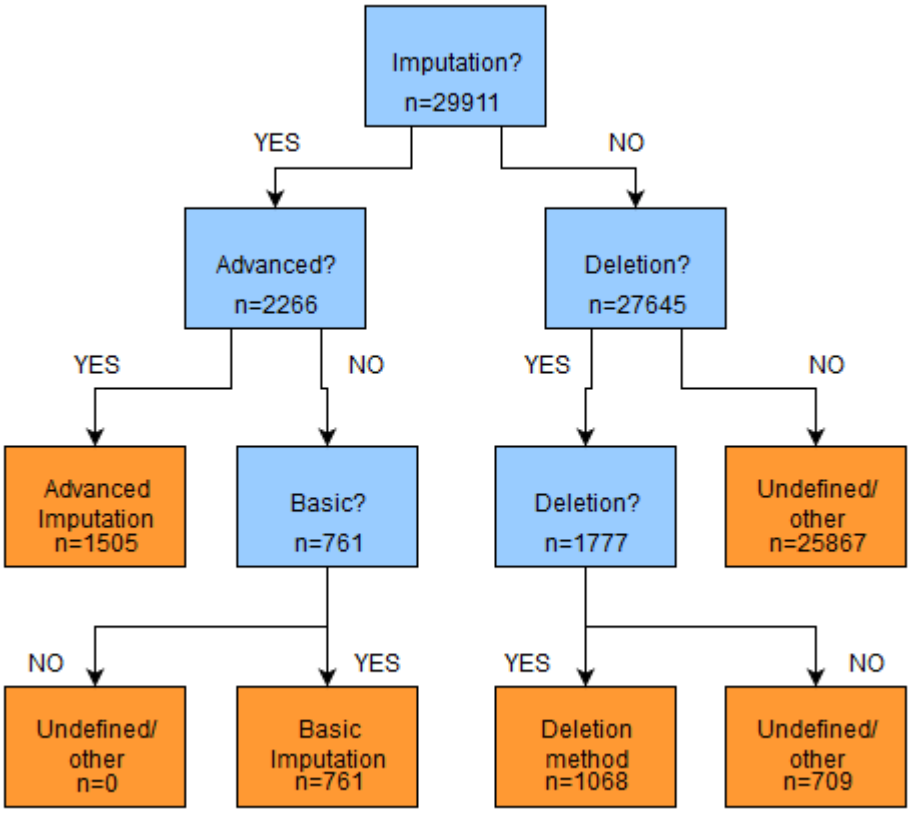


Figure 11 Decision tree for classifying papers with the number of cases in each class

3.2.1. Vector space representation

A more involved and computationally intensive text-mining method will be briefly presented in this section, called Vector space representation of words (Repplinger, Beinborn, and Zuidema 2018; Salton, Wong, and Yang 1975). To be able to place our words in a vector space, we have to utilize almost every tool that we have described in previous chapters. However, the most important aspect of the method will be word embedding, since these embedding weights will declare the place of each word in the vector space. With vector space representation, we gain not only a better visual insight into our corpus, but we can also formalize the interrelation of the words and sentences, by calculating the distance between them. Of course, the distance measures and embedding algorithms can differ, we have used the Euclidean distance and the t-distributed stochastic neighbor embedding (t-SNE) algorithm (van der Maaten and Hinton 2008). Figure 12 shows the embedding result of 100 papers into a 3-dimensional vector space, which can be interactively modified thanks to the embedding projector provided by TensorFlow.[32] We can see that words related to "missing" in these 100 papers are near to it and from the magnitude of each sphere it seems those occur more frequently. We want to emphasize that the building of embedding and representational algorithms requires deeper knowledge of neural networks and statistical learning which is out of the scope of this thesis.

[32] https://projector.tensorflow.org/

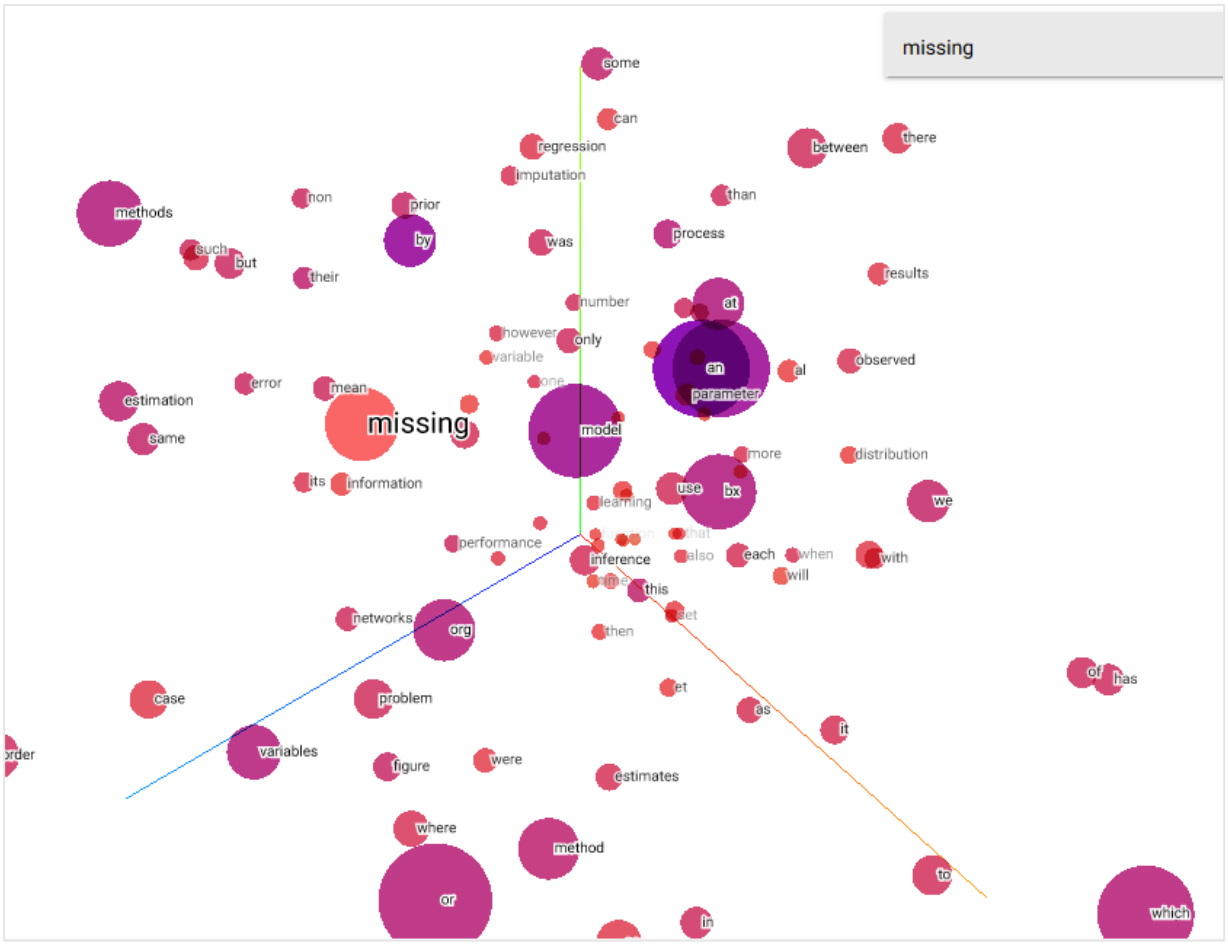


Figure 12 Vector space representation of words from 100 articles using t-SNE

4. Results

All the results that will be presented in this chapter have to be interpreted cautiously, due to the limitations and concerns that will be described in the following *Limitations of the used methods* chapter. The time period of our research was 1999-2016; the created discipline categories can be seen in Figure 13. At the end of this chapter, 4 logistic regression models will be applied to explore the possible associations between disciplines, years, and missing data handling methods.

4.1. Descriptive statistics

The majority of the articles are from the social- and educational research fields (~47%), but there are a considerable amount of scientific works that are from biology (~22%), medical and health sciences (~16%), or other quantitative research fields such as computer science, statistics, or physics (~15%). The dominance of social- and educational science papers may lead to similar results to the previous research.

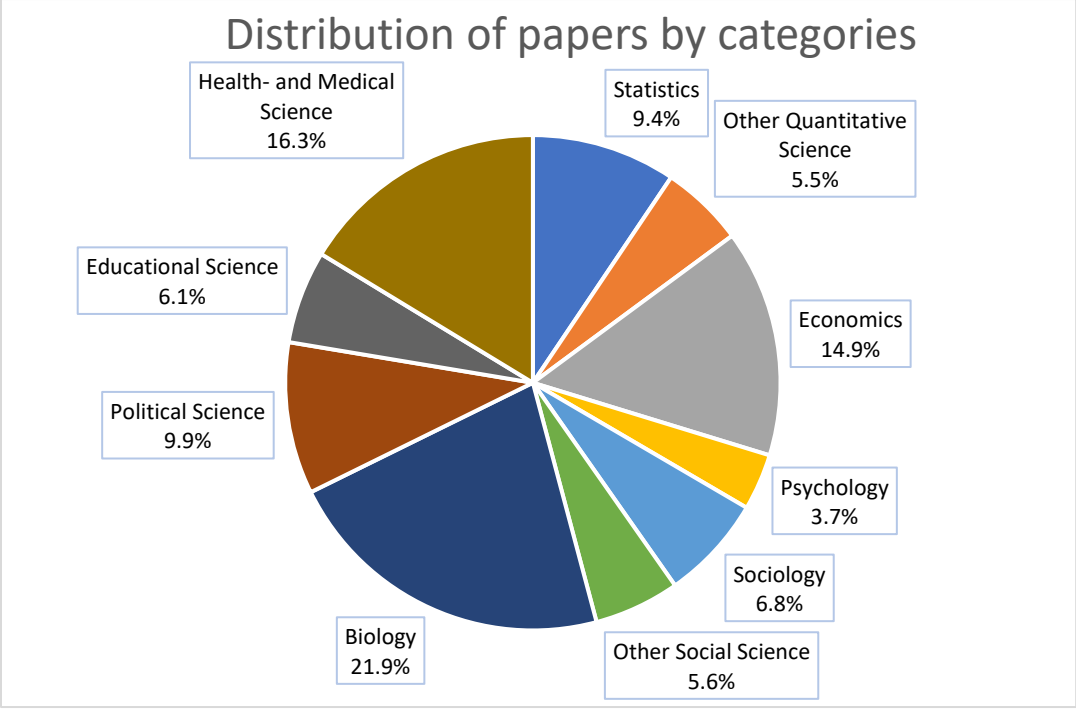


Figure 13 Distribution of papers by disciplines (1999-2016)

For better interpretation, we use the relative frequency of the detected papers in the following diagrams. The number of cases differs among years, so the raw frequency of the detected papers would be misleading i.e. the proportion of them each year is more informative. First, let us have a look at the distribution of papers that were successfully detected with the missing data handling method (Figure 14). A paper is detected if we have managed to identify the missing data handling method that was used in it. In Figure 14, we can see that the relative frequency of not advanced methods is stagnating throughout the years ($SD = 0.63pp, Mean = 6.2\%$), which could mean that the popularity of those techniques is undiminished. Besides, Figure 15 shows a significant widening between the relative frequency of advanced- and basic imputation methods. Furthermore, there is a major growing tendency in the usage of advanced missing data solutions: starting at 1.48%, the relative frequency reached over 7% in 2016. The more than 5 percentage point increase of the advanced methods could also indicate that over the years, more and more paper described the methods they used to handle missing data. Figure 17 strengthens this assumption, since it shows that the rate of successfully detected papers has grown in a similar vein as in the case of advanced methods. Additionally, a slightly decreasing tendency can be seen in the case of deletion methods from 4.69% to 3.45% (Figure 16).

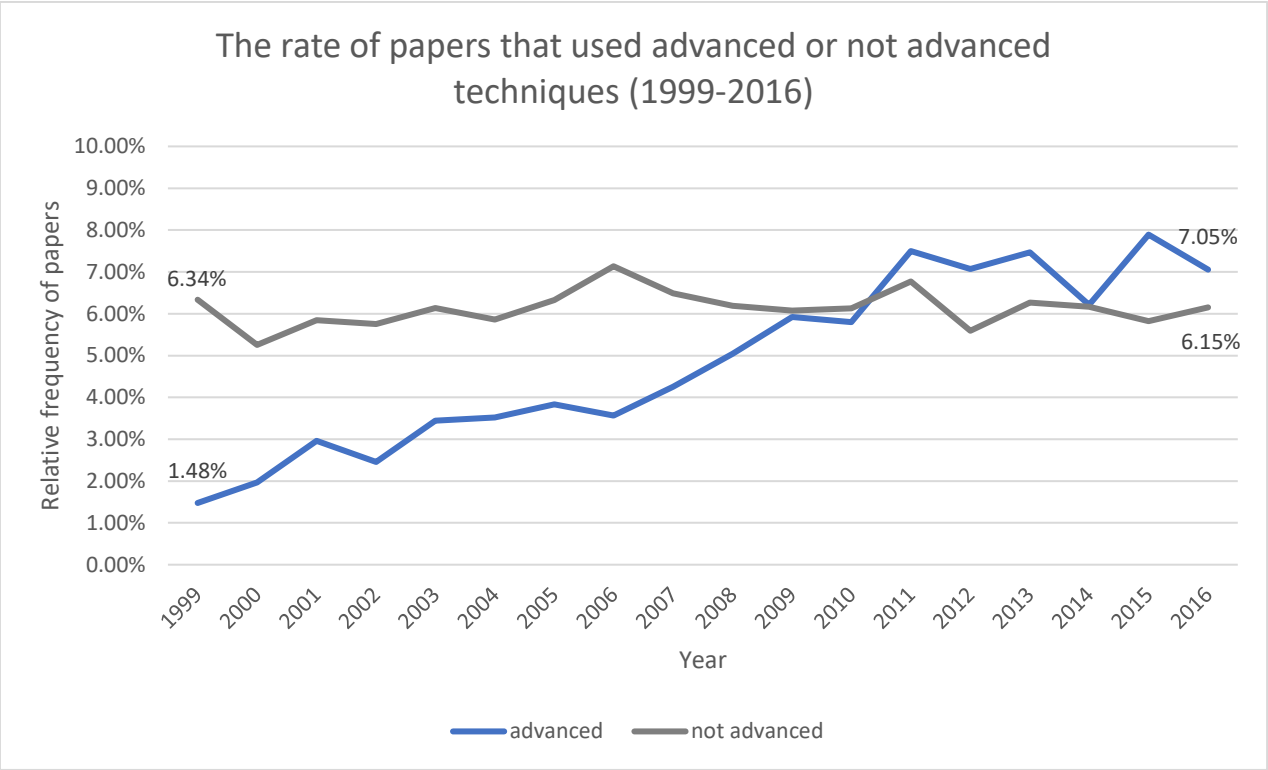


Figure 14 Usage of advanced and not advanced techniques in papers 1999-2016 (relative frequency)

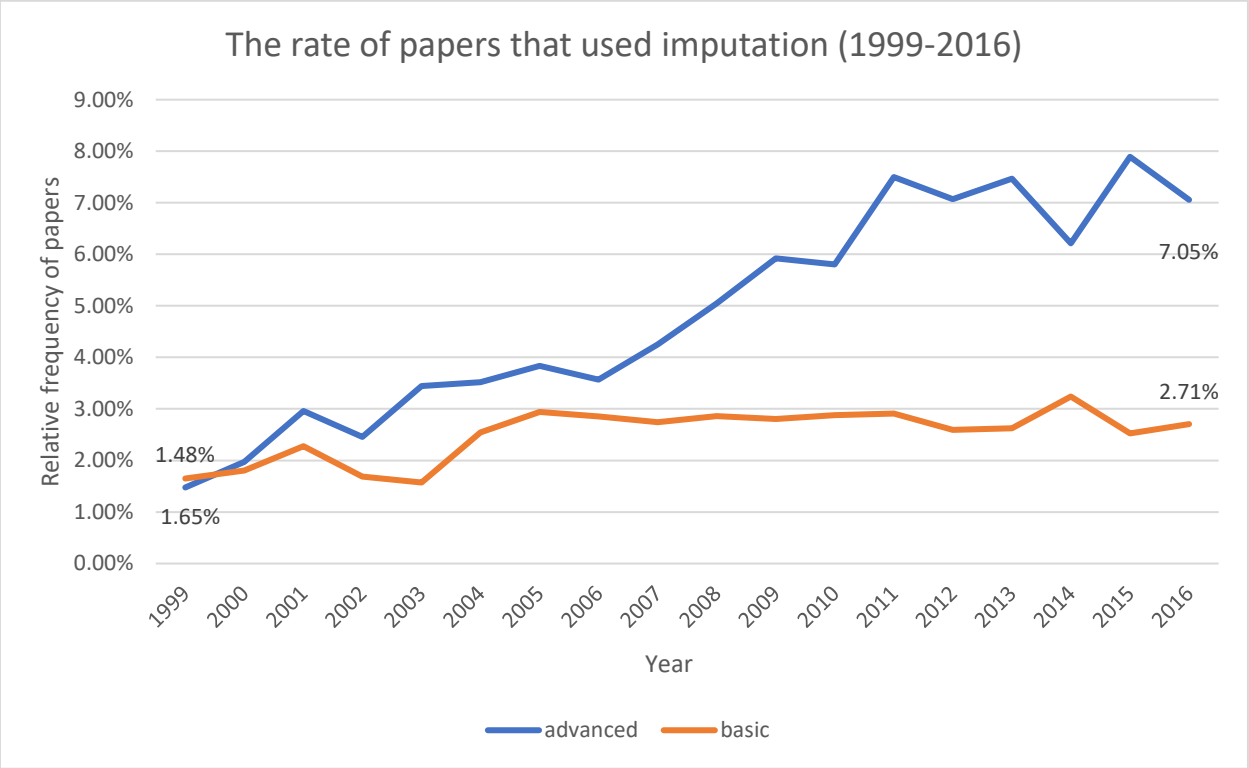


Figure 15 Imputation usage in papers 1999-2016 (relative frequency)

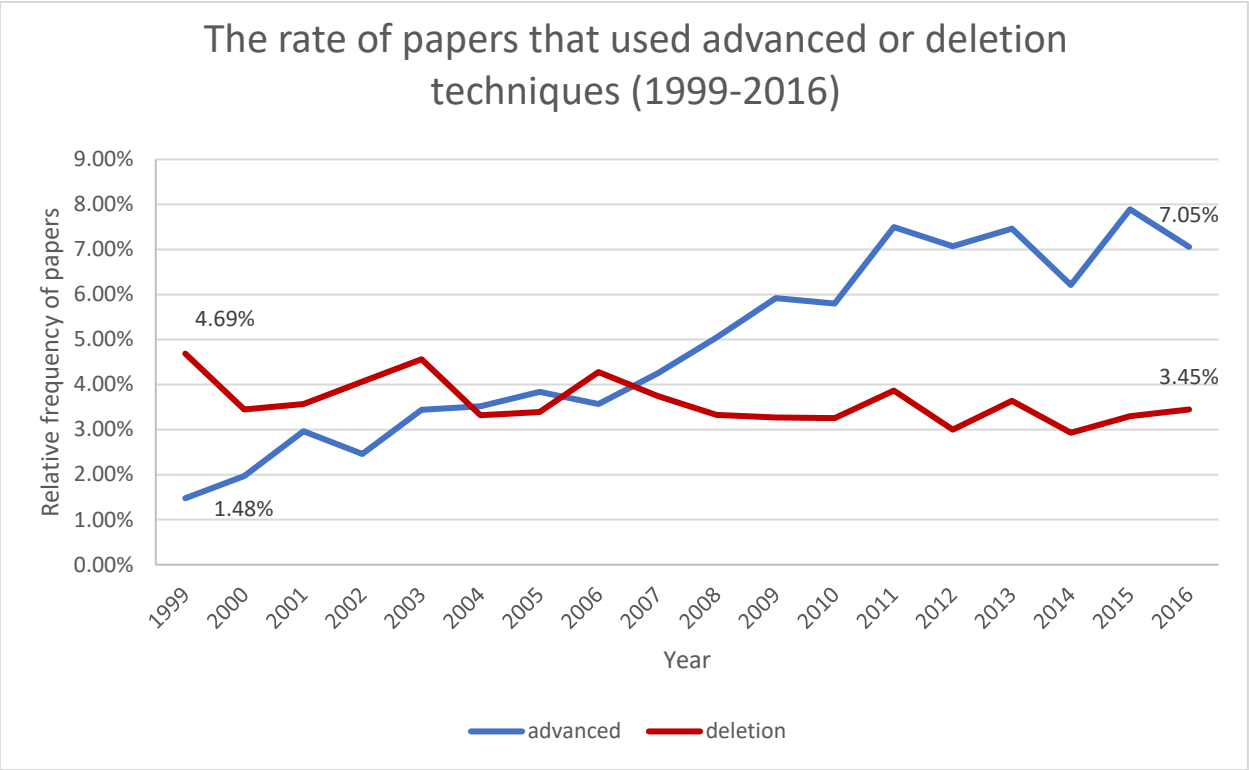


Figure 16 Usage of advanced and deletion techniques in papers 1999-2016 (relative frequency)

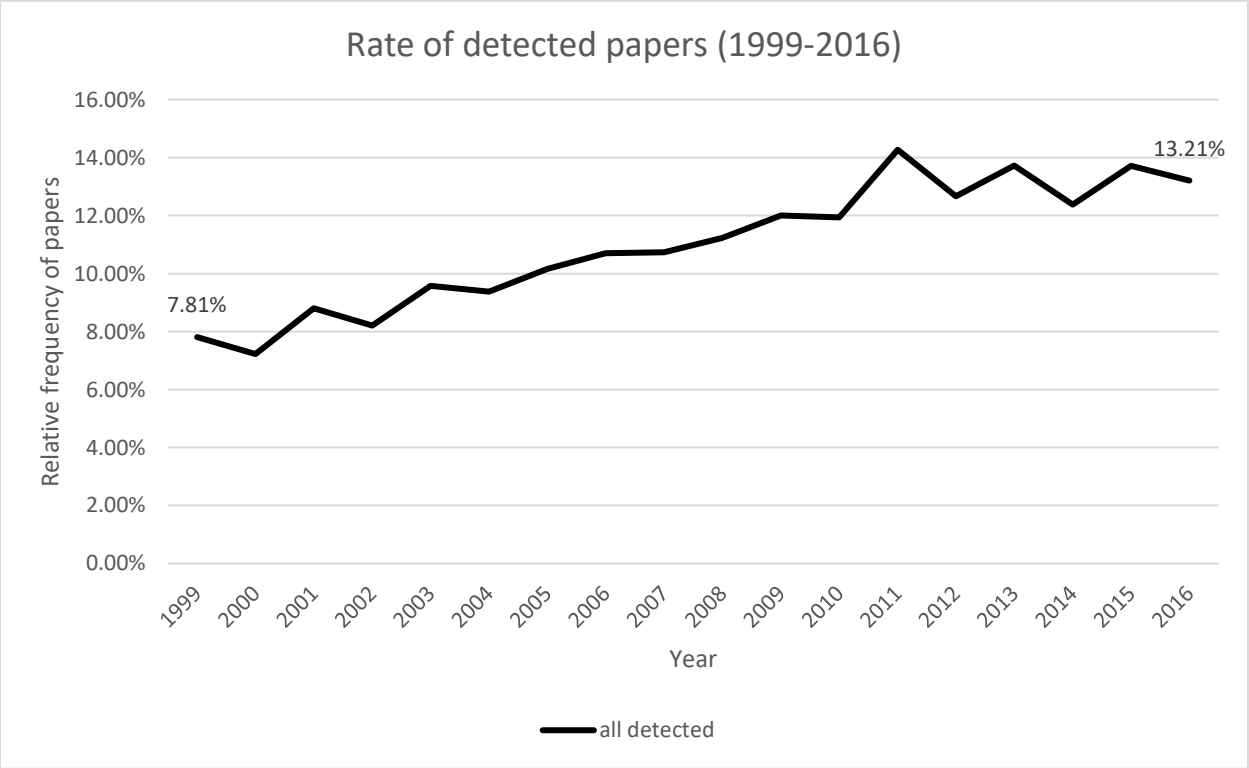
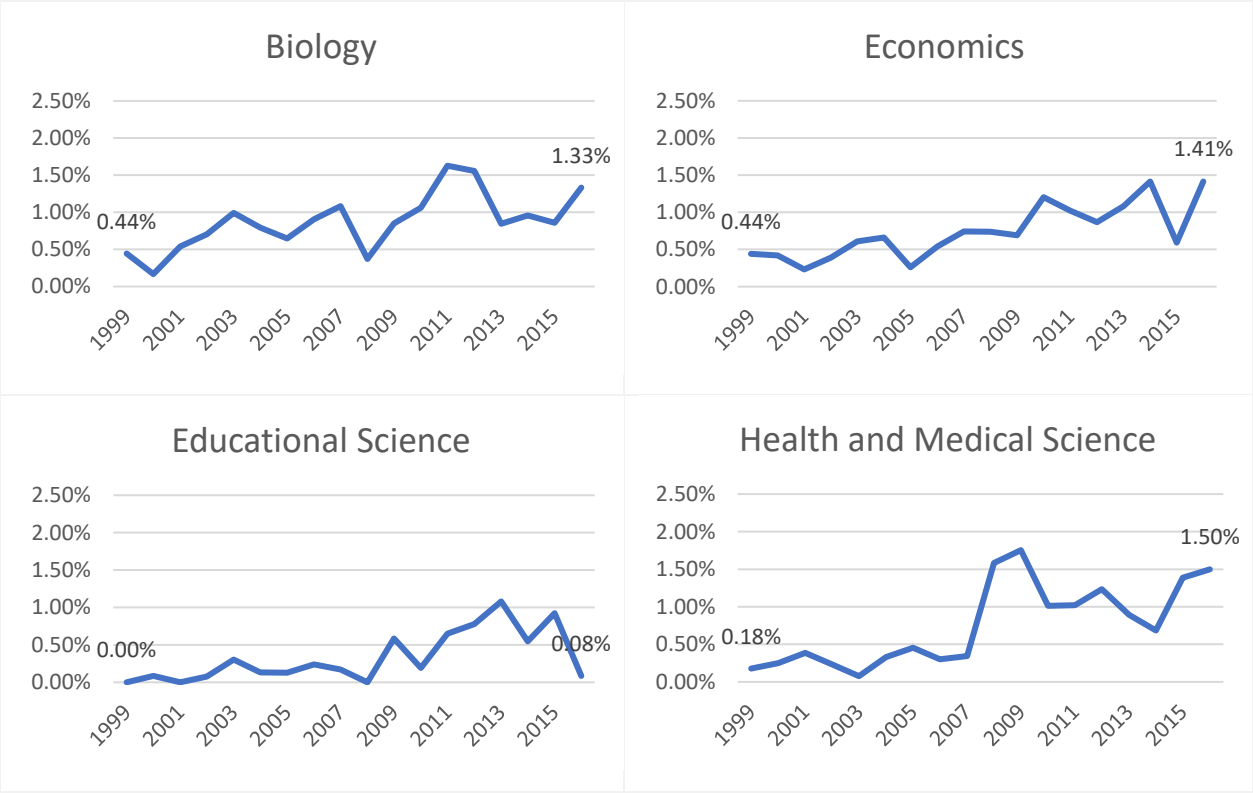


Figure 17 The rate of detected papers 1999-2016 (relative frequency)

As we will argue in the *Limitations of the used methods* section, detecting certain methods highly depends on the authors of the papers and the journals. We can identify a handling method only if it was appropriately documented in the article. Assuming methods from degrees of freedom or sample sizes would have been a more complicated task than a thesis like ours could tackle. Accordingly, the differences between missing data handling methods in disciplines could also mean that the field-specific journals have different guidelines regarding the documentations. As we can see in Figures 18 and 19, if we split our data by discipline categories, then the graphs will become somewhat haphazard because of the small amount of data per year. Nevertheless, some interesting trends can still be examined such as the growing popularity of advanced missing data handling methods in Biology, Economics, Health- and Medical Sciences, or Other Social Sciences. There is a 1 percentage point increase from 1999 to 2016 almost in every previously mentioned category. On the other hand, the usage of not-advanced methods stays nearly unchanged throughout the years among disciplines: a weak decreasing tendency can be seen in Biology and Economics.



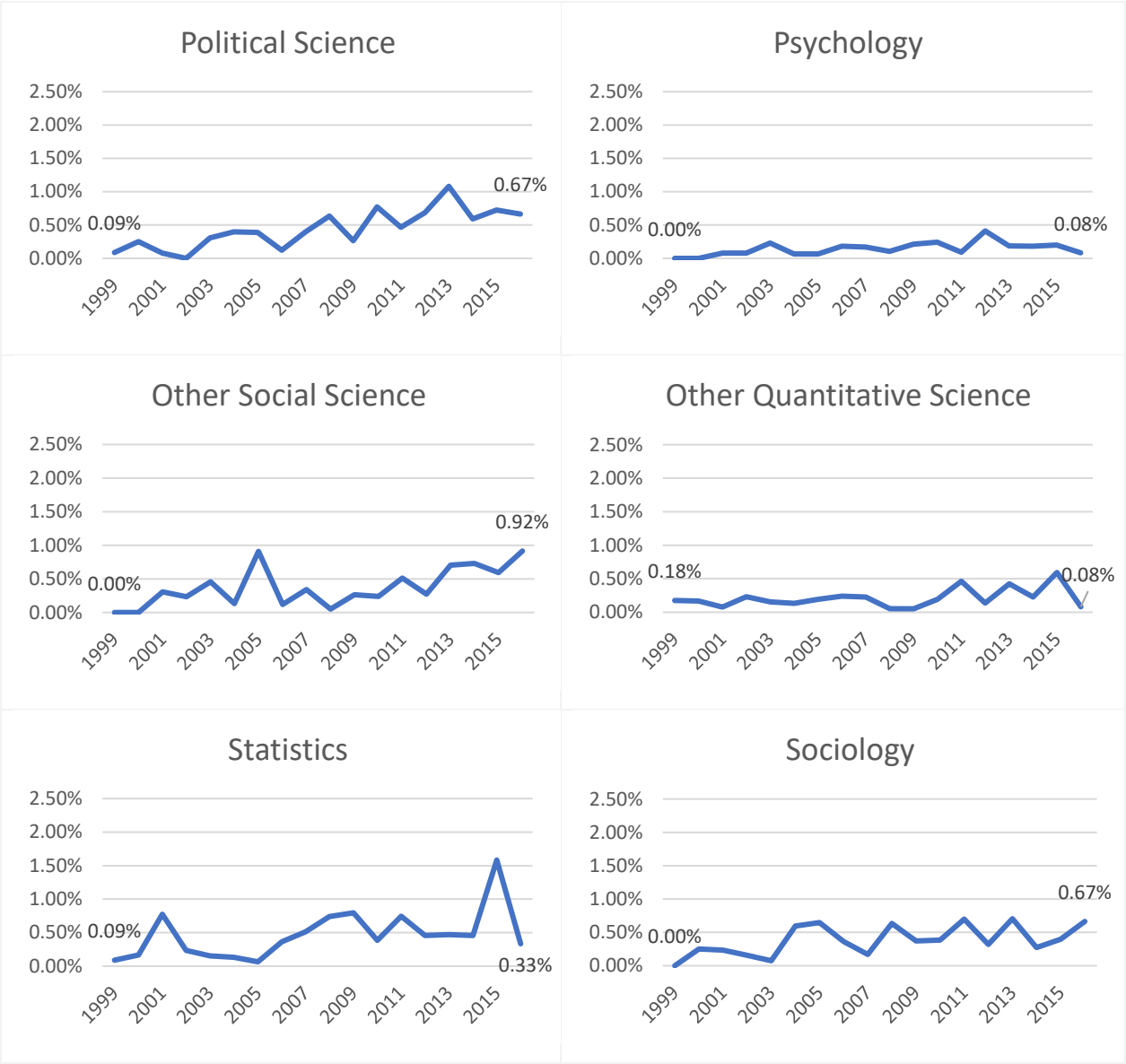
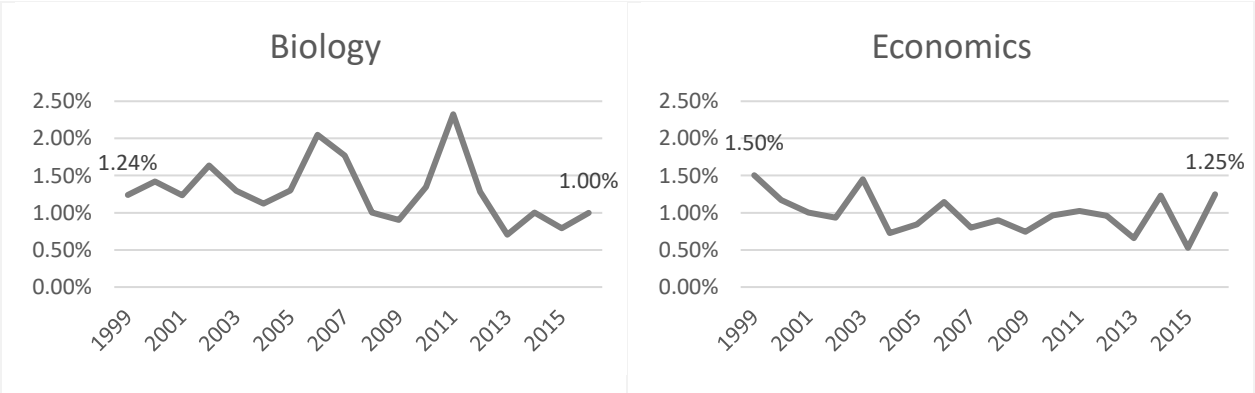


Figure 18 Usage of advanced methods among disciplines 1999-2016 (relative frequency)



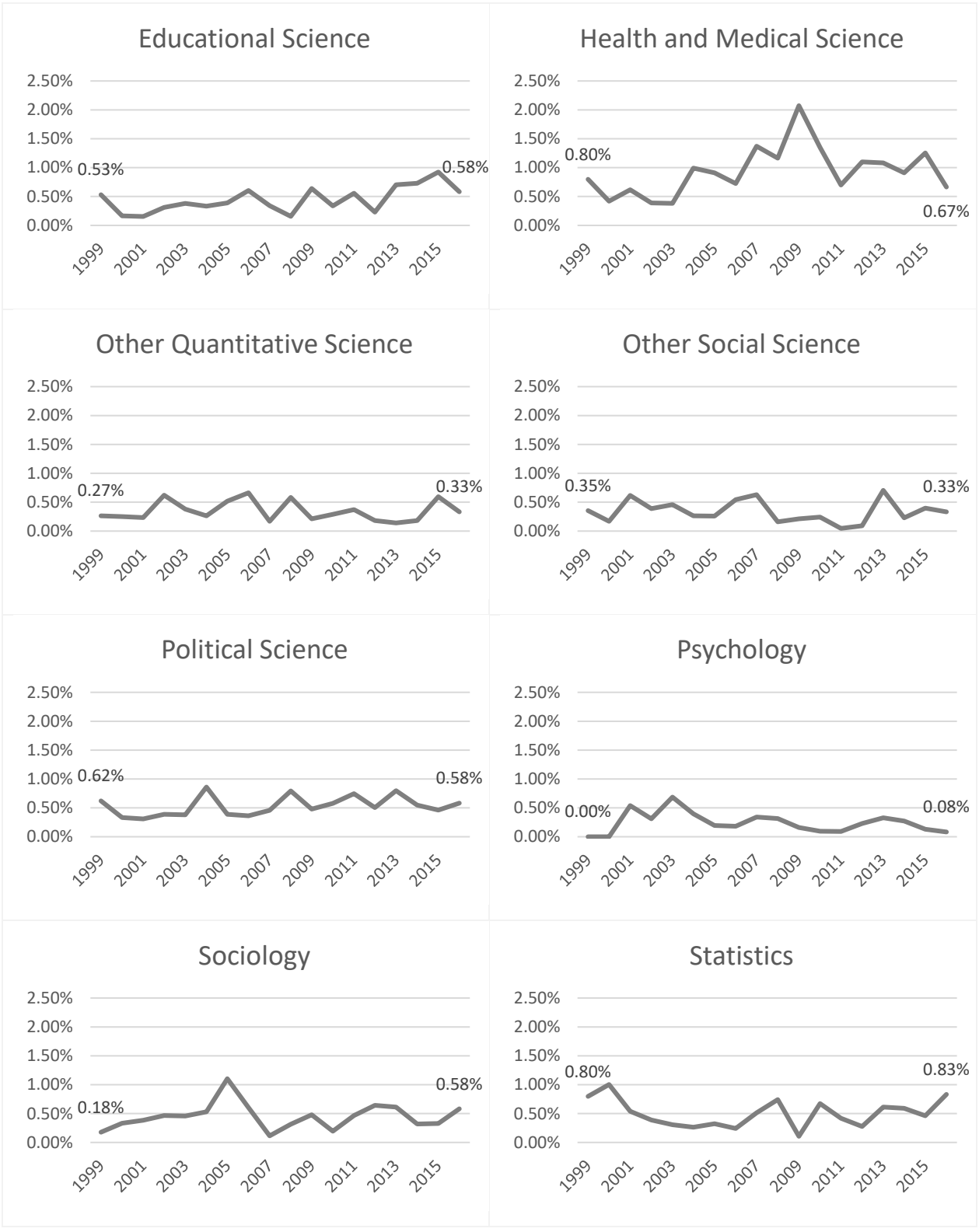


Figure 19 Usage of not advanced methods among disciplines 1999-2016 (relative frequency)

4.2. Logistic regression models

We have already seen the benefits and efficiency of logistic regression in the *Data cleaning* section where we have used it to classify the eligible and ineligible papers. In this section, however, we are interested in the potential associations between years, disciplines, and missing data handling methods. Due to this interest, we are not concerned with the classification accuracy of our models, but with the prediction ability of them. We have built 4 models with the same explanatory variables, but with different outcome variables. The explanatory variables are *year* and *discipline*, both of them as a categorical variable. The outcome variables are *advanced imputation*, *deletion method*, *basic imputation*, and *not advanced method,* respectively.[33]

Model (1) shows that *year* has a positive and significant effect on whether a paper will use some sort of advanced imputation method. Except for 2000, every year has a significant and growing effect[34] on the outcome variable, which means that as time passes, the log-odds of the usage of an advanced handling method increases. The reference categories are 1999 and Biology, which indicates that the coefficient of a category gives the log-odds-ratio of that category and the reference categories. For example, in the case of *2010*, we have a coefficient of 1.434 and it results from the following equation:

$$\log\left(\frac{odds\ of\ advanced\ method\ in\ 1999\ and\ Biology}{odds\ of\ advanced\ method\ in\ 2010\ and\ Biology}\right) \approx 1.434 \qquad 4.1$$

We can conclude that there is more likely that a paper in 2010 uses an advanced imputation method than a paper in 1999 in the field of Biology.

Nevertheless, the growing effect of *year* is not uniform: there are years where the log-odds-ratio drops compared to the previous year e.g. 2002, 2006. [35] To phrase differently – and somewhat harshly –, as we move from 1999 to 2016, it is more likely that a paper uses some kind

[33] Because of our setup these are binary variables: 1-the method was used, 0-the method was not used.

[34] Because of the large sample size and the meaning of "statistical significance", it would be misleading to rely only on the p-values of the parameters. The effect size of our model (i.e. odds ratios) is more informative.

[35] The change can be interpreted as growth because of the ordinal type of *year*.

of advanced imputation method. In the case of disciplines, it is not possible to create an ordering, since one cannot sort disciplines by any means. Consequently, all we can conclude from our models is whether a discipline category affects the usage of missing data handling methods. In model (1), Biology, Economics, Educational Science, Other Social Science, Sociology, and Statistics have a significant effect on the usage of advanced imputation methods. For example, the probability that a paper from Economics in 1999 uses an advanced missing data handling method is calculated as follows:

$$p(advanced\ method = 1|category = Economics) = \frac{e^{-4.422+0.252*1}}{1+e^{-4.422+0.252*1}} \approx 1.52\% \qquad 4.2$$

From model (2), one can deduce that *year* has a negative effect on whether there is a deletion method was used in a paper. In contrast with the previous model, fewer categories of *year* are significant, and the decreasing trend is more fluctuating. Interestingly, the log-odds-ratio of Educational Science category is greater in this model (2) than it is in (1). Model (3) and (4) do not provide additional information about the relationship between missing data handling methods and the explanatory variables, since these models seem to replicate the associations of the former two with a smaller magnitude.

We would like to emphasize that from these models one can only infer whether the usage of a method is more or less likely in a certain year or discipline.

Logistic regression models

	Dependent variable:			
	Advanced Imputation (1)	Deletion method (2)	Basic Imputation (3)	Not Advanced method (4)
Constant	-4.422***	-3.069***	-4.143***	-2.730***
	(0.258)	(0.153)	(0.254)	(0.132)
2000	0.196	-0.357*	0.210	-0.191
	(0.334)	(0.212)	(0.326)	(0.178)
2001	0.655**	-0.362*	0.409	-0.118
	(0.305)	(0.208)	(0.308)	(0.172)
2002	0.515*	-0.148	0.123	-0.078

	(0.312)	(0.198)	(0.326)	(0.171)
2003	0.883***	-0.062	0.067	-0.026
	(0.295)	(0.193)	(0.329)	(0.168)
2004	0.886***	-0.403**	0.508*	-0.101
	(0.289)	(0.201)	(0.296)	(0.165)
2005	0.992***	-0.377*	0.663**	-0.014
	(0.285)	(0.199)	(0.288)	(0.162)
2006	0.896***	-0.127	0.644**	0.125
	(0.286)	(0.185)	(0.286)	(0.156)
2007	1.111***	-0.249	0.621**	0.041
	(0.279)	(0.188)	(0.285)	(0.156)
2008	1.289***	-0.372**	0.642**	-0.021
	(0.273)	(0.190)	(0.281)	(0.156)
2009	1.460***	-0.448**	0.648**	-0.061
	(0.270)	(0.192)	(0.282)	(0.157)
2010	1.434***	-0.398**	0.642**	-0.035
	(0.269)	(0.186)	(0.278)	(0.153)
2011	1.703***	-0.217	0.626**	0.062
	(0.265)	(0.178)	(0.277)	(0.150)
2012	1.615***	-0.512***	0.531*	-0.152
	(0.266)	(0.188)	(0.280)	(0.155)
2013	1.703***	-0.288	0.586**	0.003
	(0.265)	(0.181)	(0.280)	(0.152)
2014	1.475***	-0.535***	0.747***	-0.056
	(0.267)	(0.189)	(0.274)	(0.152)
2015	1.760***	-0.383*	0.523*	-0.082
	(0.269)	(0.200)	(0.295)	(0.164)
2016	1.642***	-0.341	0.619**	-0.014
	(0.276)	(0.210)	(0.302)	(0.171)
Economics	0.252***	0.143	0.024	0.094
	(0.092)	(0.105)	(0.120)	(0.080)
Educational Science	0.339***	0.388***	0.036	0.251**
	(0.116)	(0.131)	(0.162)	(0.103)
Health and Medical Science	0.135	0.097	-0.094	0.015
	(0.091)	(0.104)	(0.121)	(0.080)
Other Quantitative Science	0.042	-0.027	0.245	0.104
	(0.141)	(0.160)	(0.162)	(0.116)
Other Social Science	0.483***	0.127	-0.171	0.002
	(0.115)	(0.149)	(0.182)	(0.117)
Political Science	0.135	0.023	-0.239	-0.088
	(0.106)	(0.123)	(0.149)	(0.096)

Psychology	0.075	0.237	-0.101	0.103
	(0.162)	(0.166)	(0.213)	(0.133)
Sociology	0.356***	0.213	-0.048	0.107
	(0.113)	(0.132)	(0.160)	(0.103)
Statistics	0.284***	-0.098	-0.126	-0.114
	(0.106)	(0.131)	(0.148)	(0.100)
Observations	29,911	29,911	29,911	29,911
Log Likelihood	-5,824.750	-4,590.054	-3,528.387	-6,871.219
Akaike Inf. Crit.	11,703.500	9,234.109	7,110.774	13,796.440

*Table 3 Logistic regression models (*p<0.1, **p<0.05, ***p<0.001)*

5. Limitations of the used methods

We have referenced several times to this section throughout the thesis because all the methods that were used in this research have their well-distinguishable methodological boundaries. The objective of this section is to point out and elaborate on these limitations and offer possible solutions to them (if any). At first, we examine the quality of the collected data and possible bias of the inference with the "Total Survey Error" (henceforth TSE) framework (Groves et al. 2009; Sen et al. 2019), then dive deeper into the challenges of the information extraction from the textual data.

5.1. Total Survey Error

Of course, we cannot fit our methodology perfectly into the TSE framework, because it was formulated specially to survey-type research designs and ours is certainly not. Nonetheless, we can utilize the underlying elements of the paradigm such as the target- and frame population, coverage error, and measurement error.

5.1.1. Coverage

To be able to evaluate our research design with the TSE framework, we had to lay down the target population and the sampling design. Our target population was all the papers that used some kind of missing data handling methods between the time-period 1999-2016 and can be accessed digitally. Furthermore, those papers that explicitly examined missing data handling methods were ineligible. Our frame population, however, can only cover a small portion of the vast amount of relevant papers that were published in this period, but it would not be a problem if the frame population were a random subset of the target population. Evidently, this is a case of a generalization problem, because we cannot ensure that our sample is a random subset of the targeted population. Our papers are from the database of Jstor and arXiv, and there is not any empirical reason to believe that these two sources provide an eligible random subset of the target population. Moreover, there is a possibility of undercoverage bias, since the popular journals with

high impact factor likely to require a subscription to access their content, therefore the papers in these journals were not present in our frame population.

An additional coverage error may arise because of our sample frame: for the search queries, we used the expressions “missing data” and “incomplete observations”, hence the sample frame contains papers which have one of these expressions in their title or the main text; furthermore, it is very likely, that there is a difference between the efficiency of the search engines of the platforms (e.g. Jstor uses OCR to process the files).

5.1.2. Measurement error

Our objective in terms of measurement is to identify the missing data handling methods that were used during each research. The measurement error, in this case, means that we do not measure exactly what we want. This error type has relevance in our research as well, since it is not guaranteed that we can successfully extract the proper information with the available tools during text-mining. As we have seen in the review of the previous results in this topic, even for a human being, it is challenging to identify whether the researcher used some kind of missing data handling method intentionally or just let the software take care of the missing values (Peugh and Enders 2004). Consequently, a classification algorithm can be as sophisticated as possible, but it cannot provide a precise answer about the nature of the missing data handling methods. And this problem does not occur only because of the limitation of the analytic tools, it also depends on the text parser and extraction techniques that were applied.

5.1.3. Processing error and extraction problems

We are certainly in a better position than researchers using survey design in a way that we do not need to scale and quantify the complex emotions or opinions of the respondents. During data processing and coding, researchers have to make serious decisions about the ways of quantification to transform the gathered data into a more feasible form. We are referring especially to those variables which are often reduced to categorical (e.g. age, education) and to the ones that are measured in an integer scale (e.g. trust, mood,). However, this does not mean that there are not any processing errors occurring in a text-mining design. The problem of

quantifying measured objects gains new meaning when we work with textual data, since our task not only consists of the transformation of words to numbers, but of the building of algorithms that do the transformations as well. Let us suppose that we want to count the frequency of the keyword “imputation” in a paper for creating a weight vector for a word embedding algorithm. As we have seen in the *Data cleaning* section, the first step is to convert the PDF file into TXT, XML, HTML, or similar “user-friendly” file-extension. There are several converting packages and programs that can be used[36], but their performance is likely to differ: the identification of equations, numbers, images or the conversion of the PDF source code can cause severe differences in the resulting text (SO, 1; SO, 2); not to mention that sometimes even the words cannot be adequately converted.[37]

Subsequently, the converted text has to be tokenized in order to get a useful structure for the later analysis and classification. As we have mentioned and presented before in the *Data cleaning* and *Text-mining* sections, there are a number of ways to tokenize a text depending on the unit of tokenization, and now we can expand this statement by pointing out that there are a considerable amount of tokenization algorithms among the packages as well. [38] The main difference between these algorithms usually lies in the way of handling punctuations and special characters. As a consequence, they could lead to different results and cause crucial information loss, especially in a case of words that contain dashes or other non-alphabetic characters. We emphasize dashes because one of the keywords is “em-algorithm” and with the RegexpTokenizer, we get “['em', 'algorithm']” without dash as the tokenized words, but with a simple word tokenizer, we get “['em-algorithm']” [39] . This dissimilarity may seem negligible, however in the case of RegexpTokenizer, if we want to remove ineligible words whose length is smaller than 3 to make shorter runtime to our program, then we would unintentionally exclude “em” from the text.

[36] E.g. **tika**, **PyPDF2**, **textract**.

[37] This issue is solely related to the part of the corpus which is from arXiv. A good deal of words were attached by “n”-s, like “multiplennnimputation”, presumably because of the misidentification of whitespaces or line breaks.

[38] For example, the tokenizers in the **nltk** package: https://www.nltk.org/api/nltk.tokenize.html (2020.04.22.).

[39] The script for comparison:

```
import nltk
tokenizer = nltk.RegexpTokenizer(r"\w+")
print(tokenizer.tokenize("em-algorithm"))
print(nltk.word_tokenize("em-algorithm"))
```

Additionally, if we check the tokens whether they are existing words in English with the **enchant** package, then with the simple word tokenizer, we end up omitting the whole "em-algorithm" token. Fortunately, we were able to by-pass this problem with a few adjustments in our tokenizer tools.

5.2. Further considerations

There are a few more important aspects of the data processing that have to be mentioned, namely the categories of disciplines, the classifier model, and the volume of the data. We have described in the *Web-scraping* section that the discipline of each paper was available in the source code and we have managed to extract them. However, there are numerous fields of science, and it was inevitable to reduce the number of categories in our analysis, thus a lot of disciplines were merged into one.

The logistic classifier model that was used in the *Data cleaning* section was a quite good model according to the MCC index, but its "goodness" was tested in a random sample (n=100) of the test set, therefore we cannot accept it without reservation. Furthermore, there likely exists a more robust and complex machine learning model that can perform better, but this question is out of the scope of this thesis.

We also have to mention that the size of the data was relatively big (over 33.000 cases) and it caused unfortunate drawbacks many times during the research. To note an example, we have made use of "nested for loops"[40] on several occasions during our research, and because of the volume of our data, the kernel[41] of the IDE collapsed a number of times.

[40] https://wiki.python.org/moin/ForLoop

[41] The kernel of an IDE is a program, that executes the script. It is like the „engine" of the IDE.

6. Conclusion

The ubiquity of missing data in quantitative research is undeniable. During data processing, researchers must consider the magnitude and type of missingness, since it may cause an undesirable bias or information loss. As argued by Wilkinson and Task Force on Statistical Inference (1999), "[…] – listwise- and pairwise deletion of missing values – are among the worst methods available for practical applications." This statement may seem slightly normative, but it highlights the fact that deletion- and basic imputation methods have serious limitations and drawbacks if those are used without attention. The required Missing Completely at Random mechanism is oftentimes not fulfilled, and the high proportion of missing data could introduce further problems. Instead, one should take into consideration the application of advanced missing data handling methods such as Multiple Imputation or Full Information Maximum Likelihood estimation.

In this thesis, we had two major aims. On one hand, we sought to examine the usage of missing data handling methods across years and disciplines. On the other hand, we have introduced a text-mining approach to perform the analysis and pointed out the limitations of such a methodology. In the case of missing data handling methods, we have found that the popularity of advanced techniques had been growing over the past 20 years, but the not-advanced techniques are still in widespread use. There is a considerable amount of growth in usage of advanced methods in the field of Biology, Economics, Health- and Medical Sciences, and Other Social Sciences. The logistic regression models strengthened the increasing trend of the usage of advanced methods over the years and showed a moderate decline in the application of deletion methods.

The applied text-mining techniques turned out to be not as time-saving as we have previously anticipated, and this approach suffers from the lack of generalizability as well. Furthermore, we have not been able to extract all the necessary and sufficient information to confidently state whether a paper used a missing data handling method, since it is difficult even for the "human eye" to decide. On the contrary, we have gained a new perspective that can be used as an auxiliary approach for further research on this topic: with text-mining techniques, some tasks such as the

data cleaning or keyword search can be automatized and the time-consuming parts of a meta-analysis can be sped up. The word embedding and representational methods can also provide useful insights into the interrelation of certain words or sentences.

7. Bibliography

Bell, Melanie L, Mallorie Fiero, Nicholas J Horton, and Chiu-Hsieh Hsu. 2014. “Handling Missing Data in RCTs; a Review of the Top Medical Journals.” *BMC Medical Research Methodology* 14 (1): 118. https://doi.org/10.1186/1471-2288-14-118.

Bengfort, Benjamin, Rebecca Bilbro, and Tony Ojeda. 2018. *Applied Text Analysis with Python*. O’Reilly.

Bodner, Todd E. 2006. "Missing data: Prevalence and Reporting practices." *Psychological Reports* 99 (3): 675–80. https://doi.org/10.2466/PR0.99.3.675-680.

Buuren, Stef van. 2019. *MICE Package*. R. CRAN. https://stefvanbuuren.name/mice/.

Cham, Heining, Evgeniya Reshetnyak, Barry Rosenfeld, and William Breitbart. 2017. “Full Information Maximum Likelihood Estimation for Latent Variable Interactions With Incomplete Indicators.” *Multivariate Behavioral Research* 52 (1): 12–30. https://doi.org/10.1080/00273171.2016.1245600.

Cheema, Jehanzeb R. 2014. “A Review of Missing Data Handling Methods in Education Research.” *Review of Educational Research* 84 (4): 487–508. https://doi.org/10.3102/0034654314532697.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. “Maximum Likelihood from Incomplete Data Via the EM Algorithm.” *Journal of the Royal Statistical Society: Series B (Methodological)* 39 (1): 1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

Dong, Yiran, and Chao-Ying Joanne Peng. 2013. “Principled Missing Data Methods for Researchers.” *SpringerPlus* 2 (1): 222. https://doi.org/10.1186/2193-1801-2-222.

Enders, Craig K. 2010. *Applied Missing Data Analysis*. Methodology in the Social Sciences. New York: Guilford Press.

Fielding, Shona, Graeme Maclennan, Jonathan A Cook, and Craig R Ramsay. 2008. “A Review of RCTs in Four Medical Journals to Assess the Use of Imputation to Overcome Missing Data in Quality of Life Outcomes.” *Trials* 9 (1): 51. https://doi.org/10.1186/1745-6215-9-51.

Graham, John W, Patricio E Cumsille, and Allison E Shevock. 2013. “Methods for Handling Missing Data.” In *Handbook of Psychology*, Second Edition, 109–41. Wiley.

Graham, John W, and Stewart I Donaldson. 1993. “Evaluating Interventions With Differential Attrition: The Importance of Nonresponse Mechanisms and Use of Follow-Up Data.” *Journal of Applied Psychology* 78 (1): 119–28. https://doi.org/doi:10.1037/0021-9010.78.1.119.

Groves, Robert M, James M Lepkowski, Eleanor Singer, Roger Tourangeau, Floyd J Fowler, and Mick Couper. 2009. *Survey Methodology*. Second Edition. Wiley Series in Survey Methodology. New Jersey: Wiley.

Ichikawa, Mari, Akihiro Hosono, Yuya Tamai, Miki Watanabe, Kiyoshi Shibata, Shoko Tsujimura, Kyoko Oka, et al. 2019. “Handling Missing Data in an FFQ: Multiple Imputation and Nutrient Intake Estimates.” *Public Health Nutrition* 22 (8): 1351–60. https://doi.org/10.1017/S1368980019000168.

Jeličić, Helena, Erin Phelps, and Richard M. Lerner. 2009. “Use of Missing Data Methods in Longitudinal Studies: The Persistence of Bad Practices in Developmental Psychology.” *Developmental Psychology* 45 (4): 1195–99. https://doi.org/10.1037/a0015665.

Kośny, Marek. 2019. “Upper Tail of the Income Distribution in Tax Records and Survey Data. Evidence from Poland.” *Argumenta Oeconomica* 1 (42): 55–80. https://doi.org/10.15611/aoe.2019.1.03.

Little, Roderick J. A., and Donald B. Rubin. 2019. *Statistical Analysis with Missing Data*. Third edition. Wiley Series in Probability and Statistics. Hoboken, NJ: Wiley.

Little, Todd D, Kyle M Lang, Wei Wu, and Mijke Rhemtulla. 2016. “Statistical Issues: What Happens When Data Go Missing?” In *Developmental Psychopathology*, Third Edition, 37. Wiley.

Maaten, Laurens van der, and Geoffrey Hinton. 2008. “Visualizing Data Using T-SNE.” *Journal of Machine Learning Research* 9: 2579–2605.

Matsuyama, Y. 2003. “The α-EM Algorithm: Surrogate Likelihood Maximization Using α-Logarithmic Information Measures.” *IEEE Transactions on Information Theory* 49 (3): 692–706. https://doi.org/10.1109/TIT.2002.808105.

Matthews, B.W. 1975. “Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme.” *Biochimica et Biophysica Acta (BBA) - Protein Structure* 405 (2): 442–51. https://doi.org/10.1016/0005-2795(75)90109-9.

Mitchell, Ryan. 2018. *Web Scraping with Python*. Second Edition. O’Reilly.

Peng, Joanne, Michael Harwell, Show-Mann Liou, and Lee H. Ehman. 2006. “Advances in Missing Data Methods and Implications for Educational Research.” In *Real Data Analysis*, 31–78. Quantitative Methods in Education and the Behavioral Sciences: Issues, Research, and Teaching. Charlotte, NC: Information Age Publishing. https://www.researchgate.net/publication/292794490_Advances_in_missing_data_methods_and_implications_for_educational_research.

Peugh, James L., and Craig K. Enders. 2004. “Missing Data in Educational Research: A Review of Reporting Practices and Suggestions for Improvement.” *Review of Educational Research* 74 (4): 525–56. https://doi.org/10.3102/00346543074004525.

Repplinger, Michael, Lisa Beinborn, and Willem Zuidema. 2018. “Vector-Space Models of Words and Sentences.” *Nieuw Archief Voor Wiskunde* 19 (3): 167– 174.

Roth, Philip L. 1994. “MISSING DATA: A CONCEPTUAL REVIEW FOR APPLIED PSYCHOLOGISTS.” *Personnel Psychology* 47 (3): 537–60. https://doi.org/10.1111/j.1744-6570.1994.tb01736.x.

Rubin, Donald B. 1976. “Inference and Missing Data.” *Biometrika* 63 (3): 581–92.

Salton, G., A. Wong, and C. S. Yang. 1975. “A Vector Space Model for Automatic Indexing.” *Communications of the ACM* 18 (11): 613–20. https://doi.org/10.1145/361219.361220.

Sarkar, Dipanjan. 2019. *Text Analytics with Python*. Second Edition. Apress. https://doi.org/10.1007/978-1-4842-4354-1.

Schafer, Joseph L., and John W. Graham. 2002. “Missing Data: Our View of the State of the Art.” *Psychological Methods* 7 (2): 147–77. https://doi.org/10.1037/1082-989X.7.2.147.

Seffens, William, Chad Evans, and Taylor Minority Health-Grid Network And Herman. 2015. “Machine Learning Data Imputation and Classification in a Multicohort Hypertension Clinical Study.” *Bioinformatics and Biology Insights* 9s3 (January): 43–54. https://doi.org/10.4137/BBI.S29473.

Sen, Indira, Fabian Floeck, Katrin Weller, Bernd Weiss, and Claudia Wagner. 2019. “A Total Error Framework for Digital Traces of Humans.” *ArXiv:1907.08228 [Cs]* Working Paper (December). http://arxiv.org/abs/1907.08228.

Takahashi, Masayoshi. 2017. “Multiple Ratio Imputation by the EMB Algorithm: Theory and Simulation.” *Journal of Modern Applied Statistical Methods* 16 (1): 630–56. https://doi.org/10.22237/jmasm/1493598840.

Takahashi, Masayoshi, Manabu Iwasaki, and Hiroe Tsubaki. 2017. “Imputing the Mean of a Heteroskedastic Log-Normal Missing Variable: A Unified Approach to Ratio Imputation.” *Statistical Journal of the IAOS* 33 (3): 763–76. https://doi.org/10.3233/SJI-160306.

Tomita, Hiroaki, Hironori Fujisawa, and Masayuki Henmi. 2018. “A Bias-Corrected Estimator in Multiple Imputation for Missing Data: A Bias-Corrected Estimator in Multiple Imputation for Missing Data.” *Statistics in Medicine* 37 (23): 3373–86. https://doi.org/10.1002/sim.7833.

Wilkinson, Leland, and Task Force on Statistical Inference. 1999. “Statistical Methods in Psychology Journals.” *American Psychologist* 54 (8): 594–604.

Wood, Angela M, Ian R White, and Simon G Thompson. 2004. “Are Missing Outcome Data Adequately Handled? A Review of Published Randomized Controlled Trials in Major Medical Journals.” *Clinical Trials: Journal of the Society for Clinical Trials* 1 (4): 368–76. https://doi.org/10.1191/1740774504cn032oa.

StackOverflow and StackExchange sources

SO, 1: “How to Extract Text from a PDF File?” Stack Overflow. Accessed April 22, 2020. https://stackoverflow.com/questions/34837707/how-to-extract-text-from-a-pdf-file.

SO, 2: “PyPdf Unable to Extract Text from Some Pages in My PDF.” Stack Overflow. Accessed April 22, 2020. https://stackoverflow.com/questions/4203414/pypdf-unable-to-extract-text-from-some-pages-in-my-pdf.

SE, 1: “How to Get Permission from Google to Use Google Scholar Data, If Needed?” Academia Stack Exchange. Accessed March 9, 2020. https://academia.stackexchange.com/questions/34970/how-to-get-permission-from-google-to-use-google-scholar-data-if-needed.

8. Appendix

8.1. Keywords for classification

8.1.1. Data cleaning: keywords to create weight vectors

„missing", „data", „handling", „observation", „method", „incomplete"

8.1.2. Decision tree

1. Level: "imputation", "substitution"
2. Level: "multiple", "em-algorithm", "full-information maximum likelihood", "listwise", "pairwise", "observation", "case", "value"
3. Level: "mean", "median", "modus", "neighbor", "deck", "regression", "delete", "omit", "exclude"